

```

1  /*<html>
2  <span id="gsh" data-title="GShell" data-author="sato@its-more.jp">
3  <meta charset="UTF-8">
4  <meta name="viewport" content="width=device-width, initial-scale=1.0">
5  <link rel="icon" id="GshFaviconURL" href=""><!-- place holder -->
6  <span id="GshVersion" hidden="">gsh--0.6.6--2020-10-11--SatoxITS</span>
7  <title id="GshTitle">GShell-0.6.6 by SatoxITS</title>
8
9  <div id="GshTopbar" class="MetaWindow"></div>
10 <div id="GshPerfMon"></div>
11 <div id="GshSidebar" class="SbSidebar" style=""><div id="GshIndexer" style=""></div></div>
12 <div id="GshMain">
13 <header id="GshBanner" height="100px" onclick="shiftBG();" style="">
14 <div align="right"><note><a href="http://archive.gshell.org">GShell</a> version 0.6.6 // 2020-10-11 // SatoxITS</note></div>
15 </header>
16
17 <!-- ----- Work { ----- -->
18 <span id="Topbar_WorkCodeSpan">
19 /*
20 <details><summary>Topbar</summary>
21 <!-- ----- Topbar // 2020-1008 SatoxITS { -->
22 <h2>Topbar</h2>
23 <input id="Topbar_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
24 <input id="Topbar_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
25 <input id="Topbar_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
26 <span id="Topbar_WorkCodeView"></span>
27 </details>
28
29 <style>
30 .ConfigIcon {
31   position:absolute;
32   top:-6px;
33   left:92%;
34   width:32px;
35   height:32px;
36 }
37 .MetaWindow {
38   z-index:1000;
39   position:relative;
40   display:block;
41   overflow:visible !important;
42   width:99.9%;
43   height:22px;
44   top:-22px;
45   border:1px solid #22a;
46   margin:0px;
47   left:0.0%;
48   line-height:1.0;
49   font-family:Georgia;
50   color:#fff;
51   font-size:12pt;
52   text-align:center;
53   vertical-align:middle;
54   padding:4px;
55   xxxbackground-color:rgba(0,8,170,0.8);
56   background-color:#3a4861;xxx-PBlue;
57   vertical-align:middle;
58 }
59 .MetaWindow:hover {
60   color:#000;
61   border:1px solid #22a;
62   background-color:rgba(255,255,255,1.0);
63 }
64 </style>
65 <script>
66 function Topbar_openWorkCodeView(){
67   function Topbar_showWorkCode(){
68     showHtmlCode(Topbar_WorkCodeView,Topbar_WorkCodeSpan);
69   }
70   Topbar_WorkCodeViewOpen.addEventListener('click',Topbar_showWorkCode);
71 }
72 Topbar_openWorkCodeView();
73 function ConfigClick(){
74   if( 0 <= AffView.style.zIndex ){
75     AffView.style.saved_zIndex = AffView.style.zIndex;
76     AffView.style.zIndex = -1000;
77     GshSidebar.style.zIndex = -1;
78     GshPerfMon.style.zIndex = -1;
79   }else{
80     AffView.style.zIndex = AffView.style.saved_zIndex;
81     GshSidebar.style.zIndex = 1;
82     GshPerfMon.style.zIndex = 1;
83   }
84   console.log('AffZidex='+AffView.style.zIndex);
85 }
86 function Gshell_initTopbar(){
87   GshTopbar.innerHTML = GshTitle.innerHTML;
88   <!--<img id="ConfigIcon" class="ConfigIcon">
89   if( true ){
90     cfigi = document.createElement('img');
91     cfigi.id = 'ConfigIcon';
92     cfigi.setAttribute('class','ConfigIcon');
93     GshTopbar.appendChild(cfigi);
94     cfigi.src = ConfigICON_DATA;
95
96     //cfigi.style.zIndex = 10000000000;
97     //cfigi.addEventListener('click',ConfigClick);
98     GshTopbar.addEventListener('click',ConfigClick);
99   }
100 }
101 </script>
102 <!-- Topbar_WorkCodeSpan } -->
103 */ </span>
104 <!-- ----- Work } ----- -->
105
106 <!-- ----- Work { ----- -->
107 <span id="Indexer_WorkCodeSpan">
108 /*
109 <details><summary>Indexer</summary>
110 <!-- ----- Indexer // 2020-1007 SatoxITS { -->
111 <h2>Indexer</h2>
112 <input id="Indexer_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
113 <input id="Indexer_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
114 <input id="Indexer_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
115 <span id="Indexer_WorkCodeView"></span>
116 </details>
117 <style id="SidebarIndex">
118 #gsh {
119   display:block;
120   xxxoverflow:scroll !important;
121 }
122 #GshMain {
123   position:relative;
124

```

```

125 width:80% !important;
126 left:19.5% !important;
127 }
128 #GshSidebar {
129 z-index:0;
130 position:relative !important;
131 overflow:auto;
132 resize:both !important;
133 xxoverflow-y:hidden !important;
134 xxxheight:100px !important;
135 xxxdisplay:inline !important;
136 left:0px;
137 top:0px;
138 width:19.5%;
139 min-width:80px;
140 xxxheight:100% !important;
141 height:0px;
142 color:#f00;
143 xxbackground-color:rgba(64,64,64,0.5);
144 xxbackground-color:#DFE3EB;xxx-PBlue;
145 background-color:#eeeeee;xxx-PBlue;
146 }
147 #GshPerfMon {
148 position:relative;
149 z-index:10000000;
150 xxheight:12pt;
151 font-family:monospace, Courier New !important;
152 font-size:9pt !important;
153 color:#f84;
154 top:-20px;
155 }
156 #GshSidebar:hover {
157 z-index:10000000;
158 overflow-x:visible !important;
159 background-color:rgba(255,255,255,0.7);
160 width:50%;
161 }
162 #GshIndexer {
163 z-index:0;
164 position:relative;
165 resize:both !important;
166 height:100%;
167 left:0px;
168 top:0px;
169 scroll-behavior: overflow !important;
170 padding-left:4pt;
171 font-size:0.5em;
172 white-space:nowrap;
173 xxx-background-color:rgba(64,160,64,0.6) !important;
174 color:#7794c6;xxx-PBlue;
175 xxbackground-color:#DFE3EB;xxx-PBlue;
176 background-color:#eeeeee;xxx-PBlue;
177 }
178 #GshIndexer:hover {
179 z-index:10000000;
180 overflow-x:visible !important;
181 color:#000000 !important;xxx-PBlue;
182 xxxbackground-color:#FFFFFF;xxx-PBlue;
183 background-color:rgba(255,255,255,0.7);
184 padding-right:0px;
185 width:80%;
186 }
187 #GshIndexer:select {
188 color:#000000 !important;xxx-PBlue;
189 background-color:#FFFFFF;xxx-PBlue;
190 }
191 .IndexLine {
192 font-size:8pt !important;
193 font-family:Georgia;
194 display:block;
195 xxx-color:#fff;
196 xxx-color:#ffif5;xxx-PBlue;
197 xxx-color:#41516d;xxx-PBlue;
198 xxx-color:#7794c6;xxx-PBlue;
199 padding-right:4pt;
200 }
201 .IndexLine:hover {
202 font-size:10pt !important;
203 xxx-color:#228;
204 xxx-background-color:#fff;
205 xxxcolor:#fff;xxx-PBlue;
206 color:#516487;xxx-PBlue;
207 background-color:rgba(220,220,255,1.0);xxx-PBlue;
208 xxxtext-shadow:1px 1px #3f3;
209 text-shadow:1px 1px #eee;
210 xxxbackground-color:#516487;xxx-PBlue;
211 xxtext-decoration:underline !important;
212 }
213 </style>
214
215 <script id="Indexer_WorkScript">
216 function Indexer_openWorkCodeView(){
217 function Indexer_showWorkCode(){
218 showHtmlCode(Indexer_WorkCodeView,Indexer_WorkCodeSpan);
219 }
220 Indexer_WorkCodeViewOpen.addEventListener('click',Indexer_showWorkCode);
221 }
222 //Indexer_WorkCodeViewOpen.addEventListener('click',Indexer_openWorkCodeView);
223 Indexer_openWorkCodeView();
224
225 var startPerfDate = new Date();
226 var prevPerfDate = startPerfDate;
227 function ShowResourceUsage(){
228 d = new Date();
229 perf = '';
230 perf += '<' + 'font color="gray">UA: ' + window.navigator.userAgent + '<' + '/font><br>\n';
231 perf += DateShort0(startPerfDate) + '<br>\n';
232 perf += DateShort1() + '<br>\n';
233 elps = d.getTime() - startPerfDate.getTime();
234 itvl = d.getTime() - prevPerfDate.getTime();
235 perf += 'Elapsed: ' + elps/1000 + ' s<br>\n';
236 perf += 'Skew: ' + (itvl-1000) + ' ms<br>\n';
237 prevPerfDate = d;
238
239 if( performance.memory !== undefined ){
240 m0 = performance.memory;
241 mu0 = (m0.usedJSHeapSize / 1000000).toFixed(6);
242 perf += 'Memory: '+mu0+' MB<br>\n';
243 }
244 perf += '<br>\n';
245
246 //GshSidebar.innerHTML = perf;
247 GshPerfMon.innerHTML = perf;
248 //GshIndexer.innerHTML = 'Memory: '+mu0+' MB';

```

```

249 //console.log('-- PerfMon heap: '+mu0+'/'+m0.totalHeapSize+'/'+m0.jsHeapSizeLimit);
250 if( true ){
251     GshSidebar.style.zIndex = 1000;
252     GshIndexer.style.zIndex = 0;
253     GshPerfMon.style.zIndex = 1;
254     //GshSidebar.appendChild(GshPerfMon);
255     if( document.getElementById('primary') == null ){ // not in WordPress
256         GshPerfMon.style.position = 'absolute';
257     }
258     GshPerfMon.style.display = 'block';
259     GshPerfMon.style.marginLeft = '4px';
260     GshPerfMon.style.top = '45px';
261     GshPerfMon.style.left = '0px';
262 }
263 }
264 function ResetPerfMon(){
265     GshPerfMon.removeAttribute('style');
266     GshSidebar.removeAttribute('style');
267 }
268
269 var iserno = 0;
270 var GeneratedId = 0;
271 function generateIndex(ni,e,chv,nch,ht){
272     // https://developer.mozilla.org/en-US/docs/Web/API/Element
273     c = '';
274     if( e.classList != null ){
275         c = e.classList.value;
276     }
277     //console.log('-- <'+e.nodeName+'> #' +e.id+' .'+c+' '+e.attributes);
278     if( e.nodeName == '#text' ){ return ''; }
279     if( e.nodeName == '#comment' ){ return ''; }
280     if( e.nodeName == 'H2' || e.nodeName == 'H3' ){
281         id = e.innerHTML;
282         GeneratedId += 1;
283         eid = 'GeneratedId-'+GeneratedId;
284         e.id = eid;
285     }else
286     if( e.nodeName == 'SUMMARY' ){
287         id = e.innerHTML;
288         GeneratedId += 1;
289         eid = 'GeneratedId-'+GeneratedId;
290         e.id = eid;
291     }else
292     if( and(e.nodeName == 'DIV',c=='xxxxxxxxxxxxxxxxentry-content' ) ){
293         console.log('-- DIV entry-content begin');
294         id = e.innerHTML;
295         GeneratedId += 1;
296         eid = 'GeneratedId-'+GeneratedId;
297         e.id = eid;
298         console.log('-- DIV entry-content end hash-child=',e.hasChildNodes());
299     }else
300     if( e.id == '' || e.id == 'undefined' ){
301         return '';
302     }else{
303         id = '#' +e.id;
304         eid = e.id;
305     }
306     iserno += 1;
307     ht = '<'+div id="GeneratedEref_'+iserno+'" class="IndexLine" href="#" +eid+">'
308         + iserno+' '+ni+' '+e.nodeName + ':' + id;
309     if( e.id == '' || e.id == 'undefined' ){ return ht + '<'+div>'; }
310     if( !e.hasChildNodes() ){ return ht + '<'+div>'; }
311     chv = e.childNodes;
312     nch = e.childNodes.length;
313     if( chv != null ){ nch = chv.length; }
314     ht += ' ('+nch+')' + '<'+div>';
315     for( let i = 0; i < chv.length; i++ ){
316         sec = ni+'.'+i;
317         if( ni == '' ){ sec = i; }
318         ht += generateIndex(sec,chv[i],null,0);
319     }
320     return ht;
321 }
322 function onClickIndex(e){
323     tid = e.target.id;
324     tge = document.getElementById(tid);
325     eid = tge.getAttribute('href');
326     rx = tge.getBoundingClientRect().left.toFixed(0)
327     ry = tge.getBoundingClientRect().top.toFixed(0)
328     if( false ){
329         alert('index clicked mouse(x='+e.x+', y='+e.y+')'
330             + '\ntid=#'+ tid + ' rx='+rx + ',ry='+ry
331             + '\neid=' + eid + '\n'
332             + '\nhtml=' + tge.outerHTML);
333     }
334     ee = document.getElementById(eid);
335     sx = 'NaN';
336     sy = ee.getBoundingClientRect().top;
337     console.log('sx='+sx+',sy'+sy);
338     ee.scrollIntoView();
339     window.scrollTo(sx,sy)
340     //window.scrollTo({left:'Non',top:sy,behavior:'smooth'});
341 }
342 function Indexer_afterLoaded(){
343     sideindex = document.getElementById('GshIndexer');
344     ht = '<'+h3>G-Index<'+h3>';
345     ht += generateIndex("",document.getElementById('gsh'),null,0,'');
346     if( (pri = document.getElementById('primary')) != null ){
347         ht += generateIndex("",pri,null,0,'');
348     }
349     ht += '<'+br>';
350     ht += '<'+br>';
351     ht += '<'+br>';
352     ht += '<'+br>';
353     sideindex.innerHTML = ht;
354     sideindex.addEventListener('click',onClickIndex);
355
356     if( (pri = document.getElementById('primary')) != null ){
357         console.log('-- Seems in WordPress');
358         pri.style.zIndex = 2000;
359
360         GshSidebar.style.setProperty('position','relative','important');
361         GshSidebar.style.top = '-1400px';
362         //GshSidebar.style.setProperty('position','absolute','important');
363         //GshSidebar.style.top = '0px';
364
365         GshSidebar.style.setProperty('width','200px','important');
366         GshSidebar.style.setProperty('overflow','scroll','important');
367         GshSidebar.style.resize = 'both';
368
369         GshSidebar.style.left = '-100px';
370         GshIndexer.style.left = '100px';
371         GshIndexer.style.height = '1400px';
372         gsh.appendChild(GshSidebar); // change parent

```

```

373 }else{
374     console.log('-- Seems not in WordPress');
375     GshSidebar.style.setProperty('position','fixed','important');
376 }
377 }
378 //document.addEventListener('load',Indexer_afterLoaded);
379
380 DestroyIndexBar = function(){
381     sideindex = document.getElementById('GshIndexer');
382     sideindex.innerHTML = "";
383     sideindex.style = "";
384 }
385 </script>
386
387 <!-- Indexer_WorkCodeSpan -->
388 *//</span>
389 //<!-- ----- Work } ----- -->
390
391
392 /*
393 <h2>GShell // a General purpose Shell built on the top of Golang</h2>
394 <p>
395 <note>
396 It is a shell for myself, by myself, of myself. --SatoxITS(^~^ )
397 <a href="gsh-0.6.2.go.html">prev.</a>
398 </note>
399 </p>
400 <div id="GJFactory_x"></div>
401
402 <div>
403 <span id="GshMenu">
404 <span class="GshMenu" id="GshMenuEdit" onclick="html_edit();">Edit</span>
405 <span class="GshMenu" id="GshMenuSave" onclick="html_save();">Save</span>
406 <span class="GshMenu" id="GshMenuLoad" onclick="html_load();">Load</span>
407 <span class="GshMenu" id="GshMenuVers" onclick="html_ver0();">Vers</span>
408 <span id="gsh-WinId" onclick="win_jump('0.1');">0</span>
409 <span class="GshMenu" id="gsh-menu-exit" onclick="html_close();"></span>
410 <span class="GshMenu" id="gsh-menu-fork" onclick="html_fork();">Fork</span>
411 <span class="GshMenu" id="GshMenuStop" onclick="html_stop(this,true);">Stop</span>
412 <span class="GshMenu" id="GshMenuFold" onclick="html_fold(this);">Unfold</span>
413 <span class="GshMenu" id="gsh-menu-cksum" onclick="html_digest();">Digest</span>
414 <span class="GshMenu" id="GshMenuSign" onclick="html_sign(this);" style="">Source</span>
415 <!-- | <span id="gsh-menu-pure" onclick="html_pure(this);">Pure</span> -->
416 </span>
417 </div>
418 */
419
420 /*
421 <details id="GshStatement" class="gsh-document"><summary>Statement</summary>
422 <h3>Fun to create a shell</h3>
423 <p>For a programmer, it must be far easy and fun to create his own simple shell
424 rightly fitting to his favor and necessities, than learning existing shells with
425 complex full features that he never use.
426 I, as one of programmers, am writing this tiny shell for my own real needs,
427 totally from scratch, with fun.
428 </p>
429 For a programmer, it is fun to learn new computer languages. For long years before
430 writing this software, I had been specialized to C and early HTML2 :-).
431 Now writing this software, I'm learning Go language, HTML5, JavaScript and CSS
432 on demand as a novice of these, with fun.
433 </p>
434 This single file "gsh.go", that is executable by Go, contains all of the code written
435 in Go. Also it can be displayed as "gsh.go.html" by browsers. It is a standalone
436 HTML file that works as the viewer of the code of itself, and as the "home page" of
437 this software.
438 </p>
439 Because this HTML file is a Go program, you may run it as a real shell program
440 on your computer.
441 But you must be aware that this program is written under situation like above.
442 Needless to say, there is no warranty for this program in any means.
443 </p>
444 <address>Aug 2020, SatoxITS (sato@its-more.jp)</address>
445 </details>
446 */
447 /*
448 <details id="GshFeatures" class="gsh-document"><summary>Features</summary><p>
449 </p>
450 <h3>Cross-browser communication</h3>
451 <p>
452 ... to be written ...
453 </p>
454 <h3>Vi compatible command line editor</h3>
455 <p>
456 The command line of GShell can be edited with commands compatible with
457 <a href="https://www.washington.edu/computing/unix/vi.html"><b>vi</b></a>.
458 As in vi, you can enter <b>command mode</b> by <b>ESC</b> key,
459 then move around in the history by <b>code-j k / ? n N</b>,
460 or within the current line by <b>code-l h f w b o $ &#x27;</b> or so.
461 </p>
462 </details>
463 */
464 /*
465 <details id="gsh-gindex">
466 <summary>Index</summary><div class="gsh-src">
467 Documents
468 <span class="gsh-link" onclick="jumpto_JavaScriptView();">Command summary</span>
469 Go lang part<span class="gsh-src" onclick="document.getElementById('gsh-gocode').open=true;">
470 Package structures
471 <a href="#import">import</a>
472 <a href="#struct">struct</a>
473 Main functions
474 <a href="#comexpansion">str-expansion</a> // macro processor
475 <a href="#finder">finder</a> // builtin find + du
476 <a href="#grep">grep</a> // builtin grep + wc + cksum + ...
477 <a href="#plugin">plugin</a> // plugin commands
478 <a href="#ex-Commands">system</a> // external commands
479 <a href="#builtin">builtin</a> // builtin commands
480 <a href="#network">network</a> // socket handler
481 <a href="#remote-sh">remote-sh</a> // remote shell
482 <a href="#redirect">redirect</a> // StdIn/Out redirection
483 <a href="#history">history</a> // command history
484 <a href="#rusage">rusage</a> // resource usage
485 <a href="#encode">encode</a> // encode / decode
486 <a href="#IME">IME</a> // command line IME
487 <a href="#getline">getline</a> // line editor
488 <a href="#scanf">scanf</a> // string decomposer
489 <a href="#interpreter">interpreter</a> // command interpreter
490 <a href="#main">main</a>
491 </span>
492 JavaScript part
493 <a href="#script-src-view" class="gsh-link" onclick="jumpto_JavaScriptView();">Source</a>
494 <a href="#gsh-data-frame" class="gsh-link" onclick="jumpto_DataView();">Builtin data</a>
495 CSS part
496 <a href="#style-src-view" class="gsh-link" onclick="jumpto_StyleView();">Source</a>

```

```

497 References
498 <a href="#" class="gsh-link" onclick="jumpto_WholeView();">Internal</a>
499 <a href="#gsh-reference" class="gsh-link" onclick="jumpto_ReferenceView();">External</a>
500 Whole parts
501 <a href="#whole-src-view" class="gsh-link" onclick="jumpto_WholeView();">Source</a>
502 <a href="#whole-src-view" class="gsh-link" onclick="jumpto_WholeView();">Download</a>
503 <a href="#whole-src-view" class="gsh-link" onclick="jumpto_WholeView();">Dump</a>
504
505 </div>
506 </details>
507 */
508 <details id="gsh-gocode">
509 <<summary>Go Source</summary><div class="gsh-src" onclick="document.getElementById('gsh-gocode').open=false;">
510 // gsh - Go lang based Shell
511 // (c) 2020 ITS more Co., Ltd.
512 // 2020-0807 created by SatoxITS (sato@its-more.jp)
513
514 package main // gsh main
515
516 // <a name="import">Imported packages</a> // <a href="https://golang.org/pkg/">Packages</a>
517 import (
518     "fmt" // <a href="https://golang.org/pkg/fmt/">fmt</a>
519     "strings" // <a href="https://golang.org/pkg/strings/">strings</a>
520     "strconv" // <a href="https://golang.org/pkg/strconv/">strconv</a>
521     "sort" // <a href="https://golang.org/pkg/sort/">sort</a>
522     "time" // <a href="https://golang.org/pkg/time/">time</a>
523     "bufio" // <a href="https://golang.org/pkg/bufio/">bufio</a>
524     "io/ioutil" // <a href="https://golang.org/pkg/io/ioutil/">ioutil</a>
525     "os" // <a href="https://golang.org/pkg/os/">os</a>
526     "syscall" // <a href="https://golang.org/pkg/syscall/">syscall</a>
527     "plugin" // <a href="https://golang.org/pkg/plugin/">plugin</a>
528     "net" // <a href="https://golang.org/pkg/net/">net</a>
529     "net/http" // <a href="https://golang.org/pkg/net/http/">http</a>
530     "html" // <a href="https://golang.org/pkg/html/">html</a>
531     "path/filepath" // <a href="https://golang.org/pkg/path/filepath/">filepath</a>
532     "go/types" // <a href="https://golang.org/pkg/go/types/">types</a>
533     "go/token" // <a href="https://golang.org/pkg/go/token/">token</a>
534     "encoding/base64" // <a href="https://golang.org/pkg/encoding/base64/">base64</a>
535     "unicode/utf8" // <a href="https://golang.org/pkg/unicode/utf8/">utf8</a>
536     // "gshdata" // gshell's logo and source code
537     "hash/crc32" // <a href="https://golang.org/pkg/unicode/hash/crc32/">crc32</a>
538     "golang.org/x/net/websocket"
539 )
540
541 // // 2020-0906 added,
542 // <a href="https://golang.org/cmd/cgo/">CGo</a>
543 // #include "poll.h" // <poll.h> // </poll.h> to be closed as HTML tag :-p
544 // typedef struct { struct pollfd fdv[8]; } pollFdv;
545 // int pollx(pollFdv *fdv, int nfds, int timeout){
546 //     return poll(fdv->fdv,nfds,timeout);
547 // }
548 import "C"
549
550 // // 2020-0906 added,
551 func CFPollIn1(fp*os.File, timeoutUs int)(ready uintptr){
552     var fdv = C.pollFdv{}
553     var nfds = 1
554     var timeout = timeoutUs/1000
555
556     fdv.fdv[0].fd = C.int(fp.Fd())
557     fdv.fdv[0].events = C.POLLIN
558     if( 0 < EventRecvFd ){
559         fdv.fdv[1].fd = C.int(EventRecvFd)
560         fdv.fdv[1].events = C.POLLIN
561         nfds += 1
562     }
563     r := C.pollx(&fdv,C.int(nfds),C.int(timeout))
564     if( r <= 0 ){
565         return 0
566     }
567     if (int(fdv.fdv[1].revents) & int(C.POLLIN)) != 0 {
568         //fprintf(stderr,"--De-- got Event\n");
569         return uintptr(EventFdOffset + fdv.fdv[1].fd)
570     }
571     if (int(fdv.fdv[0].revents) & int(C.POLLIN)) != 0 {
572         return uintptr(NormalFdOffset + fdv.fdv[0].fd)
573     }
574     return 0
575 }
576
577 const (
578     NAME = "gsh"
579     VERSION = "0.6.6"
580     DATE = "2020-10-11"
581     AUTHOR = "SatoxITS(^-^)"
582 )
583 var (
584     GSH_HOME = ".gsh" // under home directory
585     GSH_PORT = 9999
586     MaxStreamSize = int64(128*1024*1024*1024) // 128GiB is too large?
587     PROMPT = ">"
588     LINESIZE = (8*1024)
589     PTHSEP = ":" // should be ";" in Windows
590     DIRSEP = "/" // canbe \ in Windows
591 )
592
593 // -xX logging control
594 // --A-- all
595 // --I-- info.
596 // --D-- debug
597 // --T-- time and resource usage
598 // --W-- warning
599 // --E-- error
600 // --F-- fatal error
601 // --Xn- network
602
603 // <a name="struct">Structures</a>
604 type GCommandHistory struct {
605     StartAt time.Time // command line execution started at
606     EndAt time.Time // command line execution ended at
607     ResCode int // exit code of (external command)
608     CmdError error // error string
609     OutData *os.File // output of the command
610     FoundFile []string // output - result of ufind
611     Rusageve [2]syscall.Rusage // Resource consumption, CPU time or so
612     CmdId int // maybe with identified with arguments or impact
613     // redirection commands should not be the CmdId
614     WorkDir string // working directory at start
615     WorkDirX int // index in ChdirHistory
616     CmdLine string // command line
617 }
618 type GChdirHistory struct {
619     Dir string
620     MovedAt time.Time

```

```

621     CmdIndex    int
622 }
623 type CmdMode struct {
624     BackGround bool
625 }
626 type Event struct {
627     when        time.Time
628     event       int
629     evarg       int64
630     CmdIndex    int
631 }
632 var CmdIndex int
633 var Events []Event
634 type PluginInfo struct {
635     Spec        *plugin.Plugin
636     Addr        plugin.Symbol
637     Name        string // maybe relative
638     Path        string // this is in Plugin but hidden
639 }
640 type GServer struct {
641     host        string
642     port        string
643 }
644
645 // <a href="https://tools.ietf.org/html/rfc3230">Digest</a>
646 const ( // SumType
647     SUM_ITEMS    = 0x000001 // items count
648     SUM_SIZE     = 0x000002 // data length (simplly added)
649     SUM_SIZEHASH = 0x000004 // data length (hashed sequence)
650     SUM_DATEHASH = 0x000008 // date of data (hashed sequence)
651     // also envelope attributes like time stamp can be a part of digest
652     // hashed value of sizes or mod-date of files will be useful to detect changes
653
654     SUM_WORDS    = 0x000010 // word count is a kind of digest
655     SUM_LINES    = 0x000020 // line count is a kind of digest
656     SUM_SUM64    = 0x000040 // simple add of bytes, useful for human too
657
658     SUM_SUM32_BITS = 0x000100 // the number of true bits
659     SUM_SUM32_2BYTE = 0x000200 // 16bits words
660     SUM_SUM32_4BYTE = 0x000400 // 32bits words
661     SUM_SUM32_8BYTE = 0x000800 // 64bits words
662
663     SUM_SUM16_BSD = 0x001000 // UNIXsum -sum -bsd
664     SUM_SUM16_SYSV = 0x002000 // UNIXsum -sum -sysv
665     SUM_UNIXFILE  = 0x004000
666     SUM_CRCIEEE   = 0x008000
667 )
668 type CheckSum struct {
669     Files    int64 // the number of files (or data)
670     Size     int64 // content size
671     Words    int64 // word count
672     Lines    int64 // line count
673     SumType  int
674     Sum64    uint64
675     Crc32Table  crc32.Table
676     Crc32Val   uint32
677     Sum16      int
678     Ctime      time.Time
679     Atime      time.Time
680     Mtime      time.Time
681     Start      time.Time
682     Done       time.Time
683     RusageAtStart [2]syscall.Rusage
684     RusageAtEnd  [2]syscall.Rusage
685 }
686 type ValueStack [][]string
687 type GshContext struct {
688     StartDir    string // the current directory at the start
689     GetLine     string // gsh-getline command as a input line editor
690     ChdirHistory []GchdirHistory // the 1st entry is wd at the start
691     gshPA       syscall.ProcAttr
692     CommandHistory []GCommandHistory
693     CmdCurrent   GCommandHistory
694     BackGround  bool
695     BackGroundJobs []int
696     LastRusage   syscall.Rusage
697     GshHomeDir   string
698     TerminalId   int
699     CmdTrace     bool // should be [map]
700     CmdTime      bool // should be [map]
701     PluginFuncs []PluginInfo
702     iValues      []string
703     iDelimiter   string // field sepearater of print out
704     iFormat      string // default print format (of integer)
705     iValStack    ValueStack
706     LastServer   GServer
707     RSERVER      string // [gsh://]host[:port]
708     RWD          string // remote (target, there) working directory
709     lastCheckSum CheckSum
710 }
711
712 func nsleep(ns time.Duration){
713     time.Sleep(ns)
714 }
715 func usleep(ns time.Duration){
716     nsleep(ns*1000)
717 }
718 func msleep(ns time.Duration){
719     nsleep(ns*1000000)
720 }
721 func sleep(ns time.Duration){
722     nsleep(ns*1000000000)
723 }
724
725 func strBegins(str, pat string)(bool){
726     if len(pat) <= len(str){
727         yes := str[0:len(pat)] == pat
728         //fmt.Printf("---D-- strBegins(%v,%v)=%v\n",str,pat, yes)
729         return yes
730     }
731     //fmt.Printf("---D-- strBegins(%v,%v)=%v\n",str,pat,false)
732     return false
733 }
734 func isin(what string, list []string) bool {
735     for _, v := range list {
736         if v == what {
737             return true
738         }
739     }
740     return false
741 }
742 func isinX(what string,list[]string)(int){
743     for i,v := range list {
744         if v == what {

```

```

745     return i
746   }
747 }
748 return -1
749 }
750
751 func env(opts []string) {
752   env := os.Environ()
753   if isin("-", opts){
754     sort.Slice(env, func(i, j int) bool {
755       return env[i] < env[j]
756     })
757   }
758   for _, v := range env {
759     fmt.Printf("%v\n",v)
760   }
761 }
762
763 // - rewriting should be context dependent
764 // - should postpone until the real point of evaluation
765 // - should rewrite only known notation of symbol
766 func scanInt(str string)(val int,leng int){
767   leng = -1
768   for i,ch := range str {
769     if '0' <= ch && ch <= '9' {
770       leng = i+1
771     }else{
772       break
773     }
774   }
775   if 0 < leng {
776     ival,_ := strconv.Atoi(str[0:leng])
777     return ival,leng
778   }else{
779     return 0,0
780   }
781 }
782
783 func substHistory(gshCtx *GshContext,str string,i int,rstr string)(leng int,rst string){
784   if len(str[i+1:]) == 0 {
785     return 0,rstr
786   }
787   hi := 0
788   histlen := len(gshCtx.CommandHistory)
789   if str[i+1] == '!' {
790     hi = histlen - 1
791     leng = 1
792   }else{
793     hi,leng = scanInt(str[i+1:])
794     if leng == 0 {
795       return 0,rstr
796     }
797     if hi < 0 {
798       hi = histlen + hi
799     }
800   }
801   if 0 <= hi && hi < histlen {
802     var ext byte
803     if 1 < len(str[i+leng:]){
804       ext = str[i+leng:][1]
805     }
806     //fmt.Printf("--D-- %v(%c)\n",str[i+leng:],str[i+leng])
807     if ext == 'f' {
808       leng += 1
809       xlist := []string{}
810       list := gshCtx.CommandHistory[hi].FoundFile
811       for _,v := range list {
812         //list[i] = escapeWhiteSP(v)
813         xlist = append(xlist,escapeWhiteSP(v))
814       }
815       rstr += strings.Join(list," ")
816       rstr += strings.Join(xlist," ")
817     }else
818     if ext == 'e' || ext == 'd' {
819       // IN@ .- workdir at the start of the command
820       leng += 1
821       rstr += gshCtx.CommandHistory[hi].WorkDir
822     }else{
823       rstr += gshCtx.CommandHistory[hi].CmdLine
824     }
825   }else{
826     leng = 0
827   }
828   return leng,rstr
829 }
830
831 func escapeWhiteSP(str string)(string){
832   if len(str) == 0 {
833     return "\\z" // empty, to be ignored
834   }
835   rstr := ""
836   for _,ch := range str {
837     switch ch {
838       case '\\': rstr += "\\\\"
839       case ' ': rstr += "\\s"
840       case '\t': rstr += "\\t"
841       case '\r': rstr += "\\r"
842       case '\n': rstr += "\\n"
843       default: rstr += string(ch)
844     }
845   }
846   return rstr
847 }
848
849 func unescapeWhiteSP(str string)(string){ // strip original escapes
850   rstr := ""
851   for i := 0; i < len(str); i++ {
852     ch := str[i]
853     if ch == '\\' {
854       if i+1 < len(str) {
855         switch str[i+1] {
856           case 'z':
857             continue;
858         }
859       }
860     }
861     rstr += string(ch)
862   }
863   return rstr
864 }
865
866 func unescapeWhiteSPV(strv []string)([]string){ // strip original escapes
867   ustrv := []string{}
868   for _,v := range strv {
869     ustrv = append(ustrv,unescapeWhiteSP(v))
870   }
871   return ustrv
872 }

```

```

869
870 // <a name="comexpansion">str-expansion</a>
871 // - this should be a macro processor
872 func strsubst(gshCtx *GshContext, str string, histonly bool) string {
873     rbuff := []byte{}
874     if false {
875         //@@U Unicode should be cared as a character
876         return str
877     }
878     //rstr := ""
879     inEsc := 0 // escape characer mode
880     for i := 0; i < len(str); i++ {
881         //fmt.Printf("--D--Subst %v:%v\n",i,str[i:])
882         ch := str[i]
883         if inEsc == 0 {
884             if ch == '\\' {
885                 //leng,xrstr := substHistory(gshCtx,str,i,rstr)
886                 leng,rs := substHistory(gshCtx,str,i,"")
887                 if 0 < leng {
888                     //_,rs := substHistory(gshCtx,str,i,"")
889                     rbuff = append(rbuff,[]byte(rs)...)
890                     i += leng
891                     //rstr = xrstr
892                     continue
893                 }
894             }
895             switch ch {
896                 case '\\': inEsc = '\\'; continue
897                 case '%': inEsc = '%'; continue
898                 case '$':
899             }
900         }
901         switch inEsc {
902             case '\\':
903                 switch ch {
904                     case '\\': ch = '\\'
905                     case 's': ch = ' '
906                     case 't': ch = '\t'
907                     case 'r': ch = '\r'
908                     case 'n': ch = '\n'
909                     case 'z': inEsc = 0; continue // empty, to be ignored
910                 }
911                 inEsc = 0
912             case '%':
913                 switch {
914                     case ch == '%': ch = '%'
915                     case ch == 'T':
916                         //rstr = rstr + time.Now().Format(time.Stamp)
917                         rs := time.Now().Format(time.Stamp)
918                         rbuff = append(rbuff,[]byte(rs)...)
919                         inEsc = 0
920                         continue;
921                     default:
922                         // postpone the interpretation
923                         //rstr = rstr + "%" + string(ch)
924                         rbuff = append(rbuff,ch)
925                         inEsc = 0
926                         continue;
927                 }
928                 inEsc = 0
929             }
930             //rstr = rstr + string(ch)
931             rbuff = append(rbuff,ch)
932         }
933         //fmt.Printf("--D--subst(%s)(%s)\n",str,string(rbuff))
934         return string(rbuff)
935         //return rstr
936 }
937 func showFileInfo(path string, opts []string) {
938     if isin("-l",opts) || isin("-ls",opts) {
939         fi, err := os.Stat(path)
940         if err != nil {
941             fmt.Printf("----- ((%v))",err)
942         }else{
943             mod := fi.ModTime()
944             date := mod.Format(time.Stamp)
945             fmt.Printf("%v %8v %s ",fi.Mode(),fi.Size(),date)
946         }
947     }
948     fmt.Printf("%s",path)
949     if isin("-sp",opts) {
950         fmt.Printf(" ")
951     }else
952     if ! isin("-n",opts) {
953         fmt.Printf("\n")
954     }
955 }
956 func userHomeDir()(string,bool){
957     /*
958     homedir,_ = os.UserHomeDir() // not implemented in older Golang
959     */
960     homedir,found := os.LookupEnv("HOME")
961     //fmt.Printf("--I-- HOME=%v(%v)\n",homedir,found)
962     if !found {
963         return "/tmp",found
964     }
965     return homedir,found
966 }
967
968 func toFullpath(path string) (fullpath string) {
969     if path[0] == '/' {
970         return path
971     }
972     pathv := strings.Split(path,DIRSEP)
973     switch {
974         case pathv[0] == ".":
975             pathv[0],_ = os.Getwd()
976         case pathv[0] == ".": // all ones should be interpreted
977             cwd,_ := os.Getwd()
978             ppathv := strings.Split(cwd,DIRSEP)
979             pathv[0] = strings.Join(ppathv,DIRSEP)
980         case pathv[0] == "~":
981             pathv[0],_ = userHomeDir()
982         default:
983             cwd,_ := os.Getwd()
984             pathv[0] = cwd + DIRSEP + pathv[0]
985     }
986     return strings.Join(pathv,DIRSEP)
987 }
988
989 func IsRegFile(path string)(bool){
990     fi, err := os.Stat(path)
991     if err == nil {
992         fm := fi.Mode()

```



```

993     return fm.IsRegular();
994 }
995 return false
996 }
997
998 // <a name="encode">Encode / Decode</a>
999 // <a href="https://golang.org/pkg/encoding/base64/#example_NewEncoder">Encoder</a>
1000 func (gshCtx *GshContext)Enc(argv[]string){
1001     file := os.Stdin
1002     buff := make([]byte,LINESIZE)
1003     li := 0
1004     encoder := base64.NewEncoder(base64.StdEncoding,os.Stdout)
1005     for li = 0; ; li++ {
1006         count, err := file.Read(buff)
1007         if count <= 0 {
1008             break
1009         }
1010         if err != nil {
1011             break
1012         }
1013         encoder.Write(buff[0:count])
1014     }
1015     encoder.Close()
1016 }
1017 func (gshCtx *GshContext)Dec(argv[]string){
1018     decoder := base64.NewDecoder(base64.StdEncoding,os.Stdin)
1019     li := 0
1020     buff := make([]byte,LINESIZE)
1021     for li = 0; ; li++ {
1022         count, err := decoder.Read(buff)
1023         if count <= 0 {
1024             break
1025         }
1026         if err != nil {
1027             break
1028         }
1029         os.Stdout.Write(buff[0:count])
1030     }
1031 }
1032 // lnspl [N] [-crlf][-C \\\]
1033 func (gshCtx *GshContext)SplitLine(argv[]string){
1034     strRep := isin("-str",argv) // "..."+
1035     reader := bufio.NewReaderSize(os.Stdin,64*1024)
1036     ni := 0
1037     toi := 0
1038     for ni = 0; ; ni++ {
1039         line, err := reader.ReadString('\n')
1040         if len(line) <= 0 {
1041             if err != nil {
1042                 fmt.Fprintf(os.Stderr,"--I-- lnspl %d to %d (%v)\n",ni,toi,err)
1043                 break
1044             }
1045         }
1046         off := 0
1047         ilen := len(line)
1048         remlen := len(line)
1049         if strRep { os.Stdout.Write([]byte("\n")) }
1050         for oi := 0; 0 < remlen; oi++ {
1051             olen := remlen
1052             addnl := false
1053             if 72 < olen {
1054                 olen = 72
1055                 addnl = true
1056             }
1057             fmt.Fprintf(os.Stderr,"--D-- write %d [%d.%d] %d %d/%d/%d\n",
1058                 toi,ni,oi,off,olen,remlen,ilen)
1059             toi += 1
1060             os.Stdout.Write([]byte(line[0:olen]))
1061             if addnl {
1062                 if strRep {
1063                     os.Stdout.Write([]byte("\n\n"))
1064                 }else{
1065                     //os.Stdout.Write([]byte("\r\n"))
1066                     os.Stdout.Write([]byte("\n"))
1067                     os.Stdout.Write([]byte("\n"))
1068                 }
1069             }
1070             line = line[olen:]
1071             off += olen
1072             remlen -= olen
1073         }
1074         if strRep { os.Stdout.Write([]byte("\n\n")) }
1075     }
1076     fmt.Fprintf(os.Stderr,"--I-- lnspl %d to %d\n",ni,toi)
1077 }
1078
1079 // CRC32 <a href="http://golang.jp/pkg/hash-crc32">crc32</a>
1080 // 1 0000 0100 1100 0001 0001 1101 1011 0111
1081 var CRC32UNIX uint32 = uint32(0x04C11DB7) // Unix cksum
1082 var CRC32IEB uint32 = uint32(0xEDB88320)
1083 func byteCRC32add(crc uint32,str[]byte,len uint64)(uint32){
1084     var oi uint64
1085     for oi = 0; oi < len; oi++ {
1086         var oct = str[oi]
1087         for bi := 0; bi < 8; bi++ {
1088             //fprintf(stderr,"--CRC32 %d %X (%d.%d)\n",crc,oct,oi,bi)
1089             ovf1 := (crc & 0x80000000) != 0
1090             ovf2 := (oct & 0x80) != 0
1091             ovf := (ovf1 && !ovf2) || (!ovf1 && ovf2)
1092             oct <<= 1
1093             crc <<= 1
1094             if ovf { crc ^= CRC32UNIX }
1095         }
1096     }
1097     //fprintf(stderr,"--CRC32 return %d %d\n",crc,len)
1098     return crc;
1099 }
1100 func byteCRC32end(crc uint32, len uint64)(uint32){
1101     var slen = make([]byte,4)
1102     var li = 0
1103     for li = 0; li < 4; {
1104         slen[li] = byte(len)
1105         li += 1
1106         len >>= 8
1107         if( len == 0 ){
1108             break
1109         }
1110     }
1111     crc = byteCRC32add(crc,slen,uint64(li))
1112     crc ^= 0xFFFFFFFF
1113     return crc
1114 }
1115 func strCRC32(str string,len uint64)(crc uint32){
1116     crc = byteCRC32add(0,[]byte(str),len)

```

```

1117     crc = byteCRC32end(crc, len)
1118     //fprintf(stderr, "--CRC32 %d %d\n", crc, len)
1119     return crc
1120 }
1121 func CRC32Pinish(crc uint32, table *crc32.Table, len uint64)(uint32){
1122     var slen = make([]byte, 4)
1123     var li = 0
1124     for li = 0; li < 4; {
1125         slen[li] = byte(len & 0xFF)
1126         li += 1
1127         len >>= 8
1128         if( len == 0 ){
1129             break
1130         }
1131     }
1132     crc = crc32.Update(crc, table, slen)
1133     crc ^= 0xFFFFFFFF
1134     return crc
1135 }
1136
1137 func (gsh*GshContext)xChecksum(path string, argv[]string, sum*Checksum)(int64){
1138     if isin("-type/f", argv) && !IsRegFile(path){
1139         return 0
1140     }
1141     if isin("-type/d", argv) && IsRegFile(path){
1142         return 0
1143     }
1144     file, err := os.OpenFile(path, os.O_RDONLY, 0)
1145     if err != nil {
1146         fmt.Printf("--E-- cksum %v (%v)\n", path, err)
1147         return -1
1148     }
1149     defer file.Close()
1150     if gsh.CmdTrace { fmt.Printf("--I-- cksum %v %v\n", path, argv) }
1151
1152     bi := 0
1153     var buff = make([]byte, 32*1024)
1154     var total int64 = 0
1155     var initTime = time.Time{}
1156     if sum.Start == initTime {
1157         sum.Start = time.Now()
1158     }
1159     for bi = 0; ; bi++ {
1160         count, err := file.Read(buff)
1161         if count <= 0 || err != nil {
1162             break
1163         }
1164         if (sum.SumType & SUM_SUM64) != 0 {
1165             s := sum.Sum64
1166             for _, c := range buff[0:count] {
1167                 s += uint64(c)
1168             }
1169             sum.Sum64 = s
1170         }
1171         if (sum.SumType & SUM_UNIXFILE) != 0 {
1172             sum.Crc32Val = byteCRC32add(sum.Crc32Val, buff, uint64(count))
1173         }
1174         if (sum.SumType & SUM_CRCIEEE) != 0 {
1175             sum.Crc32Val = crc32.Update(sum.Crc32Val, &sum.Crc32Table, buff[0:count])
1176         }
1177         // <a href="https://en.wikipedia.org/wiki/BSD_checksum">BSD checksum</a>
1178         if (sum.SumType & SUM_SUM16_BSD) != 0 {
1179             s := sum.Sum16
1180             for _, c := range buff[0:count] {
1181                 s = (s >> 1) + ((s & 1) << 15)
1182                 s += int(c)
1183                 s ^= 0xFFFF
1184                 //fmt.Printf("BSDsum: %d[%d] %d\n", sum.Size+int64(i), i, s)
1185             }
1186             sum.Sum16 = s
1187         }
1188         if (sum.SumType & SUM_SUM16_SYSV) != 0 {
1189             for bj := 0; bj < count; bj++ {
1190                 sum.Sum16 += int(buff[bj])
1191             }
1192         }
1193         total += int64(count)
1194     }
1195     sum.Done = time.Now()
1196     sum.Files += 1
1197     sum.Size += total
1198     if !isin("-s", argv) {
1199         fmt.Printf("%v ", total)
1200     }
1201     return 0
1202 }
1203
1204 // <a name="grep">grep</a>
1205 // "lines", "lin" or "lmp" for "(text) line processor" or "scanner"
1206 // a*, lab, c, ... sequential combination of patterns
1207 // what "LINE" is should be definable
1208 // generic line-by-line processing
1209 // grep [-v]
1210 // cat -n -v
1211 // uniq [-c]
1212 // tail -f
1213 // sed s/x/y/ or awk
1214 // grep with line count like wc
1215 // rewrite contents if specified
1216 func (gsh*GshContext)xGrep(path string, rexp[]string)(int){
1217     file, err := os.OpenFile(path, os.O_RDONLY, 0)
1218     if err != nil {
1219         fmt.Printf("--E-- grep %v (%v)\n", path, err)
1220         return -1
1221     }
1222     defer file.Close()
1223     if gsh.CmdTrace { fmt.Printf("--I-- grep %v %v\n", path, rexp) }
1224     //reader := bufio.NewReaderSize(file, LINE_SIZE)
1225     reader := bufio.NewReaderSize(file, 80)
1226     li := 0
1227     found := 0
1228     for li = 0; ; li++ {
1229         line, err := reader.ReadString('\n')
1230         if len(line) <= 0 {
1231             break
1232         }
1233         if 150 < len(line) {
1234             // maybe binary
1235             break;
1236         }
1237         if err != nil {
1238             break
1239         }
1240         if 0 <= strings.Index(string(line), rexp[0]) {

```

```

1241     found += 1
1242     fmt.Printf("%s:%d: %s", path, li, line)
1243     }
1244 }
1245 //fmt.Printf("total %d lines %s\n", li, path)
1246 //if( 0 < found ){ fmt.Printf("((found %d lines %s))\n", found, path); }
1247 return found
1248 }
1249
1250 // <a name="finder">Finder</a>
1251 // finding files with it name and contents
1252 // file names are ORed
1253 // show the content with %x fmt list
1254 // ls -R
1255 // tar command by adding output
1256 type fileSum struct {
1257     Err int64 // access error or so
1258     Size int64 // content size
1259     DupSize int64 // content size from hard links
1260     Blocks int64 // number of blocks (of 512 bytes)
1261     DupBlocks int64 // Blocks pointed from hard links
1262     HLinks int64 // hard links
1263     Words int64
1264     Lines int64
1265     Files int64
1266     Dirs int64 // the num. of directories
1267     SymLink int64
1268     Flats int64 // the num. of flat files
1269     MaxDepth int64
1270     MaxNamlen int64 // max. name length
1271     nextRepo time.Time
1272 }
1273 func showFusage(dir string, fusage *fileSum) {
1274     bsume := float64(((fusage.Blocks - fusage.DupBlocks) / 2) * 1024) / 1000000.0
1275     // bsumdup := float64((fusage.Blocks / 2) * 1024) / 1000000.0
1276
1277     fmt.Printf("%v: %v files (%vd %vs %vh) %.6f MB (%.2f MBK)\n",
1278         dir,
1279         fusage.Files,
1280         fusage.Dirs,
1281         fusage.SymLink,
1282         fusage.HLinks,
1283         float64(fusage.Size) / 1000000.0, bsume);
1284 }
1285 const (
1286     S_IFMT = 0170000
1287     S_IFCHR = 0020000
1288     S_IFDIR = 0040000
1289     S_IFREG = 0100000
1290     S_IFLNK = 0120000
1291     S_IFSOCK = 0140000
1292 )
1293 func cumFinfo(fsum *fileSum, path string, staterr error, fstat syscall.Stat_t, argv []string, verb bool) (*fileSum) {
1294     now := time.Now()
1295     if time.Second <= now.Sub(fsum.nextRepo) {
1296         if !fsum.nextRepo.IsZero() {
1297             tstamp := now.Format(time.Stamp)
1298             showFusage(tstamp, fsum)
1299         }
1300         fsum.nextRepo = now.Add(time.Second)
1301     }
1302     if staterr != nil {
1303         fsum.Err += 1
1304         return fsum
1305     }
1306     fsum.Files += 1
1307     if l < fstat.Nlink {
1308         // must count only once...
1309         // at least ignore ones in the same directory
1310         //if finfo.Mode().IsRegular() {
1311             if (fstat.Mode & S_IFMT) == S_IFREG {
1312                 fsum.HLinks += 1
1313                 fsum.DupBlocks += int64(fstat.Blocks)
1314                 //fmt.Printf("---Dup HardLink %v %s\n", fstat.Nlink, path)
1315             }
1316         }
1317         //fsum.Size += finfo.Size()
1318         fsum.Size += fstat.Size
1319         fsum.Blocks += int64(fstat.Blocks)
1320         //if verb { fmt.Printf(" (%8dBk) %s", fstat.Blocks/2, path) }
1321         if isin("-ls", argv) {
1322             //if verb { fmt.Printf("%4d %8d ", fstat.Blksize, fstat.Blocks) }
1323             //fmt.Printf("%d\t", fstat.Blocks/2)
1324         }
1325         //if finfo.IsDir()
1326         if (fstat.Mode & S_IFMT) == S_IFDIR {
1327             fsum.Dirs += 1
1328         }
1329         //if (finfo.Mode() & os.ModeSymlink) != 0
1330         if (fstat.Mode & S_IFMT) == S_IFLNK {
1331             //if verb { fmt.Printf("symlink(%v,%s)\n", fstat.Mode, finfo.Name()) }
1332             //if verb { fmt.Printf("symlink(%o,%s)\n", fstat.Mode, finfo.Name()) }
1333             fsum.SymLink += 1
1334         }
1335     }
1336     return fsum
1337 }
1338 func (gsh *GshContext) xxFindEntv(depth int, total *fileSum, dir string, dstat syscall.Stat_t, ei int, env []string, npatv []string, argv []string) (*fileSum) {
1339     nols := isin("-grep", argv)
1340     // sort env
1341     /*
1342     if isin("-t", argv) {
1343         sort.Slice(filev, func(i, j int) bool {
1344             return 0 < filev[i].ModTime().Sub(filev[j].ModTime())
1345         })
1346     }
1347     */
1348     if isin("-u", argv) {
1349         sort.Slice(filev, func(i, j int) bool {
1350             return 0 < filev[i].AccTime().Sub(filev[j].AccTime())
1351         })
1352     }
1353     if isin("-U", argv) {
1354         sort.Slice(filev, func(i, j int) bool {
1355             return 0 < filev[i].CreateTime().Sub(filev[j].CreateTime())
1356         })
1357     }
1358     /*
1359     if isin("-S", argv) {
1360         sort.Slice(filev, func(i, j int) bool {
1361             return filev[j].Size() < filev[i].Size()
1362         })
1363     }
1364     */

```

```

1365  */
1366  for _, filename := range entv {
1367      for _, npat := range npatv {
1368          match := true
1369          if npat == "*" {
1370              match = true
1371          } else {
1372              match, _ = filepath.Match(npat, filename)
1373          }
1374          path := dir + DIRSEP + filename
1375          if !match {
1376              continue
1377          }
1378          var fstat syscall.Stat_t
1379          staterr := syscall.Lstat(path, &fstat)
1380          if staterr != nil {
1381              if !isin("-w", argv) { fmt.Printf("ufind: %v\n", staterr) }
1382              continue;
1383          }
1384          if isin("-du", argv) && (fstat.Mode & S_IFMT) == S_IFDIR {
1385              // should not show size of directory in "-du" mode ...
1386          } else {
1387              if !nols && !isin("-s", argv) && (!isin("-du", argv) || isin("-a", argv)) {
1388                  if isin("-du", argv) {
1389                      fmt.Printf("%d\t", fstat.Blocks/2)
1390                  }
1391                  showFileInfo(path, argv)
1392              }
1393              if true { // && isin("-du", argv)
1394                  total = cumFinfo(total, path, staterr, fstat, argv, false)
1395              }
1396              /*
1397              if isin("-wc", argv) {
1398              }
1399              */
1400              if gsh.lastCheckSum.SumType != 0 {
1401                  gsh.xCksum(path, argv, &gsh.lastCheckSum);
1402              }
1403              x := isinX("-grep", argv); // -grep will be convenient like -ls
1404              if 0 < x && x+1 <= len(argv) { // -grep will be convenient like -ls
1405                  if IsRegFile(path) {
1406                      found := gsh.xGrep(path, argv[x+1:])
1407                      if 0 < found {
1408                          foundv := gsh.CmdCurrent.FoundFile
1409                          if len(foundv) < 10 {
1410                              gsh.CmdCurrent.FoundFile =
1411                                  append(gsh.CmdCurrent.FoundFile, path)
1412                          }
1413                      }
1414                  }
1415              }
1416              if !isin("-r0", argv) { // -d 0 in du, -depth n in find
1417                  //total.Depth += 1
1418                  if (fstat.Mode & S_IFMT) == S_IFLNK {
1419                      continue
1420                  }
1421                  if dstat.Rdev != fstat.Rdev {
1422                      fmt.Printf("--I-- don't follow differnet device %v(%v) %v(%v)\n",
1423                          dir, dstat.Rdev, path, fstat.Rdev)
1424                  }
1425                  if (fstat.Mode & S_IFMT) == S_IFDIR {
1426                      total = gsh.xxFind(depth+1, total, path, npatv, argv)
1427                  }
1428              }
1429          }
1430      }
1431      return total
1432  }
1433  func (gsh*GshContext)xxFind(depth int, total *fileSum, dir string, npatv []string, argv []string) (*fileSum) {
1434      nols := isin("-grep", argv)
1435      dirfile, oerr := os.OpenFile(dir, os.O_RDONLY, 0)
1436      if oerr == nil {
1437          //fmt.Printf("--I-- %v(%v) [%d]\n", dir, dirfile, dirfile.Fd())
1438          defer dirfile.Close()
1439      } else {
1440      }
1441      prev := *total
1442      var dstat syscall.Stat_t
1443      staterr := syscall.Lstat(dir, &dstat) // should be flstat
1444      if staterr != nil {
1445          if !isin("-w", argv) { fmt.Printf("ufind: %v\n", staterr) }
1446          return total
1447      }
1448      //filev, err := ioutil.ReadDir(dir)
1449      //_, err := ioutil.ReadDir(dir) // ReadDir() heavy and bad for huge directory
1450      /*
1451      if err != nil {
1452          if !isin("-w", argv) { fmt.Printf("ufind: %v\n", err) }
1453          return total
1454      }
1455      */
1456      /*
1457      if depth == 0 {
1458          total = cumFinfo(total, dir, staterr, dstat, argv, true)
1459          if !nols && !isin("-s", argv) && (!isin("-du", argv) || isin("-a", argv)) {
1460              showFileInfo(dir, argv)
1461          }
1462      }
1463      // it it is not a directory, just scan it and finish
1464      for ei := 0; ; ei++ {
1465          entv, rderr := dirfile.Readdirnames(8*1024)
1466          if len(entv) == 0 || rderr != nil {
1467              //if rderr != nil { fmt.Printf("[%d] len=%d (%v)\n", ei, len(entv), rderr) }
1468              break
1469          }
1470          if 0 < ei {
1471              fmt.Printf("--I-- xxFind[%d] %d large-dir: %s\n", ei, len(entv), dir)
1472          }
1473          total = gsh.xxFindEntv(depth, total, dir, dstat, ei, entv, npatv, argv)
1474      }
1475      if isin("-du", argv) {
1476          // if in "du" mode
1477          fmt.Printf("%d\t%s\n", (total.Blocks-prev.Blocks)/2, dir)
1478      }
1479      return total
1480  }
1481  }
1482  }
1483  // {ufind|fu|ls} [Files] [// Names] [-- Expressions]
1484  // Files is "." by default
1485  // Names is "*" by default
1486  // Expressions is "--print" by default for "ufind", or -du for "fu" command
1487  func (gsh*GshContext)xxFind(argv []string) {

```

```

1489 if 0 < len(argv) && strBegins(argv[0],""){
1490     showFound(gsh,argv)
1491     return
1492 }
1493 if isin("-cksum",argv) || isin("-sum",argv) {
1494     gsh.lastCheckSum = CheckSum{
1495         if isin("-sum",argv) && isin("-add",argv) {
1496             gsh.lastCheckSum.SumType |= SUM_SUM64
1497         }else
1498         if isin("-sum",argv) && isin("-size",argv) {
1499             gsh.lastCheckSum.SumType |= SUM_SIZE
1500         }else
1501         if isin("-sum",argv) && isin("-bsd",argv) {
1502             gsh.lastCheckSum.SumType |= SUM_SUM16_BSD
1503         }else
1504         if isin("-sum",argv) && isin("-sysv",argv) {
1505             gsh.lastCheckSum.SumType |= SUM_SUM16_SYSV
1506         }else
1507         if isin("-sum",argv) {
1508             gsh.lastCheckSum.SumType |= SUM_SUM64
1509         }
1510         if isin("-unix",argv) {
1511             gsh.lastCheckSum.SumType |= SUM_UNIXFILE
1512             gsh.lastCheckSum.Crc32Table = *crc32.MakeTable(CRC32UNIX)
1513         }
1514         if isin("-ieee",argv){
1515             gsh.lastCheckSum.SumType |= SUM_CRCIEEE
1516             gsh.lastCheckSum.Crc32Table = *crc32.MakeTable(CRC32IEEE)
1517         }
1518         gsh.lastCheckSum.RusgAtStart = Getrusagev()
1519     }
1520     var total = fileSum{
1521         npats := []string{}
1522         for _,v := range argv {
1523             if 0 < len(v) && v[0] != '-' {
1524                 npats = append(npats,v)
1525             }
1526             if v == "/" { break }
1527             if v == "-" { break }
1528             if v == "-grep" { break }
1529             if v == "-ls" { break }
1530         }
1531         if len(npats) == 0 {
1532             npats = []string{"*"}
1533         }
1534         cwd := "."
1535         // if to be fullpath :: cwd, _ := os.Getwd()
1536         if len(npats) == 0 { npats = []string{"*"} }
1537         fusage := gsh.xxFind(0,&total,cwd,npats,argv)
1538         if gsh.lastCheckSum.SumType != 0 {
1539             var sumi uint64 = 0
1540             sum := &gsh.lastCheckSum
1541             if (sum.SumType & SUM_SIZE) != 0 {
1542                 sumi = uint64(sum.Size)
1543             }
1544             if (sum.SumType & SUM_SUM64) != 0 {
1545                 sumi = sum.Sum64
1546             }
1547             if (sum.SumType & SUM_SUM16_SYSV) != 0 {
1548                 s := uint32(sum.Sum16)
1549                 r := (s & 0xFFFF) + ((s & 0xFFFFFFF) >> 16)
1550                 s = (r & 0xFFFF) + (r >> 16)
1551                 sum.Crc32Val = uint32(s)
1552                 sumi = uint64(s)
1553             }
1554             if (sum.SumType & SUM_SUM16_BSD) != 0 {
1555                 sum.Crc32Val = uint32(sum.Sum16)
1556                 sumi = uint64(sum.Sum16)
1557             }
1558             if (sum.SumType & SUM_UNIXFILE) != 0 {
1559                 sum.Crc32Val = byteCRC32end(sum.Crc32Val,uint64(sum.Size))
1560                 sumi = uint64(byteCRC32end(sum.Crc32Val,uint64(sum.Size)))
1561             }
1562             if 1 < sum.Files {
1563                 fmt.Printf("%v %v // %v / %v files, %v/file\r\n",
1564                     sumi,sum.Size,
1565                     abssize(sum.Size),sum.Files,
1566                     abssize(sum.Size/sum.Files))
1567             }else{
1568                 fmt.Printf("%v %v %v\n",
1569                     sumi,sum.Size,npats[0])
1570             }
1571         }
1572         if !isin("-grep",argv) {
1573             showFusage("total",fusage)
1574         }
1575         if !isin("-s",argv){
1576             hits := len(gsh.CmdCurrent.FoundFile)
1577             if 0 < hits {
1578                 fmt.Printf("--I-- %d files hits // can be refered with !%df\n",
1579                     hits,len(gsh.CommandHistory))
1580             }
1581         }
1582         if gsh.lastCheckSum.SumType != 0 {
1583             if isin("-ru",argv) {
1584                 sum := &gsh.lastCheckSum
1585                 sum.Done = time.Now()
1586                 gsh.lastCheckSum.RusgAtEnd = Getrusagev()
1587                 elps := sum.Done.Sub(sum.Start)
1588                 fmt.Printf("--cksum-size: %v (%v) / %v files, %v/file\r\n",
1589                     sum.Size,abssize(sum.Size),sum.Files,abssize(sum.Size/sum.Files))
1590                 nanos := int64(elps)
1591                 fmt.Printf("--cksum-time: %v/total, %v/file, %1.f files/s, %v\r\n",
1592                     abftime(nanos),
1593                     abftime(nanos/sum.Files),
1594                     (float64(sum.Files)*1000000000.0)/float64(nanos),
1595                     abbspeed(sum.Size,nanos) )
1596                 diff := RusageSubv(sum.RusgAtEnd, sum.RusgAtStart)
1597                 fmt.Printf("--cksum-rusg: %v\n",sRusagef("",argv,diff))
1598             }
1599         }
1600         return
1601     }
1602 }
1603 func showFiles(files[]string){
1604     sp := ""
1605     for i,file := range files {
1606         if 0 < i { sp = " " } else { sp = "" }
1607         fmt.Printf(sp+"%s",escapeWhiteSP(file))
1608     }
1609 }
1610 func showFound(gshCtx *GshContext, argv[]string){
1611     for i,v := range gshCtx.CommandHistory {
1612         if 0 < len(v.FoundFile) {

```

```

1613         fmt.Printf("!%d (%d) ", i, len(v.FoundFile))
1614         if isin("-ls", argv){
1615             fmt.Printf("\n")
1616             for _, file := range v.FoundFile {
1617                 fmt.Printf("%s" //sub number?
1618                     showFileInfo(file, argv)
1619             }
1620         }else{
1621             showFiles(v.FoundFile)
1622             fmt.Printf("\n")
1623         }
1624     }
1625 }
1626 }
1627
1628 func showMatchFile(filev []os.FileInfo, npat, dir string, argv[]string)(string, bool){
1629     fname := ""
1630     found := false
1631     for _, v := range filev {
1632         match, _ := filepath.Match(npat, (v.Name()))
1633         if match {
1634             fname = v.Name()
1635             found = true
1636             //fmt.Printf("[%d] %s\n", i, v.Name())
1637             showIfExecutable(fname, dir, argv)
1638         }
1639     }
1640     return fname, found
1641 }
1642 func showIfExecutable(name, dir string, argv[]string)(ffullpath string, ffound bool){
1643     var fullpath string
1644     if strBegins(name, DIRSEP){
1645         fullpath = name
1646     }else{
1647         fullpath = dir + DIRSEP + name
1648     }
1649     fi, err := os.Stat(fullpath)
1650     if err != nil {
1651         fullpath = dir + DIRSEP + name + ".go"
1652         fi, err = os.Stat(fullpath)
1653     }
1654     if err == nil {
1655         fm := fi.Mode()
1656         if fm.IsRegular() {
1657             // R_OK=4, W_OK=2, X_OK=1, F_OK=0
1658             if syscall.Access(fullpath, 5) == nil {
1659                 ffullpath = fullpath
1660                 ffound = true
1661                 if ! isin("-s", argv) {
1662                     showFileInfo(fullpath, argv)
1663                 }
1664             }
1665         }
1666     }
1667     return ffullpath, ffound
1668 }
1669 func which(list string, argv []string) (fullpathv []string, itis bool){
1670     if len(argv) <= 1 {
1671         fmt.Printf("Usage: which comand [-s] [-a] [-ls]\n")
1672         return []string(""), false
1673     }
1674     path := argv[1]
1675     if strBegins(path, "/") {
1676         // should check if excoecutable?
1677         _, exOK := showIfExecutable(path, "/", argv)
1678         fmt.Printf("--D-- %v exOK=%v\n", path, exOK)
1679         return []string(path), exOK
1680     }
1681     pathenv, efound := os.LookupEnv(list)
1682     if ! efound {
1683         fmt.Printf("--E-- which: no \"%s\" environment\n", list)
1684         return []string(""), false
1685     }
1686     showall := isin("-a", argv) || 0 <= strings.Index(path, "+")
1687     dirv := strings.Split(pathenv, PATHSEP)
1688     ffound := false
1689     ffullpath := path
1690     for _, dir := range dirv {
1691         if 0 <= strings.Index(path, "*") { // by wild-card
1692             list, _ := ioutil.ReadDir(dir)
1693             ffullpath, ffound = showMatchFile(list, path, dir, argv)
1694         }else{
1695             ffullpath, ffound = showIfExecutable(path, dir, argv)
1696         }
1697         //if ffound && !isin("-a", argv) {
1698         if ffound && !showall {
1699             break;
1700         }
1701     }
1702     return []string(ffullpath), ffound
1703 }
1704
1705 func stripLeadingWSParg(argv[]string)([]string){
1706     for ; 0 < len(argv); {
1707         if len(argv[0]) == 0 {
1708             argv = argv[1:]
1709         }else{
1710             break
1711         }
1712     }
1713     return argv
1714 }
1715 func xEval(argv []string, nlend bool){
1716     argv = stripLeadingWSParg(argv)
1717     if len(argv) == 0 {
1718         fmt.Printf("eval [%%format] [Go-expression]\n")
1719         return
1720     }
1721     pfmt := "%v"
1722     if argv[0][0] == '%' {
1723         pfmt = argv[0]
1724         argv = argv[1:]
1725     }
1726     if len(argv) == 0 {
1727         return
1728     }
1729     gocode := strings.Join(argv, " ");
1730     //fmt.Printf("eval [%v] [%v]\n", pfmt, gocode)
1731     fset := token.NewFileSet()
1732     rval, _ := types.Eval(fset, nil, token.NoPos, gocode)
1733     fmt.Printf(pfmt, rval.Value)
1734     if nlend { fmt.Printf("\n") }
1735 }
1736

```

```

1737 func getval(name string) (found bool, val int) {
1738     /* should expand the name here */
1739     if name == "gsh.pid" {
1740         return true, os.Getpid()
1741     }else
1742     if name == "gsh.ppid" {
1743         return true, os.Getppid()
1744     }
1745     return false, 0
1746 }
1747
1748 func echo(argv []string, nlen bool){
1749     for ai := 1; ai < len(argv); ai++ {
1750         if 1 < ai {
1751             fmt.Printf(" ");
1752         }
1753         arg := argv[ai]
1754         found, val := getval(arg)
1755         if found {
1756             fmt.Printf("%d",val)
1757         }else{
1758             fmt.Printf("%s",arg)
1759         }
1760     }
1761     if nlen {
1762         fmt.Printf("\n");
1763     }
1764 }
1765
1766 func resfile() string {
1767     return "gsh.tmp"
1768 }
1769 //var resF *File
1770 func resmap() {
1771     //_, err := os.OpenFile(resfile(), os.O_RDWR|os.O_CREATE, os.ModeAppend)
1772     // https://develloppaper.com/solution-to-golang-bad-file-descriptor-problem/
1773     _, err := os.OpenFile(resfile(), os.O_RDWR|os.O_CREATE, 0600)
1774     if err != nil {
1775         fmt.Printf("refF could not open: %s\n",err)
1776     }else{
1777         fmt.Printf("refF opened\n")
1778     }
1779 }
1780
1781 // @@2020-0821
1782 func gshScanArg(str string,strip int)(argv []string){
1783     var si = 0
1784     var sb = 0
1785     var inBracket = 0
1786     var arg1 = make([]byte,LINESIZE)
1787     var ax = 0
1788     debug := false
1789
1790     for ; si < len(str); si++ {
1791         if str[si] != ' ' {
1792             break
1793         }
1794     }
1795     sb = si
1796     for ; si < len(str); si++ {
1797         if sb <= si {
1798             if debug {
1799                 fmt.Printf("--Da- +%d %2d-%2d %s ... %s\n",
1800                     inBracket,sb,si,arg1[0:ax],str[si:])
1801             }
1802         }
1803         ch := str[si]
1804         if ch == '{' {
1805             inBracket += 1
1806             if 0 < strip && inBracket <= strip {
1807                 //fmt.Printf("stripLEV %d <= %d?\n",inBracket,strip)
1808                 continue
1809             }
1810         }
1811         if 0 < inBracket {
1812             if ch == '}' {
1813                 inBracket -= 1
1814                 if 0 < strip && inBracket < strip {
1815                     //fmt.Printf("stripLEV %d < %d?\n",inBracket,strip)
1816                     continue
1817                 }
1818             }
1819             arg1[ax] = ch
1820             ax += 1
1821             continue
1822         }
1823         if str[si] == ' ' {
1824             argv = append(argv,string(arg1[0:ax]))
1825             if debug {
1826                 fmt.Printf("--Da- [%v][%v-%v] %s ... %s\n",
1827                     -1+len(argv),sb,si,str[sb:si],string(str[si:]))
1828             }
1829             sb = si+1
1830             ax = 0
1831             continue
1832         }
1833         arg1[ax] = ch
1834         ax += 1
1835     }
1836     if sb < si {
1837         argv = append(argv,string(arg1[0:ax]))
1838         if debug {
1839             fmt.Printf("--Da- [%v][%v-%v] %s ... %s\n",
1840                 -1+len(argv),sb,si,string(arg1[0:ax]),string(str[si:]))
1841         }
1842     }
1843     if debug {
1844         fmt.Printf("--Da- %d [%s] => [%d]\v\n",strip,str,len(argv),argv)
1845     }
1846     return argv
1847 }
1848
1849 // should get stderr (into tmpfile ?) and return
1850 func (gsh*GshContext)Popen(name,mode string)(pin*os.File,pout*os.File,err bool){
1851     var pv = []int{-1,-1}
1852     syscall.Pipe(pv)
1853
1854     xarg := gshScanArg(name,1)
1855     name = strings.Join(xarg," ")
1856
1857     pin = os.NewFile(uintptr(pv[0]),"StdoutOf-"+name+"")
1858     pout = os.NewFile(uintptr(pv[1]),"StdinOf-"+name+"")
1859     fdix := 0
1860     dir := "?"

```

```

1861 if mode == "r" {
1862     dir = "<"
1863     fdix = 1 // read from the stdout of the process
1864 }else{
1865     dir = ">"
1866     fdix = 0 // write to the stdin of the process
1867 }
1868 gshPA := gsh.gshPA
1869 savfd := gshPA.Files[fdix]
1870
1871 var fd uintptr = 0
1872 if mode == "r" {
1873     fd = pout.Fd()
1874     gshPA.Files[fdix] = pout.Fd()
1875 }else{
1876     fd = pin.Fd()
1877     gshPA.Files[fdix] = pin.Fd()
1878 }
1879 // should do this by Goroutine?
1880 if false {
1881     fmt.Printf("--Ip- Opened fd[%v] %s %v\n",fd,dir,name)
1882     fmt.Printf("--RDL [%d,%d,%d]->[%d,%d,%d]\n",
1883         os.Stdin.Fd(),os.Stdout.Fd(),os.Stderr.Fd(),
1884         pin.Fd(),pout.Fd(),pout.Fd())
1885 }
1886 savi := os.Stdin
1887 savo := os.Stdout
1888 save := os.Stderr
1889 os.Stdin = pin
1890 os.Stdout = pout
1891 os.Stderr = pout
1892 gsh.BackGround = true
1893 gsh.gshellh(name)
1894 gsh.BackGround = false
1895 os.Stdin = savi
1896 os.Stdout = savo
1897 os.Stderr = save
1898
1899 gshPA.Files[fdix] = savfd
1900 return pin,pout,false
1901 }
1902
1903 // <a name="ex-commands">External commands</a>
1904 func (gsh*GshContext)excommand(exec bool, argv []string) (notif bool,exit bool) {
1905     if gsh.CmdTrace { fmt.Printf("--I-- excommand[%v](%v)\n",exec,argv) }
1906
1907     gshPA := gsh.gshPA
1908     fullpathv, itis := which("PATH",[]string{"which",argv[0],"-s"})
1909     if itis == false {
1910         return true,false
1911     }
1912     fullpath := fullpathv[0]
1913     argv = unescapeWhiteSPV(argv)
1914     if 0 < strings.Index(fullpath, ".go") {
1915         nargv := argv // []string{}
1916         gofullpathv, itis := which("PATH",[]string{"which", "go", "-s"})
1917         if itis == false {
1918             fmt.Printf("--F-- Go not found\n")
1919             return false,true
1920         }
1921         gofullpath := gofullpathv[0]
1922         nargv = []string{ gofullpath, "run", fullpath }
1923         fmt.Printf("--I-- %s {%s %s %s}\n",gofullpath,
1924             nargv[0],nargv[1],nargv[2])
1925         if exec {
1926             syscall.Exec(gofullpath,nargv,os.Environ())
1927         }else{
1928             pid, _ := syscall.ForkExec(gofullpath,nargv,&gshPA)
1929             if gsh.BackGround {
1930                 fmt.Fprintf(stderr, "--Ip- in Background pid[%d](%v)\n",pid,len(argv),nargv)
1931                 gsh.BackGroundJobs = append(gsh.BackGroundJobs,pid)
1932             }else{
1933                 rusage := syscall.Rusage {}
1934                 syscall.Wait4(pid,nil,0,&rusage)
1935                 gsh.LastRusage = rusage
1936                 gsh.CmdCurrent.Rusagev[1] = rusage
1937             }
1938         }
1939     }else{
1940         if exec {
1941             syscall.Exec(fullpath,argv,os.Environ())
1942         }else{
1943             pid, _ := syscall.ForkExec(fullpath,argv,&gshPA)
1944             //fmt.Printf("[%d]\n",pid); // '&' to be background
1945             if gsh.BackGround {
1946                 fmt.Fprintf(stderr, "--Ip- in Background pid[%d](%v)\n",pid,len(argv),argv)
1947                 gsh.BackGroundJobs = append(gsh.BackGroundJobs,pid)
1948             }else{
1949                 rusage := syscall.Rusage {}
1950                 syscall.Wait4(pid,nil,0,&rusage);
1951                 gsh.LastRusage = rusage
1952                 gsh.CmdCurrent.Rusagev[1] = rusage
1953             }
1954         }
1955     }
1956     return false,false
1957 }
1958
1959 // <a name="builtin">Builtin Commands</a>
1960 func (gshCtx *GshContext) sleep(argv []string) {
1961     if len(argv) < 2 {
1962         fmt.Printf("Sleep 100ms, 100us, 100ns, ... \n")
1963         return
1964     }
1965     duration := argv[1];
1966     d, err := time.ParseDuration(duration)
1967     if err != nil {
1968         d, err = time.ParseDuration(duration+"s")
1969         if err != nil {
1970             fmt.Printf("duration ? %s (%s)\n",duration,err)
1971             return
1972         }
1973     }
1974     //fmt.Printf("Sleep %v\n",duration)
1975     time.Sleep(d)
1976     if 0 < len(argv[2:]) {
1977         gshCtx.gshellv(argv[2:])
1978     }
1979 }
1980 func (gshCtx *GshContext)repeat(argv []string) {
1981     if len(argv) < 2 {
1982         return
1983     }
1984     start0 := time.Now()

```



```

1985 for ri, _ := strconv.Atoi(argv[1]); 0 < ri; ri-- {
1986     if 0 < len(argv[2]) {
1987         //start := time.Now()
1988         gshCtx.gshelly(argv[2])
1989         end := time.Now()
1990         elps := end.Sub(start0);
1991         if( 100000000 < elps ){
1992             fmt.Printf("(repeat#%d %v)\n",ri,elps);
1993         }
1994     }
1995 }
1996 }
1997
1998 func (gshCtx *GshContext)gen(argv []string) {
1999     gshPA := gshCtx.gshPA
2000     if len(argv) < 2 {
2001         fmt.Printf("Usage: %s N\n",argv[0])
2002         return
2003     }
2004     // should br repeated by "repeat" command
2005     count, _ := strconv.Atoi(argv[1])
2006     fd := gshPA.Files[1] // Stdout
2007     file := os.NewFile(fd,"internalStdOut")
2008     fmt.Printf("--I-- Gen. Count=%d to [%d]\n",count,file.Fd())
2009     //buf := []byte{}
2010     outdata := "0123 5678 0123 5678 0123 5678 0123 5678\r"
2011     for gi := 0; gi < count; gi++ {
2012         file.WriteString(outdata)
2013     }
2014     //file.WriteString("\n")
2015     fmt.Printf("\n(%d B)\n",count*len(outdata));
2016     //file.Close()
2017 }
2018
2019 // <a name="rexec">Remote Execution</a> // 2020-0820
2020 func Elapsed(from time.Time)(string){
2021     elps := time.Now().Sub(from)
2022     if 100000000 < elps {
2023         return fmt.Sprintf("[%5d.%02ds]",elps/100000000,(elps%100000000)/1000000)
2024     }else
2025     if 1000000 < elps {
2026         return fmt.Sprintf("[%3d.%03dms]",elps/1000000,(elps%1000000)/1000)
2027     }else{
2028         return fmt.Sprintf("[%3d.%03dus]",elps/1000,(elps%1000))
2029     }
2030 }
2031 func abftime(nanos int64)(string){
2032     if 100000000 < nanos {
2033         return fmt.Sprintf("%d.%02ds",nanos/100000000,(nanos%100000000)/1000000)
2034     }else
2035     if 1000000 < nanos {
2036         return fmt.Sprintf("%d.%03dms",nanos/1000000,(nanos%1000000)/1000)
2037     }else{
2038         return fmt.Sprintf("%d.%03dus",nanos/1000,(nanos%1000))
2039     }
2040 }
2041 func abszize(size int64)(string){
2042     fsize := float64(size)
2043     if 1024*1024*1024 < size {
2044         return fmt.Sprintf("%.2fGiB",fsize/(1024*1024*1024))
2045     }else
2046     if 1024*1024 < size {
2047         return fmt.Sprintf("%.3fMiB",fsize/(1024*1024))
2048     }else{
2049         return fmt.Sprintf("%.3fKiB",fsize/1024)
2050     }
2051 }
2052 func abszize(size int64)(string){
2053     fsize := float64(size)
2054     if 1024*1024*1024 < size {
2055         return fmt.Sprintf("%.2fGiB",fsize/(1024*1024*1024))
2056     }else
2057     if 1024*1024 < size {
2058         return fmt.Sprintf("%.3fMiB",fsize/(1024*1024))
2059     }else{
2060         return fmt.Sprintf("%.3fKiB",fsize/1024)
2061     }
2062 }
2063 func abbspd(totalB int64,ns int64)(string){
2064     MBs := (float64(totalB)/1000000) / (float64(ns)/1000000000)
2065     if 1000 <= MBs {
2066         return fmt.Sprintf("%6.3fGB/s",MBs/1000)
2067     }
2068     if 1 <= MBs {
2069         return fmt.Sprintf("%6.3fMB/s",MBs)
2070     }else{
2071         return fmt.Sprintf("%6.3fKB/s",MBs*1000)
2072     }
2073 }
2074 func abbspd(totalB int64,ns time.Duration)(string){
2075     MBs := (float64(totalB)/1000000) / (float64(ns)/1000000000)
2076     if 1000 <= MBs {
2077         return fmt.Sprintf("%6.3fGBps",MBs/1000)
2078     }
2079     if 1 <= MBs {
2080         return fmt.Sprintf("%6.3fMBps",MBs)
2081     }else{
2082         return fmt.Sprintf("%6.3fKBps",MBs*1000)
2083     }
2084 }
2085 func fileRelay(what string,in*os.File,out*os.File,size int64,bsiz int)(wcount int64){
2086     Start := time.Now()
2087     buff := make([]byte,bsiz)
2088     var total int64 = 0
2089     var rem int64 = size
2090     nio := 0
2091     Prev := time.Now()
2092     var PrevSize int64 = 0
2093
2094     fmt.Printf(Elapsed(Start)+"--In- X: %s (%v/%v/%v) START\n",
2095         what,abszize(total),size,nio)
2096
2097     for i:= 0; ; i++ {
2098         var len = bsiz
2099         if int(rem) < len {
2100             len = int(rem)
2101         }
2102         Now := time.Now()
2103         Elps := Now.Sub(Prev);
2104         if 100000000 < Now.Sub(Prev) {
2105             fmt.Printf(Elapsed(Start)+"--In- X: %s (%v/%v/%v) %s\n",
2106                 what,abszize(total),size,nio,
2107                 abbspd((total-PrevSize),Elps))
2108             Prev = Now;

```

```

2109     PrevSize = total
2110 }
2111 rlen := len
2112 if in != nil {
2113     // should watch the disconnection of out
2114     rcc,err := in.Read(buff[0:rlen])
2115     if err != nil {
2116         fmt.Printf(Elapsed(Start)+"--En- X: %s read(%v,%v)<%v\n",
2117             what,rcc,err,in.Name())
2118         break
2119     }
2120     rlen = rcc
2121     if string(buff[0:10]) == "(SoftEOF " {
2122         var ecc int64 = 0
2123         fmt.Sscanf(string(buff),"(SoftEOF %v",&ecc)
2124         fmt.Printf(Elapsed(Start)+"--En- X: %s Recv ((SoftEOF %v))/%v\n",
2125             what,ecc,total)
2126         if ecc == total {
2127             break
2128         }
2129     }
2130 }
2131
2132 wlen := rlen
2133 if out != nil {
2134     wcc,err := out.Write(buff[0:rlen])
2135     if err != nil {
2136         fmt.Printf(Elapsed(Start)+"--En-- X: %s write(%v,%v)>%v\n",
2137             what,wcc,err,out.Name())
2138         break
2139     }
2140     wlen = wcc
2141 }
2142 if wlen < rlen {
2143     fmt.Printf(Elapsed(Start)+"--En- X: %s incomplete write (%v/%v)\n",
2144         what,wlen,rlen)
2145     break;
2146 }
2147
2148 nio += 1
2149 total += int64(rlen)
2150 rem -= int64(rlen)
2151 if rem <= 0 {
2152     break
2153 }
2154 }
2155 Done := time.Now()
2156 Elps := float64(Done.Sub(Start))/1000000000 //Seconds
2157 TotalMB := float64(total)/1000000 //MB
2158 MBps := TotalMB / Elps
2159 fmt.Printf(Elapsed(Start)+"--In- X: %s (%v/%v/%v) %v %.3fMB/s\n",
2160     what,total,size,nio,absize(total),MBps)
2161 return total
2162 }
2163 func tcpPush(clnt *os.File){
2164     // shrink socket buffer and recover
2165     usleep(100);
2166 }
2167 func (gsh*GshContext)RexecServer(argv[]string){
2168     debug := true
2169     Start0 := time.Now()
2170     Start := Start0
2171     // if local == "" { local = "0.0.0.0:9999" }
2172     local := "0.0.0.0:9999"
2173
2174     if 0 < len(argv) {
2175         if argv[0] == "-s" {
2176             debug = false
2177             argv = argv[1:]
2178         }
2179     }
2180     if 0 < len(argv) {
2181         argv = argv[1:]
2182     }
2183     port, err := net.ResolveTCPAddr("tcp",local);
2184     if err != nil {
2185         fmt.Printf("--En- S: Address error: %s (%s)\n",local,err)
2186         return
2187     }
2188     fmt.Printf(Elapsed(Start)+"--In- S: Listening at %s...\n",local);
2189     sconn, err := net.ListenTCP("tcp", port)
2190     if err != nil {
2191         fmt.Printf(Elapsed(Start)+"--En- S: Listen error: %s (%s)\n",local,err)
2192         return
2193     }
2194
2195     reqbuf := make([]byte,LINESIZE)
2196     res := ""
2197     for {
2198         fmt.Printf(Elapsed(Start0)+"--In- S: Listening at %s...\n",local);
2199         aconn, err := sconn.AcceptTCP()
2200         Start = time.Now()
2201         if err != nil {
2202             fmt.Printf(Elapsed(Start)+"--En- S: Accept error: %s (%s)\n",local,err)
2203             return
2204         }
2205         clnt, _ := aconn.File()
2206         fd := clnt.Fd()
2207         ar := aconn.RemoteAddr()
2208         if debug { fmt.Printf(Elapsed(Start0)+"--In- S: Accepted TCP at %s [%d] <- %v\n",
2209             local,fd,ar) }
2210         res = fmt.Sprintf("220 GShell/%s Server\r\n",VERSION)
2211         fmt.Fprintf(clnt,"%s",res)
2212         if debug { fmt.Printf(Elapsed(Start)+"--In- S: %s",res) }
2213         count, err := clnt.Read(reqbuf)
2214         if err != nil {
2215             fmt.Printf(Elapsed(Start)+"--En- C: (%v %v) %v",
2216                 count,err,string(reqbuf))
2217         }
2218         req := string(reqbuf[:count])
2219         if debug { fmt.Printf(Elapsed(Start)+"--In- C: %v",string(req)) }
2220         reqv := strings.Split(string(req),"r")
2221         cmdv := gshScanArg(reqv[0],0)
2222         //cmdv := strings.Split(reqv[0]," ")
2223         switch cmdv[0] {
2224             case "HELO":
2225                 res = fmt.Sprintf("250 %v",req)
2226             case "GET":
2227                 // download {remotefile|-zN} [localfile]
2228                 var dsize int64 = 32*1024*1024
2229                 var bsize int = 64*1024
2230                 var fname string = ""
2231                 var in *os.File = nil
2232                 var pseudoEOF = false

```

```

2233     if l < len(cmdv) {
2234         fname = cmdv[l]
2235         if strBegins(fname, "-z") {
2236             fmt.Sscanf(fname[2:], "%d", &dsize)
2237         } else {
2238             if strBegins(fname, "{") {
2239                 xin, xout, err := gsh.Popen(fname, "r")
2240                 if err {
2241                     } else {
2242                         xout.Close()
2243                         defer xin.Close()
2244                         in = xin
2245                         dsize = MaxStreamSize
2246                         pseudoEOF = true
2247                     }
2248                 } else {
2249                     xin, err := os.Open(fname)
2250                     if err != nil {
2251                         fmt.Printf("--En- GET (%v)\n", err)
2252                     } else {
2253                         defer xin.Close()
2254                         in = xin
2255                         fi, _ := xin.Stat()
2256                         dsize = fi.Size()
2257                     }
2258                 }
2259             }
2260             //fmt.Printf(Elapsed(Start)+"--In- GET %v:%v\n", dsize, bsize)
2261             res = fmt.Sprintf("200 %v\r\n", dsize)
2262             fmt.Fprintf(clnt, "%v", res)
2263             tcpPush(clnt); // should be separated as line in receiver
2264             fmt.Printf(Elapsed(Start)+"--In- S: %v", res)
2265             wcount := fileRelay("SendGET", in, clnt, dsize, bsize)
2266             if pseudoEOF {
2267                 in.Close() // pipe from the command
2268                 // show end of stream data (its size) by OOB?
2269                 SoftEOF := fmt.Sprintf("(SoftEOF %v)", wcount)
2270                 fmt.Printf(Elapsed(Start)+"--In- S: Send %v\n", SoftEOF)
2271             }
2272             tcpPush(clnt); // to let SoftEOF data appear at the top of received data
2273             fmt.Fprintf(clnt, "%v\r\n", SoftEOF)
2274             tcpPush(clnt); // to let SoftEOF alone in a packet (separate with 200 OK)
2275             // with client generated random?
2276             //fmt.Printf("--In- L: close %v (%v)\n", in.Fd(), in.Name())
2277         }
2278         res = fmt.Sprintf("200 GET done\r\n")
2279     case "PUT":
2280         // upload {srcfile[-zN]} [dstfile]
2281         var dsize int64 = 32*1024*1024
2282         var bsize int = 64*1024
2283         var fname string = ""
2284         var out *os.File = nil
2285         if l < len(cmdv) { // localfile
2286             fmt.Sscanf(cmdv[l], "%d", &dsize)
2287         }
2288         if 2 < len(cmdv) {
2289             fname = cmdv[2]
2290             if fname == "-" {
2291                 // nul dev
2292             } else {
2293                 if strBegins(fname, "{") {
2294                     xin, xout, err := gsh.Popen(fname, "w")
2295                     if err {
2296                         } else {
2297                             xin.Close()
2298                             defer xout.Close()
2299                             out = xout
2300                         }
2301                     } else {
2302                         // should write to temporary file
2303                         // should suppress ^C on tty
2304                         xout, err := os.OpenFile(fname, os.O_CREATE|os.O_RDWR|os.O_TRUNC, 0600)
2305                         //fmt.Printf("--In- S: open(%v) out(%v) err(%v)\n", fname, xout, err)
2306                         if err != nil {
2307                             fmt.Printf("--En- PUT (%v)\n", err)
2308                         } else {
2309                             out = xout
2310                         }
2311                     }
2312                 }
2313                 fmt.Printf(Elapsed(Start)+"--In- L: open(%v,w) %v (%v)\n",
2314                     fname, local, err)
2315             }
2316             fmt.Printf(Elapsed(Start)+"--In- PUT %v (%v)\n", dsize, bsize)
2317             fmt.Printf(Elapsed(Start)+"--In- S: 200 %v OK\r\n", dsize)
2318             fmt.Fprintf(clnt, "200 %v OK\r\n", dsize)
2319             fileRelay("RecvPUT", clnt, out, dsize, bsize)
2320             res = fmt.Sprintf("200 PUT done\r\n")
2321         default:
2322             res = fmt.Sprintf("400 What? %v", req)
2323         }
2324         swcc, serr := clnt.Write([]byte(res))
2325         if serr != nil {
2326             fmt.Printf(Elapsed(Start)+"--In- S: (wc=%v er=%v) %v", swcc, serr, res)
2327         } else {
2328             fmt.Printf(Elapsed(Start)+"--In- S: %v", res)
2329         }
2330         aconn.Close();
2331         clnt.Close();
2332     }
2333     sconn.Close();
2334 }
2335 func (gsh *GshContext) RexecClient(argv []string) (int, string) {
2336     debug := true
2337     Start := time.Now()
2338     if len(argv) == 1 {
2339         return -1, "EmptyARG"
2340     }
2341     argv = argv[1:]
2342     if argv[0] == "-serv" {
2343         gsh.RxecServer(argv[1:])
2344         return 0, "Server"
2345     }
2346     remote := "0.0.0.0:9999"
2347     if argv[0][0] == '@' {
2348         remote = argv[0][1:]
2349         argv = argv[1:]
2350     }
2351     if argv[0] == "-s" {
2352         debug = false
2353         argv = argv[1:]
2354     }
2355     dport, err := net.ResolveTCPAddr("tcp", remote);
2356     if err != nil {
2357         fmt.Printf(Elapsed(Start)+"Address error: %s (%s)\n", remote, err)

```

```

2357     return -1,"AddressError"
2358 }
2359 fmt.Printf(Elapsed(Start)+"--In- C: Connecting to %s\n",remote)
2360 serv, err := net.DialTCP("tcp",nil,dport)
2361 if err != nil {
2362     fmt.Printf(Elapsed(Start)+"Connection error: %s (%s)\n",remote,err)
2363     return -1,"CannotConnect"
2364 }
2365 if debug {
2366     al := serv.LocalAddr()
2367     fmt.Printf(Elapsed(Start)+"--In- C: Connected to %v <- %v\n",remote,al)
2368 }
2369
2370 req := ""
2371 res := make([]byte,LINESIZE)
2372 count,err := serv.Read(res)
2373 if err != nil {
2374     fmt.Printf("--En- S: (%3d,%v) %v",count,err,string(res))
2375 }
2376 if debug { fmt.Printf(Elapsed(Start)+"--In- S: %v",string(res)) }
2377
2378 if argv[0] == "GET" {
2379     savPA := gsh.gshPA
2380     var bsize int = 64*1024
2381     req = fmt.Sprintf("%v\r\n",strings.Join(argv, " "))
2382     fmt.Printf(Elapsed(Start)+"--In- C: %v",req)
2383     fmt.Fprintf(serv,req)
2384     count,err = serv.Read(res)
2385     if err != nil {
2386     }else{
2387         var dsize int64 = 0
2388         var out *os.File = nil
2389         var out_tobeclosed *os.File = nil
2390         var fname string = ""
2391         var rcode int = 0
2392         var pid int = -1
2393         fmt.Sscanf(string(res),"%d %d",&rcode,&dsize)
2394         fmt.Printf(Elapsed(Start)+"--In- S: %v",string(res[0:count]))
2395         if 3 <= len(argv) {
2396             fname = argv[2]
2397             if strBegins(fname,"{") {
2398                 xin,xout,err := gsh.Popen(fname,"w")
2399                 if err {
2400                     }else{
2401                         xin.Close()
2402                         defer xout.Close()
2403                         out = xout
2404                         out_tobeclosed = xout
2405                         pid = 0 // should be its pid
2406                     }
2407                 }else{
2408                     // should write to temporary file
2409                     // should suppress ^C on tty
2410                     xout,err := os.OpenFile(fname,os.O_CREATE|os.O_RDWR|os.O_TRUNC,0600)
2411                     if err != nil {
2412                         fmt.Print("--En- %v\n",err)
2413                     }
2414                     out = xout
2415                     //fmt.Printf("--In-- %d > %s\n",out.Fd(),fname)
2416                 }
2417             }
2418             in, _ := serv.File()
2419             fileRelay("RecvGET",in,out,dsize,bsize)
2420             if 0 <= pid {
2421                 gsh.gshPA = savPA // recovery of Fd(), and more?
2422                 fmt.Printf(Elapsed(Start)+"--In- L: close Pipe > %v\n",fname)
2423                 out_tobeclosed.Close()
2424                 //syscall.Wait4(pid,nil,0,nil) //##
2425             }
2426         }
2427     }else
2428 if argv[0] == "PUT" {
2429     remote, _ := serv.File()
2430     var local *os.File = nil
2431     var dsize int64 = 32*1024*1024
2432     var bsize int = 64*1024
2433     var ofile string = "-"
2434     //fmt.Printf("--I-- Rex %v\n",argv)
2435     if 1 < len(argv) {
2436         fname := argv[1]
2437         if strBegins(fname,"-z") {
2438             fmt.Sscanf(fname[2:],"%d",&dsize)
2439         }else
2440         if strBegins(fname,"{") {
2441             xin,xout,err := gsh.Popen(fname,"r")
2442             if err {
2443                 }else{
2444                     xout.Close()
2445                     defer xin.Close()
2446                     //in = xin
2447                     local = xin
2448                     fmt.Printf("--In- [%d] < Upload output of %v\n",
2449                         local.Fd(),fname)
2450                     ofile = "-from."+fname
2451                     dsize = MaxStreamSize
2452                 }
2453             }else{
2454                 xlocal,err := os.Open(fname)
2455                 if err != nil {
2456                     fmt.Printf("--En- (%s)\n",err)
2457                     local = nil
2458                 }else{
2459                     local = xlocal
2460                     fi,_ := local.Stat()
2461                     dsize = fi.Size()
2462                     defer local.Close()
2463                     //fmt.Printf("--I-- Rex in(%v / %v)\n",ofile,dsize)
2464                 }
2465                 ofile = fname
2466                 fmt.Printf(Elapsed(Start)+"--In- L: open(%v,r)=%v %v (%v)\n",
2467                     fname,dsize,local,err)
2468             }
2469         }
2470     if 2 < len(argv) && argv[2] != "" {
2471         ofile = argv[2]
2472         //fmt.Printf("(%d)%v B.ofile=%v\n",len(argv),argv,ofile)
2473     }
2474     //fmt.Printf(Elapsed(Start)+"--I-- Rex out(%v)\n",ofile)
2475     fmt.Printf(Elapsed(Start)+"--In- PUT %v (/ %v)\n",dsize,bsize)
2476     req = fmt.Sprintf("PUT %v %v \r\n",dsize,ofile)
2477     if debug { fmt.Printf(Elapsed(Start)+"--In- C: %v",req) }
2478     fmt.Fprintf(serv,"%v",req)
2479     count,err = serv.Read(res)
2480     if debug { fmt.Printf(Elapsed(Start)+"--In- S: %v",string(res[0:count])) }

```

```

2481     fileRelay("SendPUT",local,remote,dsize,bsize)
2482 }else{
2483     req = fmt.Sprintf("%v\r\n",strings.Join(argv," "))
2484     if debug { fmt.Printf(Elapsed(Start)+"--In- C: %v",req) }
2485     fmt.Fprintf(serv,"%v",req)
2486     //fmt.Printf("--In- sending RexRequest(%v)\n",len(req))
2487 }
2488 //fmt.Printf(Elapsed(Start)+"--In- waiting RexResponse...\n")
2489 count,err = serv.Read(res)
2490 ress := ""
2491 if count == 0 {
2492     ress = "(nil)\r\n"
2493 }else{
2494     ress = string(res[:count])
2495 }
2496 if err != nil {
2497     fmt.Printf(Elapsed(Start)+"--En- S: (%d,%v) %v",count,err,ress)
2498 }else{
2499     fmt.Printf(Elapsed(Start)+"--In- S: %v",ress)
2500 }
2501 serv.Close()
2502 //conn.Close()
2503
2504 var stat string
2505 var rcode int
2506 fmt.Sscanf(ress,"%d %s",&rcode,&stat)
2507 //fmt.Printf("--D-- Client: %v (%v)",rcode,stat)
2508 return rcode,ress
2509 }
2510
2511 // <a name="remote-sh">Remote Shell</a>
2512 // gcp file [...] { [host]:[port]:[dir] | dir } // -p | -no-p
2513 func (gsh*GshContext)FileCopy(argv[]string){
2514     var host = ""
2515     var port = ""
2516     var upload = false
2517     var download = false
2518     var xargv = []string{"rex-gcp"}
2519     var srcv = []string{}
2520     var dstv = []string{}
2521     argv = argv[1:]
2522
2523     for _,v := range argv {
2524         /*
2525         if v[0] == '-' { // might be a pseudo file (generated date)
2526             continue
2527         }
2528         */
2529         obj := strings.Split(v,":")
2530         //fmt.Printf("%d %v %v\n",len(obj),v,obj)
2531         if 1 < len(obj) {
2532             host = obj[0]
2533             file := ""
2534             if 0 < len(host) {
2535                 gsh.LastServer.host = host
2536             }else{
2537                 host = gsh.LastServer.host
2538                 port = gsh.LastServer.port
2539             }
2540             if 2 < len(obj) {
2541                 port = obj[1]
2542                 if 0 < len(port) {
2543                     gsh.LastServer.port = port
2544                 }else{
2545                     port = gsh.LastServer.port
2546                 }
2547                 file = obj[2]
2548             }else{
2549                 file = obj[1]
2550             }
2551             if len(srcv) == 0 {
2552                 download = true
2553                 srcv = append(srcv,file)
2554                 continue
2555             }
2556             upload = true
2557             dstv = append(dstv,file)
2558             continue
2559         }
2560         /*
2561         idx := strings.Index(v,":")
2562         if 0 <= idx {
2563             remote = v[0:idx]
2564             if len(srcv) == 0 {
2565                 download = true
2566                 srcv = append(srcv,v[idx+1:])
2567                 continue
2568             }
2569             upload = true
2570             dstv = append(dstv,v[idx+1:])
2571             continue
2572         }
2573         */
2574         if download {
2575             dstv = append(dstv,v)
2576         }else{
2577             srcv = append(srcv,v)
2578         }
2579     }
2580     hostport := "@" + host + ":" + port
2581     if upload {
2582         if host != "" { xargv = append(xargv,hostport) }
2583         xargv = append(xargv,"PUT")
2584         xargv = append(xargv,srcv[0:]...)
2585         xargv = append(xargv,dstv[0:]...)
2586         //fmt.Printf("--I-- FileCopy PUT gsh://%s/%v < %v // %v\n",hostport,dstv,srcv)
2587         fmt.Printf("--I-- FileCopy PUT gsh://%s/%v < %v\n",hostport,dstv,srcv)
2588         gsh.RexecClient(xargv)
2589     }else
2590     if download {
2591         if host != "" { xargv = append(xargv,hostport) }
2592         xargv = append(xargv,"GET")
2593         xargv = append(xargv,srcv[0:]...)
2594         xargv = append(xargv,dstv[0:]...)
2595         //fmt.Printf("--I-- FileCopy GET gsh://%v/%v > %v // %v\n",hostport,srcv,dstv,xargv)
2596         fmt.Printf("--I-- FileCopy GET gsh://%v/%v > %v\n",hostport,srcv,dstv)
2597         gsh.RexecClient(xargv)
2598     }else{
2599     }
2600 }
2601
2602 // target
2603 func (gsh*GshContext)Trelpath(rloc string)(string){
2604     cwd, _ := os.Getwd()

```

```

2605     os.Chdir(gsh.RWD)
2606     os.Chdir(rloc)
2607     twd, _ := os.Getwd()
2608     os.Chdir(cwd)
2609
2610     tpath := twd + "/" + rloc
2611     return tpath
2612 }
2613 // join to remote GShell - [user@]host[:port] or cd host[:port]:path
2614 func (gsh*GshContext)Rjoin(argv[]string){
2615     if len(argv) <= 1 {
2616         fmt.Printf("--I-- current server = %v\n",gsh.RSERV)
2617         return
2618     }
2619     serv := argv[1]
2620     servv := strings.Split(serv,":")
2621     if 1 <= len(servv) {
2622         if servv[0] == "lo" {
2623             servv[0] = "localhost"
2624         }
2625     }
2626     switch len(servv) {
2627     case 1:
2628         //if strings.Index(serv,":") < 0 {
2629             serv = servv[0] + ":" + fmt.Sprintf("%d",GSH_PORT)
2630         //}
2631     case 2: // host:port
2632         serv = strings.Join(servv,":")
2633     }
2634     xargv := []string{"rex-join", "@"+serv, "HELO"}
2635     rcode,stat := gsh.RexecClient(xargv)
2636     if (rcode / 100) == 2 {
2637         fmt.Printf("--I-- OK Joined (%v) %v\n",rcode,stat)
2638         gsh.RSERV = serv
2639     }else{
2640         fmt.Printf("--I-- NG, could not joined (%v) %v\n",rcode,stat)
2641     }
2642 }
2643 func (gsh*GshContext)Rexec(argv[]string){
2644     if len(argv) <= 1 {
2645         fmt.Printf("--I-- rexec command [ | {file || {command} ]\n",gsh.RSERV)
2646         return
2647     }
2648     /*
2649     nargv := gshScanArg(strings.Join(argv," "),0)
2650     fmt.Printf("--D-- nargc=%d [%v]\n",len(nargv),nargv)
2651     if nargv[1][0] != '{' {
2652         nargv[1] = "{" + nargv[1] + "}"
2653     }
2654     fmt.Printf("--D-- nargc=%d [%v]\n",len(nargv),nargv)
2655     */
2656     argv = nargv
2657     /*
2658     nargv := []string{}
2659     nargv = append(nargv, {"+"strings.Join(argv[1:], " ")+"})"
2660     fmt.Printf("--D-- nargc=%d %v\n",len(nargv),nargv)
2661     argv = nargv
2662     */
2663     xargv := []string{"rex-exec", "@"+gsh.RSERV, "GET"}
2664     xargv = append(xargv,argv...)
2665     xargv = append(xargv,"/dev/tty")
2666     rcode,stat := gsh.RexecClient(xargv)
2667     if (rcode / 100) == 2 {
2668         fmt.Printf("--I-- OK Rexec (%v) %v\n",rcode,stat)
2669     }else{
2670         fmt.Printf("--I-- NG Rexec (%v) %v\n",rcode,stat)
2671     }
2672 }
2673 func (gsh*GshContext)Rchdir(argv[]string){
2674     if len(argv) <= 1 {
2675         return
2676     }
2677     cwd, _ := os.Getwd()
2678     os.Chdir(gsh.RWD)
2679     os.Chdir(argv[1])
2680     twd, _ := os.Getwd()
2681     gsh.RWD = twd
2682     fmt.Printf("--I-- JWD=%v\n",twd)
2683     os.Chdir(cwd)
2684 }
2685 func (gsh*GshContext)Rpwd(argv[]string){
2686     fmt.Printf("%v\n",gsh.RWD)
2687 }
2688 func (gsh*GshContext)Rls(argv[]string){
2689     cwd, _ := os.Getwd()
2690     os.Chdir(gsh.RWD)
2691     argv[0] = "-ls"
2692     gsh.xFind(argv)
2693     os.Chdir(cwd)
2694 }
2695 func (gsh*GshContext)Rput(argv[]string){
2696     var local string = ""
2697     var remote string = ""
2698     if 1 < len(argv) {
2699         local = argv[1]
2700         remote = local // base name
2701     }
2702     if 2 < len(argv) {
2703         remote = argv[2]
2704     }
2705     fmt.Printf("--I-- jput from=%v to=%v\n",local,gsh.Trelpath(remote))
2706 }
2707 func (gsh*GshContext)Rget(argv[]string){
2708     var remote string = ""
2709     var local string = ""
2710     if 1 < len(argv) {
2711         remote = argv[1]
2712         local = remote // base name
2713     }
2714     if 2 < len(argv) {
2715         local = argv[2]
2716     }
2717     fmt.Printf("--I-- jget from=%v to=%v\n",gsh.Trelpath(remote),local)
2718 }
2719
2720 // <a name="network">network</a>
2721 // -s, -si, -so // bi-directional, source, sync (maybe socket)
2722 func (gshCtx*GshContext)sconnect(inTCP bool, argv []string) {
2723     gshPA := gshCtx.gshPA
2724     if len(argv) < 2 {
2725         fmt.Printf("Usage: -s [host]:[port[.udp]]\n")
2726         return
2727     }
2728     remote := argv[1]

```

```

2729 if remote == ":" { remote = "0.0.0.0:9999" }
2730
2731 if inTCP { // TCP
2732     dport, err := net.ResolveTCPAddr("tcp",remote);
2733     if err != nil {
2734         fmt.Printf("Address error: %s (%s)\n",remote,err)
2735         return
2736     }
2737     conn, err := net.DialTCP("tcp",nil,dport)
2738     if err != nil {
2739         fmt.Printf("Connection error: %s (%s)\n",remote,err)
2740         return
2741     }
2742     file, _ := conn.File();
2743     fd := file.Fd()
2744     fmt.Printf("Socket: connected to %s, socket[%d]\n",remote,fd)
2745
2746     savfd := gshPA.Files[1]
2747     gshPA.Files[1] = fd;
2748     gshCtx.gshellv(argv[2:])
2749     gshPA.Files[1] = savfd
2750     file.Close()
2751     conn.Close()
2752 }else{
2753     //dport, err := net.ResolveUDPAddr("udp4",remote);
2754     dport, err := net.ResolveUDPAddr("udp",remote);
2755     if err != nil {
2756         fmt.Printf("Address error: %s (%s)\n",remote,err)
2757         return
2758     }
2759     //conn, err := net.DialUDP("udp4",nil,dport)
2760     conn, err := net.DialUDP("udp",nil,dport)
2761     if err != nil {
2762         fmt.Printf("Connection error: %s (%s)\n",remote,err)
2763         return
2764     }
2765     file, _ := conn.File();
2766     fd := file.Fd()
2767
2768     ar := conn.RemoteAddr()
2769     //al := conn.LocalAddr()
2770     fmt.Printf("Socket: connected to %s [%s], socket[%d]\n",
2771         remote,ar.String(),fd)
2772
2773     savfd := gshPA.Files[1]
2774     gshPA.Files[1] = fd;
2775     gshCtx.gshellv(argv[2:])
2776     gshPA.Files[1] = savfd
2777     file.Close()
2778     conn.Close()
2779 }
2780 }
2781 func (gshCtx*GshContext)saccept(inTCP bool, argv []string) {
2782     gshPA := gshCtx.gshPA
2783     if len(argv) < 2 {
2784         fmt.Printf("Usage: -ac [host]:[port[.udp]]\n")
2785         return
2786     }
2787     local := argv[1]
2788     if local == ":" { local = "0.0.0.0:9999" }
2789     if inTCP { // TCP
2790         port, err := net.ResolveTCPAddr("tcp",local);
2791         if err != nil {
2792             fmt.Printf("Address error: %s (%s)\n",local,err)
2793             return
2794         }
2795         //fmt.Printf("Listen at %s...\n",local);
2796         sconn, err := net.ListenTCP("tcp", port)
2797         if err != nil {
2798             fmt.Printf("Listen error: %s (%s)\n",local,err)
2799             return
2800         }
2801         //fmt.Printf("Accepting at %s...\n",local);
2802         aconn, err := sconn.AcceptTCP()
2803         if err != nil {
2804             fmt.Printf("Accept error: %s (%s)\n",local,err)
2805             return
2806         }
2807         file, _ := aconn.File()
2808         fd := file.Fd()
2809         fmt.Printf("Accepted TCP at %s [%d]\n",local,fd)
2810
2811         savfd := gshPA.Files[0]
2812         gshPA.Files[0] = fd;
2813         gshCtx.gshellv(argv[2:])
2814         gshPA.Files[0] = savfd
2815
2816         sconn.Close();
2817         aconn.Close();
2818         file.Close();
2819     }else{
2820         //port, err := net.ResolveUDPAddr("udp4",local);
2821         port, err := net.ResolveUDPAddr("udp",local);
2822         if err != nil {
2823             fmt.Printf("Address error: %s (%s)\n",local,err)
2824             return
2825         }
2826         fmt.Printf("Listen UDP at %s...\n",local);
2827         //uconn, err := net.ListenUDP("udp4", port)
2828         uconn, err := net.ListenUDP("udp", port)
2829         if err != nil {
2830             fmt.Printf("Listen error: %s (%s)\n",local,err)
2831             return
2832         }
2833         file, _ := uconn.File()
2834         fd := file.Fd()
2835         ar := uconn.RemoteAddr()
2836         remote := ""
2837         if ar != nil { remote = ar.String() }
2838         if remote == "" { remote = "?" }
2839
2840         // not yet received
2841         //fmt.Printf("Accepted at %s [%d] <- %s\n",local,fd,"")
2842
2843         savfd := gshPA.Files[0]
2844         gshPA.Files[0] = fd;
2845         savenv := gshPA.Env
2846         gshPA.Env = append(savenv, "REMOTE_HOST="+remote)
2847         gshCtx.gshellv(argv[2:])
2848         gshPA.Env = savenv
2849         gshPA.Files[0] = savfd
2850
2851         uconn.Close();
2852         file.Close();

```

```

2853 }
2854 }
2855
2856 // empty line command
2857 func (gshCtx *GshContext)xPwd(argv []string){
2858 // execute context command, pwd + date
2859 // context notation, representation scheme, to be resumed at re-login
2860 cwd, _ := os.Getwd()
2861 switch {
2862 case isin("-a",argv):
2863 gshCtx.ShowChdirHistory(argv)
2864 case isin("-ls",argv):
2865 showFileInfo(cwd,argv)
2866 default:
2867 fmt.Printf("%s\n",cwd)
2868 case isin("-v",argv): // obsolete empty command
2869 t := time.Now()
2870 date := t.Format(time.UnixDate)
2871 exe, _ := os.Executable()
2872 host, _ := os.Hostname()
2873 fmt.Printf("{PWD=\"%s\" ",cwd)
2874 fmt.Printf("HOST=\"%s\" ",host)
2875 fmt.Printf("DATE=\"%s\" ",date)
2876 fmt.Printf("TIME=\"%s\" ",t.String())
2877 fmt.Printf("PID=\"%d\" ",os.Getpid())
2878 fmt.Printf("EXE=\"%s\" ",exe)
2879 fmt.Printf("}\n")
2880 }
2881 }
2882
2883 // <a name="history">History</a>
2884 // these should be browsed and edited by HTTP browser
2885 // show the time of command with -t and direcotry with -ls
2886 // openfile-history, sort by -a -m -c
2887 // sort by elapsed time by -t -s
2888 // search by "more" like interface
2889 // edit history
2890 // sort history, and wc or uniq
2891 // CPU and other resource consumptions
2892 // limit showing range (by time or so)
2893 // export / import history
2894 func (gshCtx *GshContext)xHistory(argv []string){
2895 atWorkDirX := -1
2896 if 1 < len(argv) && strBegins(argv[1],"@") {
2897 atWorkDirX, _ = strconv.Atoi(argv[1][1:])
2898 }
2899 //fmt.Printf("--D-- showHistory(%v)\n",argv)
2900 for i, v := range gshCtx.CommandHistory {
2901 // exclude commands not to be listed by default
2902 // internal commands may be suppressed by default
2903 if v.CmdLine == "" && !isin("-a",argv) {
2904 continue;
2905 }
2906 if 0 <= atWorkDirX {
2907 if v.WorkDirX != atWorkDirX {
2908 continue
2909 }
2910 }
2911 if !isin("-n",argv){ // like "fc"
2912 fmt.Printf("!%-2d ",i)
2913 }
2914 if isin("-v",argv){
2915 fmt.Println(v) // should be with it date
2916 }else{
2917 if isin("-l",argv) || isin("-l0",argv) {
2918 elps := v.EndAt.Sub(v.StartAt);
2919 start := v.StartAt.Format(time.Stamp)
2920 fmt.Printf("@%d ",v.WorkDirX)
2921 fmt.Printf("[%v] %11v/t ",start,elps)
2922 }
2923 if isin("-l",argv) && !isin("-l0",argv){
2924 fmt.Printf("%v",Rusagef("%t %u\t// %s",argv,v.Rusagev))
2925 }
2926 if isin("-at",argv) { // isin("-ls",argv){
2927 dhi := v.WorkDirX // workdir history index
2928 fmt.Printf("@%d %s\t",dhi,v.WorkDir)
2929 // show the FileInfo of the output command??
2930 }
2931 fmt.Printf("%s",v.CmdLine)
2932 fmt.Printf("\n")
2933 }
2934 }
2935 }
2936 // !n - history index
2937 func searchHistory(gshCtx GshContext, gline string) (string, bool, bool){
2938 if gline[0] == '!' {
2939 hix, err := strconv.Atoi(gline[1:])
2940 if err != nil {
2941 fmt.Printf("--E-- (%s : range)\n",hix)
2942 return "", false, true
2943 }
2944 if hix < 0 || len(gshCtx.CommandHistory) <= hix {
2945 fmt.Printf("--E-- (%d : out of range)\n",hix)
2946 return "", false, true
2947 }
2948 return gshCtx.CommandHistory[hix].CmdLine, false, false
2949 }
2950 // search
2951 //for i, v := range gshCtx.CommandHistory {
2952 //}
2953 return gline, false, false
2954 }
2955 func (gsh *GshContext)cmdStringInHistory(hix int)(cmd string, ok bool){
2956 if 0 <= hix && hix < len(gsh.CommandHistory) {
2957 return gsh.CommandHistory[hix].CmdLine,true
2958 }
2959 return "",false
2960 }
2961
2962 // temporary adding to PATH environment
2963 // cd name -lib for LD_LIBRARY_PATH
2964 // chdir with directory history (date + full-path)
2965 // -s for sort option (by visit date or so)
2966 func (gsh *GshContext)ShowChdirHistory1(i int, v GChdirHistory, argv []string){
2967 fmt.Printf("!%-2d ",v.CmdIndex) // the first command at this WorkDir
2968 fmt.Printf("@%d ",i)
2969 fmt.Printf("[%v] ",v.MovedAt.Format(time.Stamp))
2970 showFileInfo(v.Dir,argv)
2971 }
2972 func (gsh *GshContext)ShowChdirHistory(argv []string){
2973 for i, v := range gsh.ChdirHistory {
2974 gsh.ShowChdirHistory1(i,v,argv)
2975 }
2976 }

```



```

2977 func skipOpts(argv []string)(int){
2978     for i,v := range argv {
2979         if strBegins(v,"-") {
2980             }else{
2981                 return i
2982             }
2983     }
2984     return -1
2985 }
2986 func (gshCtx*GshContext)xChdir(argv []string){
2987     cdhist := gshCtx.CdhistHistory
2988     if isin("?",argv) || isin("-t",argv) || isin("-a",argv) {
2989         gshCtx.ShowChdirHistory(argv)
2990         return
2991     }
2992     pwd, _ := os.Getwd()
2993     dir := ""
2994     if len(argv) <= 1 {
2995         dir = toFullpath("-")
2996     }else{
2997         i := skipOpts(argv[1:])
2998         if i < 0 {
2999             dir = toFullpath("-")
3000         }else{
3001             dir = argv[1+i]
3002         }
3003     }
3004     if strBegins(dir,"@") {
3005         if dir == "@0" { // obsolete
3006             dir = gshCtx.StartDir
3007         }else
3008         if dir == "@!" {
3009             index := len(cdhist) - 1
3010             if 0 < index { index -= 1 }
3011             dir = cdhist[index].Dir
3012         }else{
3013             index, err := strconv.Atoi(dir[1:])
3014             if err != nil {
3015                 fmt.Printf("--E-- xChdir(%v)\n",err)
3016                 dir = "?"
3017             }else
3018             if len(gshCtx.CdhistHistory) <= index {
3019                 fmt.Printf("--E-- xChdir(history range error)\n")
3020                 dir = "?"
3021             }else{
3022                 dir = cdhist[index].Dir
3023             }
3024         }
3025     }
3026     if dir != "?" {
3027         err := os.Chdir(dir)
3028         if err != nil {
3029             fmt.Printf("--E-- xChdir(%s)(%v)\n",argv[1],err)
3030         }else{
3031             cwd, _ := os.Getwd()
3032             if cwd != pwd {
3033                 hist1 := GChdirHistory { }
3034                 hist1.Dir = cwd
3035                 hist1.MovedAt = time.Now()
3036                 hist1.CmdIndex = len(gshCtx.CommandHistory)+1
3037                 gshCtx.CdhistHistory = append(cdhist,hist1)
3038                 if !isin("-s",argv){
3039                     //cwd, _ := os.Getwd()
3040                     //fmt.Printf("%s\n",cwd)
3041                     ix := len(gshCtx.CdhistHistory)-1
3042                     gshCtx.ShowChdirHistory1(ix,hist1,argv)
3043                 }
3044             }
3045         }
3046     }
3047     if isin("-ls",argv){
3048         cwd, _ := os.Getwd()
3049         showFileInfo(cwd,argv);
3050     }
3051 }
3052 func TimeValSub(tv1 *syscall.Timeval, tv2 *syscall.Timeval){
3053     *tv1 = syscall.NsecToTimeval(tv1.Nano() - tv2.Nano())
3054 }
3055 func RusageSubv(ru1, ru2 [2]syscall.Rusage){[2]syscall.Rusage}{
3056     TimeValSub(&ru1[0].Utime,&ru2[0].Utime)
3057     TimeValSub(&ru1[0].Stime,&ru2[0].Stime)
3058     TimeValSub(&ru1[1].Utime,&ru2[1].Utime)
3059     TimeValSub(&ru1[1].Stime,&ru2[1].Stime)
3060     return ru1
3061 }
3062 func TimeValAdd(tv1 syscall.Timeval, tv2 syscall.Timeval)(syscall.Timeval){
3063     tvs := syscall.NsecToTimeval(tv1.Nano() + tv2.Nano())
3064     return tvs
3065 }
3066 /*
3067 func RusageAddv(ru1, ru2 [2]syscall.Rusage){[2]syscall.Rusage}{
3068     TimeValAdd(ru1[0].Utime,ru2[0].Utime)
3069     TimeValAdd(ru1[0].Stime,ru2[0].Stime)
3070     TimeValAdd(ru1[1].Utime,ru2[1].Utime)
3071     TimeValAdd(ru1[1].Stime,ru2[1].Stime)
3072     return ru1
3073 }
3074 */
3075 // <a name="rusage">Resource Usage</a>
3076 func sRusagef(fmtspec string, argv []string, ru [2]syscall.Rusage)(string){
3077     // ru[0] self , ru[1] children
3078     ut := TimeValAdd(ru[0].Utime,ru[1].Utime)
3079     st := TimeValAdd(ru[0].Stime,ru[1].Stime)
3080     uu := (ut.Sec*1000000 + int64(ut.Usec)) * 1000
3081     su := (st.Sec*1000000 + int64(st.Usec)) * 1000
3082     tu := uu + su
3083     ret := fmt.Sprintf("%v/sum",abftime(tu))
3084     ret += fmt.Sprintf(", %v/usr",abftime(uu))
3085     ret += fmt.Sprintf(", %v/sys",abftime(su))
3086     return ret
3087 }
3088 }
3089 func Rusagef(fmtspec string, argv []string, ru [2]syscall.Rusage)(string){
3090     ut := TimeValAdd(ru[0].Utime,ru[1].Utime)
3091     st := TimeValAdd(ru[0].Stime,ru[1].Stime)
3092     fmt.Printf("%d.%06ds/u ",ut.Sec,ut.Usec) //ru[1].Utime.Sec,ru[1].Utime.Usec)
3093     fmt.Printf("%d.%06ds/s ",st.Sec,st.Usec) //ru[1].Stime.Sec,ru[1].Stime.Usec)
3094     return ""
3095 }
3096 func Getrusagev(){[2]syscall.Rusage}{
3097     var ruv = [2]syscall.Rusage{}
3098     syscall.Getrusage(syscall.RUSAGE_SELF,&ruv[0])
3099     syscall.Getrusage(syscall.RUSAGE_CHILDREN,&ruv[1])
3100     return ruv

```

```

3101 }
3102 func showRusage(what string,argv []string, ru *syscall.Rusage){
3103     fmt.Printf("%s: ",what);
3104     fmt.Printf("Uus=%d.%06ds",ru.Utime.Sec,ru.Utime.Usec)
3105     fmt.Printf(" Sys=%d.%06ds",ru.Stime.Sec,ru.Stime.Usec)
3106     fmt.Printf(" Rss=%vB",ru.Maxrss)
3107     if isin("-l",argv) {
3108         fmt.Printf(" MinFlt=%v",ru.Minflt)
3109         fmt.Printf(" MajFlt=%v",ru.Majflt)
3110         fmt.Printf(" IxRSS=%vB",ru.Ixrss)
3111         fmt.Printf(" IdRSS=%vB",ru.Idrss)
3112         fmt.Printf(" Nswap=%vB",ru.Nswap)
3113     }
3114     fmt.Printf(" Read=%v",ru.Inblock)
3115     fmt.Printf(" Write=%v",ru.Oblock)
3116     }
3117     fmt.Printf(" Snd=%v",ru.Msgsnd)
3118     fmt.Printf(" Rcv=%v",ru.Msgrcv)
3119     //if isin("-l",argv) {
3120         fmt.Printf(" Sig=%v",ru.Nsignals)
3121     //}
3122     fmt.Printf("\n");
3123 }
3124 func (gshCtx *GshContext)xTime(argv[]string)(bool){
3125     if 2 <= len(argv){
3126         gshCtx.LastRusage = syscall.Rusage{}
3127         rusagev1 := Getrusagev()
3128         fin := gshCtx.gshelv(argv[1:])
3129         rusagev2 := Getrusagev()
3130         showRusage(argv[1],argv,&gshCtx.LastRusage)
3131         rusagev := RusageSubv(rusagev2,rusagev1)
3132         showRusage("self",argv,&rusagev[0])
3133         showRusage("chld",argv,&rusagev[1])
3134         return fin
3135     }else{
3136         rusage:= syscall.Rusage {}
3137         syscall.Getrusage(syscall.RUSAGE_SELF,&rusage)
3138         showRusage("self",argv, &rusage)
3139         syscall.Getrusage(syscall.RUSAGE_CHILDREN,&rusage)
3140         showRusage("chld",argv, &rusage)
3141         return false
3142     }
3143 }
3144 func (gshCtx *GshContext)xJobs(argv[]string){
3145     fmt.Printf("%d Jobs\n",len(gshCtx.BackGroundJobs))
3146     for ji, pid := range gshCtx.BackGroundJobs {
3147         //wstat := syscall.WaitStatus {0}
3148         rusage := syscall.Rusage {}
3149         //wpid, err := syscall.Wait4(pid,&wstat,syscall.WNOHANG,&rusage);
3150         wpid, err := syscall.Wait4(pid,nil,syscall.WNOHANG,&rusage);
3151         if err != nil {
3152             fmt.Printf("--E-- %%d [%d] (%v)\n",ji,pid,err)
3153         }else{
3154             fmt.Printf("%%d[%d](%d)\n",ji,pid,wpid)
3155             showRusage("chld",argv,&rusage)
3156         }
3157     }
3158 }
3159 func (gsh*GshContext)inBackground(argv[]string)(bool){
3160     if gsh.CmdTrace { fmt.Printf("--I-- inBackground(%v)\n",argv) }
3161     gsh.BackGround = true // set background option
3162     xfin := false
3163     xfin = gsh.gshelv(argv)
3164     gsh.BackGround = false
3165     return xfin
3166 }
3167 // -o file without command means just opening it and refer by #N
3168 // should be listed by "files" command
3169 func (gshCtx*GshContext)xOpen(argv[]string){
3170     var pv = []int{-1,-1}
3171     err := syscall.Pipe(pv)
3172     fmt.Printf("--I-- pipe()=(#d,#d)(%v)\n",pv[0],pv[1],err)
3173 }
3174 func (gshCtx*GshContext)fromPipe(argv[]string){
3175 }
3176 func (gshCtx*GshContext)xClose(argv[]string){
3177 }
3178 // <a name="redirect">redirect</a>
3179 func (gshCtx*GshContext)redirect(argv[]string)(bool){
3180     if len(argv) < 2 {
3181         return false
3182     }
3183 }
3184 cmd := argv[0]
3185 fname := argv[1]
3186 var file *os.File = nil
3187
3188 fdix := 0
3189 mode := os.O_RDONLY
3190
3191 switch {
3192 case cmd == "-i" || cmd == "<":
3193     fdix = 0
3194     mode = os.O_RDONLY
3195 case cmd == "-o" || cmd == ">":
3196     fdix = 1
3197     mode = os.O_RDWR | os.O_CREATE
3198 case cmd == "-a" || cmd == ">>":
3199     fdix = 1
3200     mode = os.O_RDWR | os.O_CREATE | os.O_APPEND
3201 }
3202 if fname[0] == '#' {
3203     fd, err := strconv.Atoi(fname[1:])
3204     if err != nil {
3205         fmt.Printf("--E-- (%v)\n",err)
3206         return false
3207     }
3208     file = os.NewFile(uintptr(fd),"MaybePipe")
3209 }else{
3210     xfile, err := os.OpenFile(argv[1], mode, 0600)
3211     if err != nil {
3212         fmt.Printf("--E-- (%s)\n",err)
3213         return false
3214     }
3215     file = xfile
3216 }
3217 gshPA := gshCtx.gshPA
3218 savfd := gshPA.Files[fdix]
3219 gshPA.Files[fdix] = file.Fd()
3220 fmt.Printf("--I-- Opened [%d] %s\n",file.Fd(),argv[1])
3221 gshCtx.gshelv(argv[2:])
3222 gshPA.Files[fdix] = savfd
3223
3224 return false

```

```

3225 }
3226 //fmt.Fprintf(res, "GShell Status: %q", html.EscapeString(req.URL.Path))
3227 func httpHandler(res http.ResponseWriter, req *http.Request){
3228     path := req.URL.Path
3229     fmt.Printf("--I-- Got HTTP Request(%s)\n",path)
3230     {
3231         gshCtxBuf, _ := setupGshContext()
3232         gshCtx := *gshCtxBuf
3233         fmt.Printf("--I-- %s\n",path[1:])
3234         gshCtx.tgshelll(path[1:])
3235     }
3236     fmt.Fprintf(res, "Hello(^-^)/\n%s\n",path)
3237 }
3238 func (gshCtx *GshContext) httpServer(argv []string){
3239     http.HandleFunc("/", httpHandler)
3240     accport := "localhost:9999"
3241     fmt.Printf("--I-- HTTP Server Start at [%s]\n",accport)
3242     http.ListenAndServe(accport,nil)
3243 }
3244 func (gshCtx *GshContext)xGo(argv[]string){
3245     go gshCtx.gshellv(argv[1:]);
3246 }
3247 func (gshCtx *GshContext) xPs(argv[]string){}
3248 }
3249 }
3250 // <a name="plugin">Plugin</a>
3251 // plugin [-ls [names]] to list plugins
3252 // Reference: <a href="https://golang.org/src/plugin/">plugin</a> source code
3253 func (gshCtx *GshContext) whichPlugin(name string,argv[]string)(pi *PluginInfo){
3254     pi = nil
3255     for _,p := range gshCtx.PluginFuncs {
3256         if p.Name == name && pi == nil {
3257             pi = *p
3258         }
3259         if !isin("-s",argv){
3260             //fmt.Printf("%v %v ",i,p)
3261             if isin("-ls",argv){
3262                 showFileInfo(p.Path,argv)
3263             }else{
3264                 fmt.Printf("%s\n",p.Name)
3265             }
3266         }
3267     }
3268     return pi
3269 }
3270 func (gshCtx *GshContext) xPlugin(argv[]string) (error) {
3271     if len(argv) == 0 || argv[0] == "-ls" {
3272         gshCtx.whichPlugin("",argv)
3273         return nil
3274     }
3275     name := argv[0]
3276     Pin := gshCtx.whichPlugin(name,[]string{"-s"})
3277     if Pin != nil {
3278         os.Args = argv // should be recovered?
3279         Pin.Addr.(func())()
3280         return nil
3281     }
3282     sofile := toFullpath(argv[0] + ".so") // or find it by which($PATH)
3283     p, err := plugin.Open(sofile)
3284     if err != nil {
3285         fmt.Printf("--E-- plugin.Open(%s)(%v)\n",sofile,err)
3286         return err
3287     }
3288     fname := "Main"
3289     f, err := p.Lookup(fname)
3290     if( err != nil ){
3291         fmt.Printf("--E-- plugin.Lookup(%s)(%v)\n",fname,err)
3292         return err
3293     }
3294     pin := PluginInfo {p,f,name,sofile}
3295     gshCtx.PluginFuncs = append(gshCtx.PluginFuncs,pin)
3296     fmt.Printf("--I-- added (%d)\n",len(gshCtx.PluginFuncs))
3297 }
3298 //fmt.Printf("--I-- first call(%s:%s)%v\n",sofile,fname,argv)
3299 os.Args = argv
3300 f.(func())()
3301 return err
3302 }
3303 func (gshCtx*GshContext)Args(argv[]string){
3304     for i,v := range os.Args {
3305         fmt.Printf("[%v] %v\n",i,v)
3306     }
3307 }
3308 func (gshCtx *GshContext) showVersion(argv[]string){
3309     if isin("-l",argv) {
3310         fmt.Printf("%v/%v (%v)",NAME,VERSION,DATE);
3311     }else{
3312         fmt.Printf("%v",VERSION);
3313     }
3314     if isin("-a",argv) {
3315         fmt.Printf("%s",AUTHOR)
3316     }
3317     if !isin("-n",argv) {
3318         fmt.Printf("\n")
3319     }
3320 }
3321 }
3322 }
3323 // <a name="scanf">Scanf</a> // string decomposer
3324 // scanf [format] [input]
3325 func scanv(sstr string)(strv[]string){
3326     strv = strings.Split(sstr, " ")
3327     return strv
3328 }
3329 func scanUntil(src,end string)(rstr string,leng int){
3330     idx := strings.Index(src,end)
3331     if 0 <= idx {
3332         rstr = src[0:idx]
3333         return rstr,idx+leng(end)
3334     }
3335     return src,0
3336 }
3337 }
3338 // -bn -- display base-name part only // can be in some %fmt, for sed rewriting
3339 func (gsh*GshContext)printVal(fmts string, vstr string, optv[]string){
3340     //vint,err := strconv.Atoi(vstr)
3341     var ival int64 = 0
3342     n := 0
3343     err := error(nil)
3344     if strBegins(vstr, "_") {
3345         vx, _ := strconv.Atoi(vstr[1:])
3346         if vx < len(gsh.iValues) {
3347             vstr = gsh.iValues[vx]
3348         }
3349     }

```

```

3349     }else{
3350     }
3351     }
3352     // should use Eval()
3353     if strBegins(vstr,"0x") {
3354     n,err = fmt.Sscanf(vstr[2:], "%x", &ival)
3355     }else{
3356     n,err = fmt.Sscanf(vstr, "%d", &ival)
3357     //fmt.Printf("--D-- n=%d err=(%v) {s}=%v\n",n,err,vstr, ival)
3358     }
3359     if n == 1 && err == nil {
3360     //fmt.Printf("--D-- formatn(%v) ival(%v)\n",fmts,ival)
3361     fmt.Printf("%"+fmts,ival)
3362     }else{
3363     if isin("-bn",optv){
3364     fmt.Printf("%"+fmts,filepath.Base(vstr))
3365     }else{
3366     fmt.Printf("%"+fmts,vstr)
3367     }
3368     }
3369     }
3370     func (gsh*GshContext)printfv(fmts,div string,argv []string,optv []string,list []string){
3371     //fmt.Printf("%d",len(list))
3372     //curfmt := "v"
3373     outlen := 0
3374     curfmt := gsh.iFormat
3375
3376     if 0 < len(fmts) {
3377     for xi := 0; xi < len(fmts); xi++ {
3378     fch := fmts[xi]
3379     if fch == '%' {
3380     if xi+1 < len(fmts) {
3381     curfmt = string(fmts[xi+1])
3382     gsh.iFormat = curfmt
3383     xi += 1
3384     if xi+1 < len(fmts) && fmts[xi+1] == '(' {
3385     vals, leng := scanUntil(fmts[xi+2:],")")
3386     //fmt.Printf("--D-- show fmt(%v) val(%v) next(%v)\n",curfmt,vals,leng)
3387     gsh.printVal(curfmt,vals,optv)
3388     xi += 2+leng-1
3389     outlen += 1
3390     }
3391     continue
3392     }
3393     }
3394     if fch == '.' {
3395     hi, leng := scanInt(fmts[xi+1:])
3396     if 0 < leng {
3397     if hi < len(gsh.iValues) {
3398     gsh.printVal(curfmt,gsh.iValues[hi],optv)
3399     outlen += 1 // should be the real length
3400     }else{
3401     fmt.Printf("((out-range))")
3402     }
3403     xi += leng
3404     continue;
3405     }
3406     }
3407     fmt.Printf("%c",fch)
3408     outlen += 1
3409     }
3410     }else{
3411     //fmt.Printf("--D-- print {s}\n")
3412     for i,v := range list {
3413     if 0 < i {
3414     fmt.Printf(div)
3415     }
3416     gsh.printVal(curfmt,v,optv)
3417     outlen += 1
3418     }
3419     }
3420     if 0 < outlen {
3421     fmt.Printf("\n")
3422     }
3423     }
3424     func (gsh*GshContext)Scanv(argv []string){
3425     //fmt.Printf("--D-- Scanv(%v)\n",argv)
3426     if len(argv) == 1 {
3427     return
3428     }
3429     argv = argv[1:]
3430     fmts := ""
3431     if strBegins(argv[0],"-F") {
3432     fmts = argv[0]
3433     gsh.iDelimiter = fmts
3434     argv = argv[1:]
3435     }
3436     input := strings.Join(argv, " ")
3437     if fmts == "" { // simple decomposition
3438     v := scanv(input)
3439     gsh.iValues = v
3440     //fmt.Printf("%v\n",strings.Join(v,","))
3441     }else{
3442     v := make([]string,8)
3443     n,err = fmt.Sscanf(input,fmts,&v[0],&v[1],&v[2],&v[3])
3444     fmt.Printf("--D-- Sscanf ->(%v) n=%d err=(%v)\n",v,n,err)
3445     gsh.iValues = v
3446     }
3447     }
3448     func (gsh*GshContext)Printv(argv []string){
3449     if false { //@@@
3450     fmt.Printf("%v\n",strings.Join(argv[1:], " "))
3451     return
3452     }
3453     //fmt.Printf("--D-- Printv(%v)\n",argv)
3454     //fmt.Printf("%v\n",strings.Join(gsh.iValues,","))
3455     div := gsh.iDelimiter
3456     fmts := ""
3457     argv = argv[1:]
3458     if 0 < len(argv) {
3459     if strBegins(argv[0],"-F") {
3460     div = argv[0][2:]
3461     argv = argv[1:]
3462     }
3463     }
3464
3465     optv := []string{}
3466     for _,v := range argv {
3467     if strBegins(v,"-"){
3468     optv = append(optv,v)
3469     argv = argv[1:]
3470     }else{
3471     break;
3472     }

```

```

3473     }
3474     if 0 < len(argv) {
3475         fmts = strings.Join(argv, " ")
3476     }
3477     gsh.printf(fmts,div,argv,optv,gsh.iValues)
3478 }
3479 func (gsh*GshContext)Basename(argv[]string){
3480     for i,v := range gsh.iValues {
3481         gsh.iValues[i] = filepath.Base(v)
3482     }
3483 }
3484 func (gsh*GshContext)Sortv(argv[]string){
3485     sv := gsh.iValues
3486     sort.Slice(sv , func(i,j int) bool {
3487         return sv[i] < sv[j]
3488     })
3489 }
3490 func (gsh*GshContext)Shiftv(argv[]string){
3491     vi := len(gsh.iValues)
3492     if 0 < vi {
3493         if isin("-r",argv) {
3494             top := gsh.iValues[0]
3495             gsh.iValues = append(gsh.iValues[1:],top)
3496         }else{
3497             gsh.iValues = gsh.iValues[1:]
3498         }
3499     }
3500 }
3501 }
3502 func (gsh*GshContext)Enq(argv[]string){
3503 }
3504 func (gsh*GshContext)Deq(argv[]string){
3505 }
3506 func (gsh*GshContext)Push(argv[]string){
3507     gsh.iValStack = append(gsh.iValStack,argv[1:])
3508     fmt.Printf("depth=%d\n",len(gsh.iValStack))
3509 }
3510 func (gsh*GshContext)Dump(argv[]string){
3511     for i,v := range gsh.iValStack {
3512         fmt.Printf("%d %v\n",i,v)
3513     }
3514 }
3515 func (gsh*GshContext)Pop(argv[]string){
3516     depth := len(gsh.iValStack)
3517     if 0 < depth {
3518         v := gsh.iValStack[depth-1]
3519         if isin("-cat",argv){
3520             gsh.iValues = append(gsh.iValues,v...)
3521         }else{
3522             gsh.iValues = v
3523         }
3524         gsh.iValStack = gsh.iValStack[0:depth-1]
3525         fmt.Printf("depth=%d %s\n",len(gsh.iValStack),gsh.iValues)
3526     }else{
3527         fmt.Printf("depth=%d\n",depth)
3528     }
3529 }
3530 }
3531 // <a name="interpreter">Command Interpreter</a>
3532 func (gshCtx*GshContext)gshellv(argv []string) (fin bool) {
3533     fin = false
3534 }
3535 if gshCtx.CmdTrace { fmt.Fprintf(os.Stderr,"--I-- gshellv(%d)\n",len(argv)) }
3536 if len(argv) <= 0 {
3537     return false
3538 }
3539 xargv := []string{}
3540 for ai := 0; ai < len(argv); ai++ {
3541     xargv = append(xargv, strsubst(gshCtx,argv[ai],false))
3542 }
3543 argv = xargv
3544 if false {
3545     for ai := 0; ai < len(argv); ai++ {
3546         fmt.Printf("[%d] %s [%d]T\n",
3547             ai,argv[ai],len(argv[ai]),argv[ai])
3548     }
3549 }
3550 cmd := argv[0]
3551 if gshCtx.CmdTrace { fmt.Fprintf(os.Stderr,"--I-- gshellv(%d)%v\n",len(argv),argv) }
3552 switch { // https://tour.golang.org/flowcontrol/11
3553 case cmd == "":
3554     gshCtx.xPwd([]string{}); // empty command
3555 case cmd == "-x":
3556     gshCtx.CmdTrace = ! gshCtx.CmdTrace
3557 case cmd == "-xt":
3558     gshCtx.CmdTime = ! gshCtx.CmdTime
3559 case cmd == "-ot":
3560     gshCtx.sconnect(true, argv)
3561 case cmd == "-ou":
3562     gshCtx.sconnect(false, argv)
3563 case cmd == "-it":
3564     gshCtx.saccept(true , argv)
3565 case cmd == "-iu":
3566     gshCtx.saccept(false, argv)
3567 case cmd == "-i" || cmd == "<" || cmd == "-o" || cmd == ">" || cmd == "-a" || cmd == ">>" || cmd == "-s" || cmd == "><":
3568     gshCtx.redirect(argv)
3569 case cmd == "|":
3570     gshCtx.fromPipe(argv)
3571 case cmd == "args":
3572     gshCtx.Args(argv)
3573 case cmd == "bg" || cmd == "-bg":
3574     rfin := gshCtx.inBackground(argv[1:])
3575     return rfin
3576 case cmd == "-bn":
3577     gshCtx.Basename(argv)
3578 case cmd == "call":
3579     _ = gshCtx.excommand(false,argv[1:])
3580 case cmd == "cd" || cmd == "chdir":
3581     gshCtx.xChdir(argv);
3582 case cmd == "-cksum":
3583     gshCtx.xFind(argv)
3584 case cmd == "-sum":
3585     gshCtx.xFind(argv)
3586 case cmd == "-sumtest":
3587     str := ""
3588     if 1 < len(argv) { str = argv[1] }
3589     crc := strCRC32(str,uint64(len(str)))
3590     fprintf(stderr,"%v %v\n",crc,len(str))
3591 case cmd == "close":
3592     gshCtx.xClose(argv)
3593 case cmd == "gcp":
3594     gshCtx.FileCopy(argv)
3595 case cmd == "dec" || cmd == "decode":
3596     gshCtx.Dec(argv)

```

```

3597 case cmd == "#define":
3598 case cmd == "dic" || cmd == "d":
3599     xDic(argv)
3600 case cmd == "dump":
3601     gshCtx.Dump(argv)
3602 case cmd == "echo" || cmd == "e":
3603     echo(argv,true)
3604 case cmd == "enc" || cmd == "encode":
3605     gshCtx.Enc(argv)
3606 case cmd == "env":
3607     env(argv)
3608 case cmd == "eval":
3609     xEval(argv[1:],true)
3610 case cmd == "ev" || cmd == "events":
3611     dumpEvents(argv)
3612 case cmd == "exec":
3613     _,_ = gshCtx.excommand(true,argv[1:])
3614     // should not return here
3615 case cmd == "exit" || cmd == "quit":
3616     // write Result code EXIT to 3>
3617     return true
3618 case cmd == "fds":
3619     // dump the attributes of fds (of other process)
3620 case cmd == "-find" || cmd == "fin" || cmd == "ufind" || cmd == "uf":
3621     gshCtx.xFind(argv[1:])
3622 case cmd == "fu":
3623     gshCtx.xFind(argv[1:])
3624 case cmd == "fork":
3625     // mainly for a server
3626 case cmd == "-gen":
3627     gshCtx.gen(argv)
3628 case cmd == "-go":
3629     gshCtx.xGo(argv)
3630 case cmd == "-grep":
3631     gshCtx.xFind(argv)
3632 case cmd == "gdeq":
3633     gshCtx.Deq(argv)
3634 case cmd == "genq":
3635     gshCtx.Bng(argv)
3636 case cmd == "gpop":
3637     gshCtx.Pop(argv)
3638 case cmd == "gpush":
3639     gshCtx.Push(argv)
3640 case cmd == "history" || cmd == "hi": // hi should be alias
3641     gshCtx.xHistory(argv)
3642 case cmd == "jobs":
3643     gshCtx.xJobs(argv)
3644 case cmd == "lnsp" || cmd == "nlspl":
3645     gshCtx.SplitLine(argv)
3646 case cmd == "-ls":
3647     gshCtx.xFind(argv)
3648 case cmd == "nop":
3649     // do nothing
3650 case cmd == "pipe":
3651     gshCtx.xOpen(argv)
3652 case cmd == "plug" || cmd == "plugin" || cmd == "pin":
3653     gshCtx.xPlugin(argv[1:])
3654 case cmd == "print" || cmd == "-pr":
3655     // output internal slice // also sprintf should be
3656     gshCtx.Printv(argv)
3657 case cmd == "ps":
3658     gshCtx.xPs(argv)
3659 case cmd == "pstitle":
3660     // to be gsh.title
3661 case cmd == "rexeed" || cmd == "rexd":
3662     gshCtx.RexecServer(argv)
3663 case cmd == "rexec" || cmd == "rex":
3664     gshCtx.RexecClient(argv)
3665 case cmd == "repeat" || cmd == "rep": // repeat cond command
3666     gshCtx.repeat(argv)
3667 case cmd == "replay":
3668     gshCtx.xReplay(argv)
3669 case cmd == "scan":
3670     // scan input (or so in fscanf) to internal slice (like Files or map)
3671     gshCtx.Scanv(argv)
3672 case cmd == "set":
3673     // set name ...
3674 case cmd == "serv":
3675     gshCtx.httpServer(argv)
3676 case cmd == "shift":
3677     gshCtx.Shiftv(argv)
3678 case cmd == "sleep":
3679     gshCtx.sleep(argv)
3680 case cmd == "-sort":
3681     gshCtx.Sortv(argv)
3682
3683 case cmd == "j" || cmd == "join":
3684     gshCtx.RJoin(argv)
3685 case cmd == "a" || cmd == "alpa":
3686     gshCtx.Rexec(argv)
3687 case cmd == "jcd" || cmd == "jchdir":
3688     gshCtx.Rchdir(argv)
3689 case cmd == "jget":
3690     gshCtx.Rget(argv)
3691 case cmd == "jls":
3692     gshCtx.Rls(argv)
3693 case cmd == "jput":
3694     gshCtx.Rput(argv)
3695 case cmd == "jpwd":
3696     gshCtx.Rpwd(argv)
3697
3698 case cmd == "time":
3699     fin = gshCtx.xTime(argv)
3700 case cmd == "ungets":
3701     if 1 < len(argv) {
3702         ungets(argv[1]+"\\n")
3703     }else{
3704     }
3705 case cmd == "pwd":
3706     gshCtx.xPwd(argv);
3707 case cmd == "ver" || cmd == "-ver" || cmd == "version":
3708     gshCtx.showVersion(argv)
3709 case cmd == "where":
3710     // data file or so?
3711 case cmd == "which":
3712     which("PATH",argv);
3713 case cmd == "gj" && 1 < len(argv) && argv[1] == "listen":
3714     go gj_server(argv[1:]);
3715 case cmd == "gj" && 1 < len(argv) && argv[1] == "serve":
3716     go gj_server(argv[1:]);
3717 case cmd == "gj" && 1 < len(argv) && argv[1] == "join":
3718     go gj_client(argv[1:]);
3719 case cmd == "gj":
3720     jsend(argv);

```

```

3721     case cmd == "jsend":
3722         jsend(argv);
3723     default:
3724         if gshCtx.whichPlugin(cmd, []string{"-s"}) != nil {
3725             gshCtx.xPlugin(argv)
3726         }else{
3727             notfound, _ := gshCtx.excommand(false, argv)
3728             if notfound {
3729                 fmt.Printf("--E-- command not found (%v)\n", cmd)
3730             }
3731         }
3732     }
3733     return fin
3734 }
3735
3736 func (gsh*GshContext)gshellll(gline string) (rfin bool) {
3737     argv := strings.Split(string(gline), " ")
3738     fin := gsh.gshelllv(argv)
3739     return fin
3740 }
3741
3742 func (gsh*GshContext)tgshellll(gline string)(xfn bool){
3743     start := time.Now()
3744     fin := gsh.gshellll(gline)
3745     end := time.Now()
3746     elps := end.Sub(start);
3747     if gsh.CmdTime {
3748         fmt.Printf("--T-- " + time.Now().Format(time.Stamp) + " (%d.%09ds)\n",
3749             elps/1000000000, elps%1000000000)
3750     }
3751     return fin
3752 }
3753
3754 func Ttyid() (int) {
3755     fi, err := os.Stdin.Stat()
3756     if err != nil {
3757         return 0;
3758     }
3759     //fmt.Printf("Stdin: %v Dev=%d\n",
3760     // fi.Mode(), fi.Mode() & os.ModeDevice)
3761     if (fi.Mode() & os.ModeDevice) != 0 {
3762         stat := syscall.Stat_t{};
3763         err := syscall.Fstat(0, &stat)
3764         if err != nil {
3765             //fmt.Printf("--I-- Stdin: (%v)\n", err)
3766         }else{
3767             //fmt.Printf("--I-- Stdin: rdev=%d %d\n",
3768             // stat.Rdev&0xFF, stat.Rdev);
3769             //fmt.Printf("--I-- Stdin: tty%d\n", stat.Rdev&0xFF);
3770             return int(stat.Rdev & 0xFF)
3771         }
3772     }
3773     return 0
3774 }
3775
3776 func (gshCtx *GshContext) ttyfile() string {
3777     //fmt.Printf("--I-- GSH_HOME=%s\n", gshCtx.GshHomeDir)
3778     ttyfile := gshCtx.GshHomeDir + "/" + "gsh-tty" +
3779         fmt.Sprintf("%02d", gshCtx.TerminalId)
3780     //strconv.Itoa(gshCtx.TerminalId)
3781     //fmt.Printf("--I-- ttyfile=%s\n", ttyfile)
3782     return ttyfile
3783 }
3784
3785 func (gshCtx *GshContext) ttyline>(*os.File){
3786     file, err := os.OpenFile(gshCtx.ttyfile(), os.O_RDWR|os.O_CREATE|os.O_TRUNC, 0600)
3787     if err != nil {
3788         fmt.Printf("--F-- cannot open %s (%s)\n", gshCtx.ttyfile(), err)
3789         return file;
3790     }
3791     return file
3792 }
3793
3794 func (gshCtx *GshContext)getline(hix int, skipping bool, prevline string) (string) {
3795     if( skipping ){
3796         reader := bufio.NewReaderSize(os.Stdin, LINESIZE)
3797         line, _, _ := reader.ReadLine()
3798         return string(line)
3799     }else
3800     if true {
3801         return xgetline(hix, prevline, gshCtx)
3802     }
3803     /*
3804     else
3805     if( with_exgetline && gshCtx.GetLine != "" ){
3806         //var xhix int64 = int64(hix); // cast
3807         newenv := os.Environ()
3808         newenv = append(newenv, "GSH_LINENO="+strconv.FormatInt(int64(hix), 10) )
3809
3810         tty := gshCtx.ttyline()
3811         tty.WriteString(prevline)
3812         Pa := os.ProcAttr {
3813             // start dir
3814             newenv, //os.Environ(),
3815             []*os.File{os.Stdin, os.Stdout, os.Stderr, tty},
3816             nil,
3817         }
3818         //fmt.Printf("--I-- getline=%s // %s\n", gsh_getline[0], gshCtx.GetLine)
3819         proc, err := os.StartProcess(gsh_getline[0], []string{"getline", "getline"}, &Pa)
3820         if err != nil {
3821             fmt.Printf("--F-- getline process error (%v)\n", err)
3822             // for ; ; { }
3823             return "exit (getline program failed)"
3824         }
3825         //stat, err := proc.Wait()
3826         proc.Wait()
3827         buff := make([]byte, LINESIZE)
3828         count, err := tty.Read(buff)
3829         //_, err = tty.Read(buff)
3830         //fmt.Printf("--D-- getline (%d)\n", count)
3831         if err != nil {
3832             if ! (count == 0) { // && err.String() == "EOF" } {
3833                 fmt.Printf("--E-- getline error (%s)\n", err)
3834             }
3835         }else{
3836             //fmt.Printf("--I-- getline OK %s\n", buff)
3837         }
3838         tty.Close()
3839         gline := string(buff[0:count])
3840         return gline
3841     }else
3842     /*
3843     {
3844         // if isatty {
3845         fmt.Printf("!\d", hix)
3846         fmt.Print(PROMPT)
3847         // }
3848         reader := bufio.NewReaderSize(os.Stdin, LINESIZE)
3849         line, _, _ := reader.ReadLine()

```

```

3845     return string(line)
3846 }
3847 }
3848
3849 //== begin ===== getline
3850 /*
3851 * getline.c
3852 * 2020-0819 extracted from dog.c
3853 * getline.go
3854 * 2020-0822 ported to Go
3855 */
3856 /*
3857 package main // getline main
3858 import (
3859     "fmt" // <a href="https://golang.org/pkg/fmt/">fmt</a>
3860     "strings" // <a href="https://golang.org/pkg/strings/">strings</a>
3861     "os" // <a href="https://golang.org/pkg/os/">os</a>
3862     "syscall" // <a href="https://golang.org/pkg/syscall/">syscall</a>
3863     //"bytes" // <a href="https://golang.org/pkg/bytes/">bytes</a>
3864     //"os/exec" // <a href="https://golang.org/pkg/os/exec/">os/exec</a>
3865 )
3866 */
3867
3868 // C language compatibility functions
3869 var errno = 0
3870 var stdin *os.File = os.Stdin
3871 var stdout *os.File = os.Stdout
3872 var stderr *os.File = os.Stderr
3873 var EOF = -1
3874 var NULL = 0
3875 type FILE os.File
3876 type StrBuff []byte
3877 var NULL_FP *os.File = nil
3878 var NULLSP = 0
3879 //var LINESIZE = 1024
3880
3881 func system(cmdstr string)(int){
3882     PA := syscall.ProcAttr {
3883         "", // the starting directory
3884         os.Environ(),
3885         []uintptr{os.Stdin.Fd(),os.Stdout.Fd(),os.Stderr.Fd()},
3886         nil,
3887     }
3888     argv := strings.Split(cmdstr, " ")
3889     pid,err := syscall.ForkExec(argv[0],argv,&PA)
3890     if( err != nil ){
3891         fmt.Printf("--E-- syscall(%v) err(%v)\n",cmdstr,err)
3892     }
3893     syscall.Wait4(pid,nil,0,nil)
3894 }
3895 /*
3896 argv := strings.Split(cmdstr, " ")
3897 fmt.Fprintf(os.Stderr,"--I-- system(%v)\n",argv)
3898 //cmd := exec.Command(argv[0]:...)
3899 cmd := exec.Command(argv[0],argv[1],argv[2])
3900 cmd.Stdin = strings.NewReader("output of system")
3901 var out bytes.Buffer
3902 cmd.Stdout = &out
3903 var serr bytes.Buffer
3904 cmd.Stderr = &serr
3905 err := cmd.Run()
3906 if err != nil {
3907     fmt.Fprintf(os.Stderr,"--E-- system(%v)err(%v)\n",argv,err)
3908     fmt.Printf("ERR:%s\n",serr.String())
3909 }else{
3910     fmt.Printf("%s",out.String())
3911 }
3912 */
3913 return 0
3914 }
3915 func atoi(str string)(ret int){
3916     ret,err := fmt.Sscanf(str,"%d",&ret)
3917     if err == nil {
3918         return ret
3919     }else{
3920         // should set errno
3921         return 0
3922     }
3923 }
3924 func getenv(name string)(string){
3925     val,got := os.LookupEnv(name)
3926     if got {
3927         return val
3928     }else{
3929         return "?"
3930     }
3931 }
3932 func strcpy(dst StrBuff, src string){
3933     var i int
3934     srcb := []byte(src)
3935     for i = 0; i < len(src) && srcb[i] != 0; i++ {
3936         dst[i] = srcb[i]
3937     }
3938     dst[i] = 0
3939 }
3940 func xstrcpy(dst StrBuff, src StrBuff){
3941     dst = src
3942 }
3943 func strcat(dst StrBuff, src StrBuff){
3944     dst = append(dst,src...)
3945 }
3946 func strdup(str StrBuff)(string){
3947     return string(str[0:strlen(str)])
3948 }
3949 func strlen(str string)(int){
3950     return len(str)
3951 }
3952 func strlen(str StrBuff)(int){
3953     var i int
3954     for i = 0; i < len(str) && str[i] != 0; i++ {
3955     }
3956     return i
3957 }
3958 func sizeof(data StrBuff)(int){
3959     return len(data)
3960 }
3961 func isatty(fd int)(ret int){
3962     return 1
3963 }
3964
3965 func fopen(file string,mode string)(fp*os.File){
3966     if mode == "r" {
3967         fp,err := os.Open(file)
3968         if( err != nil ){

```



```

3969         fmt.Printf("--E-- fopen(%s,%s)=(%v)\n",file,mode,err)
3970         return NULL_FP;
3971     }
3972     return fp;
3973 }else{
3974     fp,err := os.OpenFile(file,os.O_RDWR|os.O_CREATE|os.O_TRUNC,0600)
3975     if( err != nil ){
3976         return NULL_FP;
3977     }
3978     return fp;
3979 }
3980 }
3981 func fclose(fp*os.File){
3982     fp.Close()
3983 }
3984 func fflush(fp *os.File)(int){
3985     return 0
3986 }
3987 func fgetc(fp*os.File)(int){
3988     var buf [1]byte
3989     _,err := fp.Read(buf[0:1])
3990     if( err != nil ){
3991         return EOF;
3992     }else{
3993         return int(buf[0])
3994     }
3995 }
3996 func sfgets(str*string, size int, fp*os.File)(int){
3997     buf := make(StrBuff,size)
3998     var ch int
3999     var i int
4000     for i = 0; i < len(buf)-1; i++ {
4001         ch = fgetc(fp)
4002         //fprintf(stderr,"--fgets %d/%d %X\n",i,len(buf),ch)
4003         if( ch == EOF ){
4004             break;
4005         }
4006         buf[i] = byte(ch);
4007         if( ch == '\n' ){
4008             break;
4009         }
4010     }
4011     buf[i] = 0
4012     //fprintf(stderr,"--fgets %d/%d (%s)\n",i,len(buf),buf[0:i])
4013     return i
4014 }
4015 func fgets(buf StrBuff, size int, fp*os.File)(int){
4016     var ch int
4017     var i int
4018     for i = 0; i < len(buf)-1; i++ {
4019         ch = fgetc(fp)
4020         //fprintf(stderr,"--fgets %d/%d %X\n",i,len(buf),ch)
4021         if( ch == EOF ){
4022             break;
4023         }
4024         buf[i] = byte(ch);
4025         if( ch == '\n' ){
4026             break;
4027         }
4028     }
4029     buf[i] = 0
4030     //fprintf(stderr,"--fgets %d/%d (%s)\n",i,len(buf),buf[0:i])
4031     return i
4032 }
4033 func fputc(ch int , fp*os.File)(int){
4034     var buf [1]byte
4035     buf[0] = byte(ch)
4036     fp.Write(buf[0:1])
4037     return 0
4038 }
4039 func fputs(buf StrBuff, fp*os.File)(int){
4040     fp.Write(buf)
4041     return 0
4042 }
4043 func xputcss(str string, fp*os.File)(int){
4044     return fputs([]byte(str),fp)
4045 }
4046 func sscanf(str StrBuff,fmts string, params ...interface{})(int){
4047     fmt.Sscanf(string(str[0:strlen(str)]),fmts,params...)
4048     return 0
4049 }
4050 func fprintf(fp*os.File,fmts string, params ...interface{})(int){
4051     fmt.Fprintf(fp,fmts,params...)
4052     return 0
4053 }
4054 }
4055 // <a name="IME">Command Line IME</a>
4056 //----- MyIME
4057 var MyIMEVER = "MyIME/0.0.2";
4058 type Romkana struct {
4059     dic string // dictionaly ID
4060     pat string // input pattern
4061     out string // output pattern
4062     hit int64 // count of hit and used
4063 }
4064 var dicents = 0
4065 var romkana [1024]Romkana
4066 var Romkan []Romkana
4067
4068 func isinDic(str string)(int){
4069     for i,v := range Romkan {
4070         if v.pat == str {
4071             return i
4072         }
4073     }
4074     return -1
4075 }
4076 const (
4077     DIC_COM_LOAD = "im"
4078     DIC_COM_DUMP = "s"
4079     DIC_COM_LIST = "ls"
4080     DIC_COM_ENA = "en"
4081     DIC_COM_DIS = "di"
4082 )
4083 func helpDic(argv []string){
4084     out := stderr
4085     cmd := ""
4086     if 0 < len(argv) { cmd = argv[0] }
4087     fprintf(out,"-- %v Usage\n",cmd)
4088     fprintf(out,"... Commands\n")
4089     fprintf(out,"... %v %v [dicName] [dicURL] -- Import dictionary\n",cmd,DIC_COM_LOAD)
4090     fprintf(out,"... %v %v [pattern] -- Search in dictionary\n",cmd,DIC_COM_DUMP)
4091     fprintf(out,"... %v %v [dicName] -- List dictionaries\n",cmd,DIC_COM_LIST)
4092     fprintf(out,"... %v %v [dicName] -- Disable dictionaries\n",cmd,DIC_COM_DIS)

```

```

4093     fprintf(out, "... %v %-3v [dicName] -- Enable dictionaries\n", cmd, DIC_COM_ENA)
4094     fprintf(out, "... Keys ... %v\n", "ESC can be used for '\\')
4095     fprintf(out, "... \\c -- Reverse the case of the last character\n",)
4096     fprintf(out, "... \\i -- Replace input with translated text\n",)
4097     fprintf(out, "... \\j -- On/off translation mode\n",)
4098     fprintf(out, "... \\l -- Force Lower Case\n",)
4099     fprintf(out, "... \\u -- Force Upper Case (software CapsLock)\n",)
4100     fprintf(out, "... \\v -- Show translation actions\n",)
4101     fprintf(out, "... \\x -- Replace the last input character with it Hexa-Decimal\n",)
4102 }
4103 func xDic(argv[]string){
4104     if len(argv) <= 1 {
4105         helpDic(argv)
4106         return
4107     }
4108     argv = argv[1:]
4109     var debug = false
4110     var info = false
4111     var silent = false
4112     var dump = false
4113     var builtin = false
4114     cmd := argv[0]
4115     argv = argv[1:]
4116     opt := ""
4117     arg := ""
4118
4119     if 0 < len(argv) {
4120         arg1 := argv[0]
4121         if arg1[0] == '-' {
4122             switch arg1 {
4123                 default: fmt.Printf("--Ed-- Unknown option(%v)\n", arg1)
4124                     return
4125                 case "-b": builtin = true
4126                 case "-d": debug = true
4127                 case "-s": silent = true
4128                 case "-v": info = true
4129             }
4130             opt = arg1
4131             argv = argv[1:]
4132         }
4133     }
4134
4135     dicName := ""
4136     dicURL := ""
4137     if 0 < len(argv) {
4138         arg = argv[0]
4139         dicName = arg
4140         argv = argv[1:]
4141     }
4142     if 0 < len(argv) {
4143         dicURL = argv[0]
4144         argv = argv[1:]
4145     }
4146     if false {
4147         fprintf(stderr, "--Dd-- com(%v) opt(%v) arg(%v)\n", cmd, opt, arg)
4148     }
4149     if cmd == DIC_COM_LOAD {
4150         //dicType := ""
4151         dicBody := ""
4152         if !builtin && dicName != "" && dicURL == "" {
4153             f, err := os.Open(dicName)
4154             if err == nil {
4155                 dicURL = dicName
4156             }else{
4157                 f, err = os.Open(dicName+".html")
4158                 if err == nil {
4159                     dicURL = dicName+".html"
4160                 }else{
4161                     f, err = os.Open("gshdic-"+dicName+".html")
4162                     if err == nil {
4163                         dicURL = "gshdic-"+dicName+".html"
4164                     }
4165                 }
4166             }
4167             if err == nil {
4168                 var buf = make([]byte, 128*1024)
4169                 count, err := f.Read(buf)
4170                 f.Close()
4171                 if info {
4172                     fprintf(stderr, "--Id-- ReadDic(%v,%v)\n", count, err)
4173                 }
4174                 dicBody = string(buf[0:count])
4175             }
4176         }
4177         if dicBody == "" {
4178             switch arg {
4179                 default:
4180                     dicName = "WorldDic"
4181                     dicURL = "WorldDic"
4182                     if info {
4183                         fprintf(stderr, "--Id-- default dictionary \"%v\"\n",
4184                             dicName);
4185                     }
4186                 case "wnn":
4187                     dicName = "WnnDic"
4188                     dicURL = "WnnDic"
4189                 case "sumomo":
4190                     dicName = "SumomoDic"
4191                     dicURL = "SumomoDic"
4192                 case "sijimi":
4193                     dicName = "SijimiDic"
4194                     dicURL = "SijimiDic"
4195                 case "jkl":
4196                     dicName = "JKLJaDic"
4197                     dicURL = "JA_JKLDic"
4198             }
4199             if debug {
4200                 fprintf(stderr, "--Id-- %v URL=%v\n", dicName, dicURL);
4201             }
4202             dicv := strings.Split(dicURL, ",")
4203             if debug {
4204                 fprintf(stderr, "--Id-- %v encoded data...\n", dicName)
4205                 fprintf(stderr, "Type: %v\n", dicv[0])
4206                 fprintf(stderr, "Body: %v\n", dicv[1])
4207                 fprintf(stderr, "\n")
4208             }
4209             body, _ := base64.StdEncoding.DecodeString(dicv[1])
4210             dicBody = string(body)
4211         }
4212         if info {
4213             fmt.Printf("--Id-- %v %v\n", dicName, dicURL)
4214             fmt.Printf("%s\n", dicBody)
4215         }
4216         if debug {

```



```

4341 var si int;
4342 var sx = len(src);
4343 var di int;
4344 var mi int;
4345 var dstb []byte
4346
4347 for si = 0; si < sx; { // search max. match from the position
4348   if strBegins(src[si:], "%x/") {
4349     // %x/integer/ // s/a/b/
4350     ix := strings.Index(src[si+3:], "/")
4351     if 0 < ix {
4352       var iv int = 0
4353       //fmt.Sscanf(src[si+3:si+3+ix], "%d", &iv)
4354       fmt.Sscanf(src[si+3:si+3+ix], "%v", &iv)
4355       sval := fmt.Sprintf("%x", iv)
4356       bval := []byte(sval)
4357       dstb = append(dstb, bval...)
4358       si = si+3+ix+1
4359       continue
4360     }
4361   }
4362   if strBegins(src[si:], "%d/") {
4363     // %d/integer/ // s/a/b/
4364     ix := strings.Index(src[si+3:], "/")
4365     if 0 < ix {
4366       var iv int = 0
4367       fmt.Sscanf(src[si+3:si+3+ix], "%v", &iv)
4368       sval := fmt.Sprintf("%d", iv)
4369       bval := []byte(sval)
4370       dstb = append(dstb, bval...)
4371       si = si+3+ix+1
4372       continue
4373     }
4374   }
4375   if strBegins(src[si:], "%t") {
4376     now := time.Now()
4377     if true {
4378       date := now.Format(time.Stamp)
4379       dstb = append(dstb, []byte(date)...)
4380       si = si+3
4381     }
4382     continue
4383   }
4384   var maxlen int = 0;
4385   var len int;
4386   mi = -1;
4387   for di = 0; di < dicents; di++ {
4388     len = matchlen(src[si:], romkana[di].pat);
4389     if( maxlen < len ){
4390       maxlen = len;
4391       mi = di;
4392     }
4393   }
4394   if( 0 < maxlen ){
4395     out := romkana[mi].out;
4396     dstb = append(dstb, []byte(out)...);
4397     si += maxlen;
4398   }else{
4399     dstb = append(dstb, src[si])
4400     si += 1;
4401   }
4402 }
4403 return string(dstb)
4404 }
4405 func trans(src string)(int){
4406   dst := convs(src);
4407   xfprintf(dst, stderr);
4408   return 0;
4409 }
4410 //----- LINEEDIT
4411 // "?? at the top of the line means searching history
4412 // "?? at the top of the line means searching history
4413 // "?? at the top of the line means searching history
4414 // should be compatilbe with Telnet
4415 const (
4416   EV_MODE      = 255
4417   EV_IDLE      = 254
4418   EV_TIMEOUT   = 253
4419
4420   GO_UP        = 252 // k
4421   GO_DOWN      = 251 // j
4422   GO_RIGHT     = 250 // l
4423   GO_LEFT      = 249 // h
4424   DEL_RIGHT    = 248 // x
4425   GO_TOPL      = 'A'-0x40 // 0
4426   GO_ENDL      = 'E'-0x40 // $
4427
4428   GO_TOPW      = 239 // b
4429   GO_ENDW      = 238 // e
4430   GO_NEXTW     = 237 // w
4431
4432   GO_FORWCH    = 229 // f
4433   GO_PAIRCH    = 228 // %
4434
4435   GO_DEL       = 219 // d
4436
4437   HI_SRCH_FW   = 209 // /
4438   HI_SRCH_BK   = 208 // ?
4439   HI_SRCH_RFW  = 207 // n
4440   HI_SRCH_RBK  = 206 // N
4441 )
4442
4443 // should return number of octets ready to be read immediately
4444 //fprintf(stderr, "\n--Select(%v %v)\n", err, r.Bits[0])
4445
4446
4447 var EventRecvFd = -1 // file descriptor
4448 var EventSendFd = -1
4449 const EventFdOffset = 1000000
4450 const NormalFdOffset = 100
4451
4452 func putEvent(event int, evarg int){
4453   if true {
4454     if EventRecvFd < 0 {
4455       var pv = []int{-1, -1}
4456       syscall.Pipe(pv)
4457       EventRecvFd = pv[0]
4458       EventSendFd = pv[1]
4459       //fmt.Printf("--De-- EventPipe created[%v, %v]\n", EventRecvFd, EventSendFd)
4460     }
4461   }else{
4462     if EventRecvFd < 0 {
4463       // the document differs from this spec
4464       // https://golang.org/src/syscall/syscall_unix.go?s=8096:8158#L340

```

```

4465     sv, err := syscall.Socketpair(syscall.AF_UNIX, syscall.SOCK_STREAM, 0)
4466     EventRecvFd = sv[0]
4467     EventSendFd = sv[1]
4468     if err != nil {
4469         fmt.Printf("--De-- EventSock created[%v,%v] (%v)\n",
4470             EventRecvFd, EventSendFd, err)
4471     }
4472 }
4473 }
4474 var buf = []byte{ byte(event)}
4475 n, err := syscall.Write(EventSendFd, buf)
4476 if err != nil {
4477     fmt.Printf("--De-- putEvent[%v] (%3v) (%v %v)\n", EventSendFd, event, n, err)
4478 }
4479 }
4480 func ungets(str string){
4481     for _, ch := range str {
4482         putEvent(int(ch), 0)
4483     }
4484 }
4485 func (gsh*GshContext)xReplay(argv []string){
4486     hix := 0
4487     tempo := 1.0
4488     xtempo := 1.0
4489     repeat := 1
4490
4491     for _, a := range argv { // tempo
4492         if strBegins(a, "x") {
4493             fmt.Sscanf(a[1:], "%f", &xtempo)
4494             tempo = 1 / xtempo
4495             //fprintf(stderr, "--Dr-- tempo=[%v]%v\n", a[2:], tempo);
4496         }else
4497         if strBegins(a, "r") { // repeat
4498             fmt.Sscanf(a[1:], "%v", &repeat)
4499         }else
4500         if strBegins(a, "!") {
4501             fmt.Sscanf(a[1:], "%d", &hix)
4502         }else{
4503             fmt.Sscanf(a, "%d", &hix)
4504         }
4505     }
4506     if hix == 0 || len(argv) <= 1 {
4507         hix = len(gsh.CommandHistory)-1
4508     }
4509     fmt.Printf("--Ir-- Replay(!%v x%v r%v)\n", hix, xtempo, repeat)
4510     //dumpEvents(hix)
4511     //gsh.xScanReplay(hix, false, repeat, tempo, argv)
4512     go gsh.xScanReplay(hix, true, repeat, tempo, argv)
4513 }
4514 }
4515 // <a href="https://golang.org/pkg/syscall/#FdSet">syscall.Select</a>
4516 // 2020-0827 GShell-0.2.3
4517 /*
4518 func FpollIn1(fp *os.File, usec int)(uintptr){
4519     nfd := 1
4520
4521     rdv := syscall.FdSet {}
4522     fd1 := fp.Fd()
4523     bank1 := fd1/32
4524     mask1 := int32(1 << fd1)
4525     rdv.Bits[bank1] = mask1
4526
4527     fd2 := -1
4528     bank2 := -1
4529     var mask2 int32 = 0
4530
4531     if 0 <= EventRecvFd {
4532         fd2 = EventRecvFd
4533         nfd = fd2 + 1
4534         bank2 = fd2/32
4535         mask2 = int32(1 << fd2)
4536         rdv.Bits[bank2] |= mask2
4537         //fmt.Printf("--De-- EventPoll mask added [%d][%v][%v]\n", fd2, bank2, mask2)
4538     }
4539
4540     tout := syscall.NsecToTimeval(int64(usec*1000))
4541     //n, err := syscall.Select(nfd, &rdv, nil, nil, & tout) // spec. mismatch
4542     err := syscall.Select(nfd, &rdv, nil, nil, & tout)
4543     if err != nil {
4544         //fmt.Printf("--De-- select() err(%v)\n", err)
4545     }
4546     if err == nil {
4547         if 0 <= fd2 && (rdv.Bits[bank2] & mask2) != 0 {
4548             if false {
4549                 fmt.Printf("--De-- got Event\n")
4550             }
4551             return uintptr(EventFdOffset + fd2)
4552         }else
4553         if (rdv.Bits[bank1] & mask1) != 0 {
4554             return uintptr(NormalFdOffset + fd1)
4555         }else{
4556             return 1
4557         }
4558     }else{
4559         return 0
4560     }
4561 }
4562 */
4563 func fgetcTimeout1(fp *os.File, usec int)(int){
4564     READ1:
4565     //readyFd := FpollIn1(fp, usec)
4566     readyFd := CFpollIn1(fp, usec)
4567     if readyFd < 100 {
4568         return EV_TIMEOUT
4569     }
4570
4571     var buf [1]byte
4572
4573     if EventFdOffset <= readyFd {
4574         fd := int(readyFd-EventFdOffset)
4575         _, err := syscall.Read(fd, buf[0:1])
4576         if( err != nil ){
4577             return EOF;
4578         }else{
4579             if buf[0] == EV_MODE {
4580                 recvEvent(fd)
4581                 goto READ1
4582             }
4583             return int(buf[0])
4584         }
4585     }
4586     _, err := fp.Read(buf[0:1])
4587     if( err != nil ){

```

```

4589     return EOF;
4590 }else{
4591     return int(buf[0])
4592 }
4593 }
4594 }
4595 func visibleChar(ch int)(string){
4596     switch {
4597     case '! ' <= ch && ch <= '-':
4598         return string(ch)
4599     }
4600     switch ch {
4601     case '\t': return "\\s"
4602     case '\n': return "\\n"
4603     case '\r': return "\\r"
4604     case '\t': return "\\t"
4605     }
4606     switch ch {
4607     case 0x00: return "NUL"
4608     case 0x07: return "BEL"
4609     case 0x08: return "BS"
4610     case 0x0E: return "SO"
4611     case 0x0F: return "SI"
4612     case 0x1B: return "ESC"
4613     case 0x7F: return "DEL"
4614     }
4615     switch ch {
4616     case EV_IDLE: return fmt.Sprintf("IDLE")
4617     case EV_MODE: return fmt.Sprintf("MODE")
4618     }
4619     return fmt.Sprintf("%X",ch)
4620 }
4621 func recvEvent(fd int){
4622     var buf = make([]byte,1)
4623     _,_ = syscall.Read(fd,buf[0:1])
4624     if( buf[0] != 0 ){
4625         romkanmode = true
4626     }else{
4627         romkanmode = false
4628     }
4629 }
4630 func (gsh*GshContext)xScanReplay(hix int,replay bool,repeat int,tempo float64,argv[string]){
4631     var Start time.Time
4632     var events = []Event{}
4633     for _,e := range Events {
4634         if hix == 0 || e.CmdIndex == hix {
4635             events = append(events,e)
4636         }
4637     }
4638     elen := len(events)
4639     if 0 < elen {
4640         if events[elen-1].event == EV_IDLE {
4641             events = events[0:elen-1]
4642         }
4643     }
4644     for r := 0; r < repeat; r++ {
4645         for i,e := range events {
4646             nano := e.when.Nanosecond()
4647             micro := nano / 1000
4648             if Start.Second() == 0 {
4649                 Start = time.Now()
4650             }
4651             diff := time.Now().Sub(Start)
4652             if replay {
4653                 if e.event != EV_IDLE {
4654                     putEvent(e.event,0)
4655                     if e.event == EV_MODE { // event with arg
4656                         putEvent(int(e.evarg),0)
4657                     }
4658                 }else{
4659                     fmt.Printf("%7.3fms %#-3v !%-3v [%v.%06d] %3v %02X %-4v %10.3fms\n",
4660                         float64(diff)/1000000.0,
4661                         i,
4662                         e.CmdIndex,
4663                         e.when.Format(time.Stamp),micro,
4664                         e.event,e.event,visibleChar(e.event),
4665                         float64(e.evarg)/1000000.0)
4666                 }
4667             }
4668             if e.event == EV_IDLE {
4669                 d := time.Duration(float64(time.Duration(e.evarg)) * tempo)
4670                 //nsleep(time.Duration(e.evarg))
4671                 nsleep(d)
4672             }
4673         }
4674     }
4675 }
4676 func dumpEvents(arg[string]){
4677     hix := 0
4678     if 1 < len(arg) {
4679         fmt.Sscanf(arg[1],"%d",&hix)
4680     }
4681     for i,e := range Events {
4682         nano := e.when.Nanosecond()
4683         micro := nano / 1000
4684         //if e.event != EV_TIMEOUT {
4685         if hix == 0 || e.CmdIndex == hix {
4686             fmt.Printf("#%-3v !%-3v [%v.%06d] %3v %02X %-4v %10.3fms\n",i,
4687                 e.CmdIndex,
4688                 e.when.Format(time.Stamp),micro,
4689                 e.event,e.event,visibleChar(e.event),float64(e.evarg)/1000000.0)
4690         }
4691         //}
4692     }
4693 }
4694 func fgetcTimeout(fp *os.File,usec int)(int){
4695     ch := fgetcTimeout1(fp,usec)
4696     if ch != EV_TIMEOUT {
4697         now := time.Now()
4698         if 0 < len(Events) {
4699             last := Events[len(Events)-1]
4700             dura := int64(now.Sub(last.when))
4701             Events = append(Events,Event{last.when,EV_IDLE,dura,last.CmdIndex})
4702         }
4703         Events = append(Events,Event{time.Now(),ch,0,CmdIndex})
4704     }
4705     return ch
4706 }
4707 }
4708 var AtConsoleLineTop = true
4709 var TtyMaxCol = 72 // to be obtained by ioctl1
4710 var EscTimeout = (100*1000)
4711 var {
4712     MODE_VicMode    bool    // vi compatible command mode

```

```

4713 MODE_ShowMode bool
4714 romkanmode bool // shown translation mode, the mode to be retained
4715 MODE_Recursive bool // recursive translation
4716 MODE_CapsLock bool // software CapsLock
4717 MODE_LowerLock bool // force lower-case character lock
4718 MODE_ViInsert int // visible insert mode, should be like "I" icon in X Window
4719 MODE_ViTrace bool // output newline before translation
4720 )
4721 type IInput struct {
4722     lno int
4723     lastlno int
4724     pch []int // input queue
4725     prompt string
4726     line string
4727     right string
4728     inJmode bool
4729     pinJmode bool
4730     waitingMeta string // waiting meta character
4731     LastCmd string
4732 }
4733 func (iin*IInput)Getc(timeoutUs int)(int){
4734     ch1 := EOF
4735     ch2 := EOF
4736     ch3 := EOF
4737     if( 0 < len(iin.pch) ){ // deQ
4738         ch1 = iin.pch[0]
4739         iin.pch = iin.pch[1:]
4740     }else{
4741         ch1 = fgetcTimeout(stdin,timeoutUs);
4742     }
4743     if( ch1 == 033 ){ // escape sequence
4744         ch2 = fgetcTimeout(stdin,EscTimeout);
4745         if( ch2 == EV_TIMEOUT ){
4746             }else{
4747                 ch3 = fgetcTimeout(stdin,EscTimeout);
4748                 if( ch3 == EV_TIMEOUT ){
4749                     iin.pch = append(iin.pch,ch2) // enQ
4750                 }else{
4751                     switch( ch2 ){
4752                         default:
4753                             iin.pch = append(iin.pch,ch2) // enQ
4754                             iin.pch = append(iin.pch,ch3) // enQ
4755                         case '[':
4756                             switch( ch3 ){
4757                                 case 'A': ch1 = GO_UP; // ^
4758                                 case 'B': ch1 = GO_DOWN; // v
4759                                 case 'C': ch1 = GO_RIGHT; // >
4760                                 case 'D': ch1 = GO_LEFT; // <
4761                                 case '3':
4762                                     ch4 := fgetcTimeout(stdin,EscTimeout);
4763                                     if( ch4 == '-' ){
4764                                         //fprintf(stderr,"x[%02X %02X %02X %02X]\n",ch1,ch2,ch3,ch4);
4765                                         ch1 = DEL_RIGHT
4766                                     }
4767                                 }
4768                             case '\\':
4769                                 //ch4 := fgetcTimeout(stdin,EscTimeout);
4770                                 //fprintf(stderr,"y[%02X %02X %02X %02X]\n",ch1,ch2,ch3,ch4);
4771                                 switch( ch3 ){
4772                                     case '-': ch1 = DEL_RIGHT
4773                                 }
4774                             }
4775                         }
4776                 }
4777             }
4778         return ch1
4779     }
4780 func (inn*IInput)clearline(){
4781     var i int
4782     fprintf(stderr,"\\r");
4783     // should be ANSI ESC sequence
4784     for i = 0; i < TtyMaxCol; i++ { // to the max. position in this input action
4785         fputc(' ',os.Stderr);
4786     }
4787     fprintf(stderr,"\\r");
4788 }
4789 func (iin*IInput)Redraw(){
4790     redraw(iin,iin.lno,iin.line,iin.right)
4791 }
4792 func redraw(iin *IInput,lno int,line string,right string){
4793     inMeta := false
4794     showMode := ""
4795     showMeta := "" // visible Meta mode on the cursor position
4796     showLino := fmt.Sprintf("%d!", lno)
4797     InsertMark := "" // in visible insert mode
4798
4799     if MODE_VicMode {
4800     }else
4801     if 0 < len(iin.right) {
4802         InsertMark = " "
4803     }
4804
4805     if( 0 < len(iin.waitingMeta) ){
4806         inMeta = true
4807         if iin.waitingMeta[0] != 033 {
4808             showMeta = iin.waitingMeta
4809         }
4810     }
4811     if( romkanmode ){
4812         //romkanmark = " *";
4813     }else{
4814         //romkanmark = "";
4815     }
4816     if MODE_ShowMode {
4817         romkan := "-"
4818         inmeta := "-"
4819         inveri := ""
4820         if MODE_CapsLock {
4821             inmeta = "A"
4822         }
4823         if MODE_LowerLock {
4824             inmeta = "a"
4825         }
4826         if MODE_ViTrace {
4827             inveri = "v"
4828         }
4829         if MODE_VicMode {
4830             inveri = ":"
4831         }
4832         if romkanmode {
4833             romkan = "\343\201\202"
4834             if MODE_CapsLock {
4835                 inmeta = "R"
4836             }else{

```

```

4837         inmeta = "r"
4838     }
4839 }
4840 if inMeta {
4841     inmeta = ""
4842 }
4843 showMode = "["+romkan+inmeta+inveri+"]";
4844 }
4845 Pre := "\r" + showMode + showLino
4846 Output := ""
4847 Left := ""
4848 Right := ""
4849 if romkanmode {
4850     Left = convs(line)
4851     Right = InsertMark+convs(right)
4852 }else{
4853     Left = line
4854     Right = InsertMark+right
4855 }
4856 Output = Pre+Left
4857 if MODE_ViTrace {
4858     Output += iin.LastCmd
4859 }
4860 Output += showMeta+Right
4861 for len(Output) < TtyMaxCol { // to the max. position that may be dirty
4862     Output += " "
4863     // should be ANSI ESC sequence
4864     // not necessary just after newline
4865 }
4866 Output += Pre+Left+showMeta // to set the cursor to the current input position
4867 fprintf(stderr,"%s",Output)
4868
4869 if MODE_ViTrace {
4870     if 0 < len(iin.LastCmd) {
4871         iin.LastCmd = ""
4872         fprintf(stderr,"\r\n")
4873     }
4874 }
4875 AtConsoleLineTop = false
4876 }
4877 // <a href="https://golang.org/pkg/unicode/utf8/">utf8</a>
4878 func delHeadChar(str string)(rline string,head string){
4879     _,clen := utf8.DecodeRune([]byte(str))
4880     head = string(str[0:clen])
4881     return str[clen:],head
4882 }
4883 func delTailChar(str string)(rline string, last string){
4884     var i = 0
4885     var clen = 0
4886     for {
4887         _,siz := utf8.DecodeRune([]byte(str)[i:])
4888         if siz <= 0 { break }
4889         clen = siz
4890         i += siz
4891     }
4892     last = str[len(str)-clen:]
4893     return str[0:len(str)-clen],last
4894 }
4895
4896 // 3> for output and history
4897 // 4> for keylog?
4898 // <a name="getline">Command Line Editor</a>
4899 func xgetline(lno int, prevline string, gsh*GshContext)(string){
4900     var iin IInput
4901     iin.lastlno = lno
4902     iin.lno = lno
4903
4904     CmdIndex = len(gsh.CommandHistory)
4905     if( isatty(0) == 0 ){
4906         if( sfgets(&iin.line,LINESIZE,stdin) == NULL ){
4907             iin.line = "exit\n";
4908         }else{
4909             return iin.line
4910         }
4911     }
4912     if( true ){
4913         //var pts string;
4914         //pts = ptsname(0);
4915         //pts = ttyname(0);
4916         //fprintf(stderr,"--pts[0] = %s\n",pts?pts:"?");
4917     }
4918     if( false ){
4919         fprintf(stderr,"! ");
4920         fflush(stderr);
4921         sfgets(&iin.line,LINESIZE,stdin);
4922         return iin.line
4923     }
4924     system("/bin/stty -echo -icanon");
4925     xline := iin.xgetline1(prevline,gsh)
4926     system("/bin/stty echo sane");
4927     return xline
4928 }
4929 func (iin*IInput)Translate(cmdch int){
4930     romkanmode = !romkanmode;
4931     if MODE_ViTrace {
4932         fprintf(stderr,"%v\r\n",string(cmdch));
4933     }else
4934     if( cmdch == 'J' ){
4935         fprintf(stderr,"J\r\n");
4936         iin.inJmode = true
4937     }
4938     iin.Redraw();
4939     loadDefaultDic(cmdch);
4940     iin.Redraw();
4941 }
4942 func (iin*IInput)Replace(cmdch int){
4943     iin.LastCmd = fmt.Sprintf("%v",string(cmdch))
4944     iin.Redraw();
4945     loadDefaultDic(cmdch);
4946     dst := convs(iin.Line+iin.Right);
4947     iin.Line = dst
4948     iin.Right = ""
4949     if( cmdch == 'I' ){
4950         fprintf(stderr,"I\r\n");
4951         iin.inJmode = true
4952     }
4953     iin.Redraw();
4954 }
4955 // aa 12 alal
4956 func isAlpha(ch rune)(bool){
4957     if 'a' <= ch && ch <= 'z' || 'A' <= ch && ch <= 'Z' {
4958         return true
4959     }
4960     return false

```



```

4961 }
4962 func isAlnum(ch rune)(bool){
4963     if 'a' <= ch && ch <= 'z' || 'A' <= ch && ch <= 'Z' {
4964         return true
4965     }
4966     if '0' <= ch && ch <= '9' {
4967         return true
4968     }
4969     return false
4970 }
4971 }
4972 // 0.2.8 2020-0901 created
4973 // <a href="https://golang.org/pkg/unicode/utf8/#DecodeRuneInString">DecodeRuneInString</a>
4974 func (iin*Input)GotoTOPW(){
4975     str := iin.line
4976     i := len(str)
4977     if i <= 0 {
4978         return
4979     }
4980     //i0 := i
4981     i -= 1
4982     lastSize := 0
4983     var lastRune rune
4984     var found = -1
4985     for 0 < i { // skip preamble spaces
4986         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
4987         if !isAlnum(lastRune) { // character, type, or string to be searched
4988             i -= lastSize
4989             continue
4990         }
4991         break
4992     }
4993     for 0 < i {
4994         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
4995         if lastSize <= 0 { continue } // not the character top
4996         if !isAlnum(lastRune) { // character, type, or string to be searched
4997             found = 1
4998             break
4999         }
5000         i -= lastSize
5001     }
5002     if found < 0 && i == 0 {
5003         found = 0
5004     }
5005     if 0 <= found {
5006         if isAlnum(lastRune) { // or non-kana character
5007             }else{ // when positioning to the top o the word
5008                 i += lastSize
5009             }
5010             iin.right = str[i:] + iin.right
5011             if 0 < i {
5012                 iin.line = str[0:i]
5013             }else{
5014                 iin.line = ""
5015             }
5016         }
5017         //fmt.Printf("\n(%d,%d,%d)[%s][%s]\n",i0,i,found,iin.line,iin.right)
5018         //fmt.Printf("") // set debug messae at the end of line
5019     }
5020 // 0.2.8 2020-0901 created
5021 func (iin*Input)GotoENDW(){
5022     str := iin.right
5023     if len(str) <= 0 {
5024         return
5025     }
5026     lastSize := 0
5027     var lastRune rune
5028     var lastW = 0
5029     i := 0
5030     inWord := false
5031
5032     lastRune,lastSize = utf8.DecodeRuneInString(str[0:])
5033     if isAlnum(lastRune) {
5034         r,z := utf8.DecodeRuneInString(str[lastSize:])
5035         if 0 < z && isAlnum(r) {
5036             inWord = true
5037         }
5038     }
5039     for i < len(str) {
5040         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
5041         if lastSize <= 0 { break } // broken data?
5042         if !isAlnum(lastRune) { // character, type, or string to be searched
5043             break
5044         }
5045         lastW = i // the last alnum if in alnum word
5046         i += lastSize
5047     }
5048     if inWord {
5049         goto DISP
5050     }
5051     for i < len(str) {
5052         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
5053         if lastSize <= 0 { break } // broken data?
5054         if isAlnum(lastRune) { // character, type, or string to be searched
5055             break
5056         }
5057         i += lastSize
5058     }
5059     for i < len(str) {
5060         lastRune,lastSize = utf8.DecodeRuneInString(str[i:])
5061         if lastSize <= 0 { break } // broken data?
5062         if !isAlnum(lastRune) { // character, type, or string to be searched
5063             break
5064         }
5065         lastW = i
5066         i += lastSize
5067     }
5068     DISP:
5069     if 0 < lastW {
5070         iin.line = iin.line + str[0:lastW]
5071         iin.right = str[lastW:]
5072     }
5073     //fmt.Printf("\n(%d)[%s][%s]\n",i,iin.line,iin.right)
5074     //fmt.Printf("") // set debug messae at the end of line
5075 }
5076 // 0.2.8 2020-0901 created
5077 func (iin*Input)GotoNEXTW(){
5078     str := iin.right
5079     if len(str) <= 0 {
5080         return
5081     }
5082     lastSize := 0
5083     var lastRune rune
5084     var found = -1

```

```

5085     i := 1
5086     for i < len(str) {
5087         lastRune, lastSize = utf8.DecodeRuneInString(str[i:])
5088         if lastSize <= 0 { break } // broken data?
5089         if !isAlnum(lastRune) { // character, type, or string to be searched
5090             found = i
5091             break
5092         }
5093         i += lastSize
5094     }
5095     if 0 < found {
5096         if isAlnum(lastRune) { // or non-kana character
5097         }else{ // when positioning to the top o the word
5098             found += lastSize
5099         }
5100         iin.line = iin.line + str[0:found]
5101         if 0 < found {
5102             iin.right = str[found:]
5103         }else{
5104             iin.right = ""
5105         }
5106     }
5107     //fmt.Printf("\n(%d)[%s][%s]\n",i,iin.line,iin.right)
5108     //fmt.Printf("") // set debug messae at the end of line
5109 }
5110 // 0.2.8 2020-0902 created
5111 func (iin*IInput)GotoPAIRCH()
5112     str := iin.right
5113     if len(str) <= 0 {
5114         return
5115     }
5116     lastRune, lastSize := utf8.DecodeRuneInString(str[0:])
5117     if lastSize <= 0 {
5118         return
5119     }
5120     forw := false
5121     back := false
5122     pair := ""
5123     switch string(lastRune){
5124     case "{": pair = "}"; forw = true
5125     case "}": pair = "{"; back = true
5126     case "(": pair = ")"; forw = true
5127     case ")": pair = "("; back = true
5128     case "[": pair = "]"; forw = true
5129     case "]": pair = "["; back = true
5130     case "<": pair = ">"; forw = true
5131     case ">": pair = "<"; back = true
5132     case "\\": pair = "\\"; // context depednet, can be f" or back-double quote
5133     case "'": pair = "'"; // context depednet, can be f' or back-quote
5134     // case Japanese Kakkos
5135     }
5136     if forw {
5137         iin.SearchForward(pair)
5138     }
5139     if back {
5140         iin.SearchBackward(pair)
5141     }
5142 }
5143 // 0.2.8 2020-0902 created
5144 func (iin*IInput)SearchForward(pat string)(bool){
5145     right := iin.right
5146     found := -1
5147     i := 0
5148     if strBegins(right,pat) {
5149         _,z := utf8.DecodeRuneInString(right[i:])
5150         if 0 < z {
5151             i += z
5152         }
5153     }
5154     for i < len(right) {
5155         if strBegins(right[i:],pat) {
5156             found = i
5157             break
5158         }
5159         _,z := utf8.DecodeRuneInString(right[i:])
5160         if z <= 0 { break }
5161         i += z
5162     }
5163     if 0 <= found {
5164         iin.line = iin.line + right[0:found]
5165         iin.right = iin.right[found:]
5166         return true
5167     }else{
5168         return false
5169     }
5170 }
5171 // 0.2.8 2020-0902 created
5172 func (iin*IInput)SearchBackward(pat string)(bool){
5173     line := iin.line
5174     found := -1
5175     i := len(line)-1
5176     for i = i; 0 <= i; i-- {
5177         _,z := utf8.DecodeRuneInString(line[i:])
5178         if z <= 0 {
5179             continue
5180         }
5181         //fprintf(stderr,"-- %v %v\n",pat,line[i:])
5182         if strBegins(line[i:],pat) {
5183             found = i
5184             break
5185         }
5186     }
5187     //fprintf(stderr,"--%d\n",found)
5188     if 0 <= found {
5189         iin.right = line[found:] + iin.right
5190         iin.line = line[0:found]
5191         return true
5192     }else{
5193         return false
5194     }
5195 }
5196 // 0.2.8 2020-0902 created
5197 // search from top, end, or current position
5198 func (gsh*GshContext)SearchHistory(pat string, forw bool)(bool,string){
5199     if forw {
5200         for _,v := range gsh.CommandHistory {
5201             if 0 <= strings.Index(v.CmdLine,pat) {
5202                 //fprintf(stderr,"\n--De-- found !%v [%v]%v\n",i,pat,v.CmdLine)
5203                 return true,v.CmdLine
5204             }
5205         }
5206     }else{
5207         hlen := len(gsh.CommandHistory)
5208         for i := hlen-1; 0 < i; i-- {

```

```

5209     v := gsh.CommandHistory[i]
5210     if 0 < strings.Index(v.CmdLine,pat) {
5211         //fprintf(stderr,"\n--De-- found !%v [%v]%v\n",i,pat,v.CmdLine)
5212         return true,v.CmdLine
5213     }
5214 }
5215 }
5216 //fprintf(stderr,"\n--De-- not-found(%v)\n",pat)
5217 return false,"(Not Found in History)"
5218 }
5219 // 0.2.8 2020-0902 created
5220 func (iin*IInput)GotoFORWSTR(pat string,gsh*GshContext){
5221     found := false
5222     if 0 < len(iin.right) {
5223         found = iin.SearchForward(pat)
5224     }
5225     if !found {
5226         found,line := gsh.SearchHistory(pat,true)
5227         if found {
5228             iin.line = line
5229             iin.right = ""
5230         }
5231     }
5232 }
5233 func (iin*IInput)GotoBACKSTR(pat string, gsh*GshContext){
5234     found := false
5235     if 0 < len(iin.line) {
5236         found = iin.SearchBackward(pat)
5237     }
5238     if !found {
5239         found,line := gsh.SearchHistory(pat,false)
5240         if found {
5241             iin.line = line
5242             iin.right = ""
5243         }
5244     }
5245 }
5246 func (iin*IInput)getString1(prompt string)(string){ // should be editable
5247     iin.clearline();
5248     fprintf(stderr,"\r%v",prompt)
5249     str := ""
5250     for {
5251         ch := iin.Getc(10*1000*1000)
5252         if ch == '\n' || ch == '\r' {
5253             break
5254         }
5255         sch := string(ch)
5256         str += sch
5257         fprintf(stderr,"%s",sch)
5258     }
5259     return str
5260 }
5261 // search pattern must be an array and selectable with 'N'/P
5262 var SearchPat = ""
5263 var SearchForw = true
5264 var SearchForw = true
5265 }
5266 func (iin*IInput)xgetline(prevline string, gsh*GshContext)(string){
5267     var ch int;
5268 }
5269 MODE_ShowMode = false
5270 MODE_VicMode = false
5271 iin.Redraw();
5272 first := true
5273 }
5274 for cix := 0; ; cix++ {
5275     iin.pinJmode = iin.inJmode
5276     iin.inJmode = false
5277 }
5278 ch = iin.Getc(1000*1000)
5279 if ch != EV_TIMEOUT && first {
5280     first = false
5281     mode := 0
5282     if romkanmode {
5283         mode = 1
5284     }
5285 }
5286 now := time.Now()
5287 Events = append(Events,Event{now,EV_MODE,int64(mode),CmdIndex})
5288 }
5289 if ch == 033 {
5290     MODE_ShowMode = true
5291     MODE_VicMode = !MODE_VicMode
5292     iin.Redraw();
5293     continue
5294 }
5295 if MODE_VicMode {
5296     switch ch {
5297     case '0': ch = GO_TOPL
5298     case '$': ch = GO_ENDL
5299     case 'b': ch = GO_TOPW
5300     case 'e': ch = GO_ENDW
5301     case 'w': ch = GO_NEXTW
5302     case '%': ch = GO_PAIRCH
5303 }
5304 case 'j': ch = GO_DOWN
5305 case 'k': ch = GO_UP
5306 case 'h': ch = GO_LEFT
5307 case 'l': ch = GO_RIGHT
5308 case 'x': ch = DEL_RIGHT
5309 case 'a': MODE_VicMode = !MODE_VicMode
5310     ch = GO_RIGHT
5311 case 'i': MODE_VicMode = !MODE_VicMode
5312     iin.Redraw();
5313     continue
5314 case '-':
5315     right,head := delHeadChar(iin.right)
5316     if len([]byte(head)) == 1 {
5317         ch = int(head[0])
5318         if( 'a' <= ch && ch <= 'z' ){
5319             ch = ch + 'A'-'a'
5320         }else
5321         if( 'A' <= ch && ch <= 'Z' ){
5322             ch = ch + 'a'-'A'
5323         }
5324         iin.right = string(ch) + right
5325     }
5326     iin.Redraw();
5327     continue
5328 case 'f': // GO_FORWCH
5329     iin.Redraw();
5330     ch = iin.Getc(3*1000*1000)
5331     if ch == EV_TIMEOUT {
5332         iin.Redraw();

```

```

5333         continue
5334     }
5335     SearchPat = string(ch)
5336     SearchForw = true
5337     iin.GotoFORWSTR(SearchPat,gsh)
5338     iin.Redraw();
5339     continue
5340 case '/':
5341     SearchPat = iin.getstringl("/") // should be editable
5342     SearchForw = true
5343     iin.GotoFORWSTR(SearchPat,gsh)
5344     iin.Redraw();
5345     continue
5346 case '?':
5347     SearchPat = iin.getstringl("??") // should be editable
5348     SearchForw = false
5349     iin.GotoBACKSTR(SearchPat,gsh)
5350     iin.Redraw();
5351     continue
5352 case 'n':
5353     if SearchForw {
5354         iin.GotoFORWSTR(SearchPat,gsh)
5355     }else{
5356         iin.GotoBACKSTR(SearchPat,gsh)
5357     }
5358     iin.Redraw();
5359     continue
5360 case 'N':
5361     if !SearchForw {
5362         iin.GotoFORWSTR(SearchPat,gsh)
5363     }else{
5364         iin.GotoBACKSTR(SearchPat,gsh)
5365     }
5366     iin.Redraw();
5367     continue
5368     }
5369 }
5370 switch ch {
5371 case GO_TOPW:
5372     iin.GotoTOPW()
5373     iin.Redraw();
5374     continue
5375 case GO_ENDW:
5376     iin.GotoENDW()
5377     iin.Redraw();
5378     continue
5379 case GO_NEXTW:
5380     // to next space then
5381     iin.GotoNEXTW()
5382     iin.Redraw();
5383     continue
5384 case GO_PAIRCH:
5385     iin.GotoPAIRCH()
5386     iin.Redraw();
5387     continue
5388 }
5389
5390 //fprintf(stderr,"A[%02X]\n",ch);
5391 if( ch == '\\' || ch == 033 ){
5392     MODE_ShowMode = true
5393     metach := ch
5394     iin.waitingMeta = string(ch)
5395     iin.Redraw();
5396     // set cursor //fprintf(stderr,"???\b\b\b")
5397     ch = fgetcTimeout(stdin,2000*1000)
5398     // reset cursor
5399     iin.waitingMeta = ""
5400
5401     cmdch := ch
5402     if( ch == EV_TIMEOUT ){
5403         if metach == 033 {
5404             continue
5405         }
5406         ch = metach
5407     }else
5408     /*
5409     if( ch == 'm' || ch == 'M' ){
5410         mch := fgetcTimeout(stdin,1000*1000)
5411         if mch == 'r' {
5412             romkanmode = true
5413         }else{
5414             romkanmode = false
5415         }
5416         continue
5417     }else
5418     /*
5419     if( ch == 'k' || ch == 'K' ){
5420         MODE_Recursive = !MODE_Recursive
5421         iin.Translate(cmdch);
5422         continue
5423     }else
5424     if( ch == 'j' || ch == 'J' ){
5425         iin.Translate(cmdch);
5426         continue
5427     }else
5428     if( ch == 'i' || ch == 'I' ){
5429         iin.Replace(cmdch);
5430         continue
5431     }else
5432     if( ch == 'l' || ch == 'L' ){
5433         MODE_LowerLock = !MODE_LowerLock
5434         MODE_CapsLock = false
5435         if MODE_ViTrace {
5436             fprintf(stderr,"%v\r\n",string(cmdch));
5437         }
5438         iin.Redraw();
5439         continue
5440     }else
5441     if( ch == 'u' || ch == 'U' ){
5442         MODE_CapsLock = !MODE_CapsLock
5443         MODE_LowerLock = false
5444         if MODE_ViTrace {
5445             fprintf(stderr,"%v\r\n",string(cmdch));
5446         }
5447         iin.Redraw();
5448         continue
5449     }else
5450     if( ch == 'v' || ch == 'V' ){
5451         MODE_ViTrace = !MODE_ViTrace
5452         if MODE_ViTrace {
5453             fprintf(stderr,"%v\r\n",string(cmdch));
5454         }
5455         iin.Redraw();
5456         continue

```

```

5457     }else
5458     if( ch == 'c' || ch == 'C' ){
5459         if 0 < len(iin.line) {
5460             xline,tail := delTailChar(iin.line)
5461             if len([]byte(tail)) == 1 {
5462                 ch = int(tail[0])
5463                 if( 'a' <= ch && ch <= 'z' ){
5464                     ch = ch + 'A'-'a'
5465                 }else
5466                 if( 'A' <= ch && ch <= 'Z' ){
5467                     ch = ch + 'a'-'A'
5468                 }
5469                 iin.line = xline + string(ch)
5470             }
5471         }
5472         if MODE_ViTrace {
5473             fprintf(stderr,"%v\r\n",string(cmdch));
5474         }
5475         iin.Redraw();
5476         continue
5477     }else{
5478         iin.pch = append(iin.pch,ch) // push
5479         ch = '\\'
5480     }
5481 }
5482 switch( ch ){
5483 case 'P'-0x40: ch = GO_UP
5484 case 'N'-0x40: ch = GO_DOWN
5485 case 'B'-0x40: ch = GO_LEFT
5486 case 'F'-0x40: ch = GO_RIGHT
5487 }
5488 //fprintf(stderr,"B[%02X]\n",ch);
5489 switch( ch ){
5490 case 0:
5491     continue;
5492
5493 case '\t':
5494     iin.Replace('j');
5495     continue
5496 case 'X'-0x40:
5497     iin.Replace('j');
5498     continue
5499
5500 case EV_TIMEOUT:
5501     iin.Redraw();
5502     if iin.pinJmode {
5503         fprintf(stderr,"\\J\r\n")
5504         iin.inJmode = true
5505     }
5506     continue
5507 case GO_UP:
5508     if iin.lno == 1 {
5509         continue
5510     }
5511     cmd,ok := gsh.cmdStringInHistory(iin.lno-1)
5512     if ok {
5513         iin.line = cmd
5514         iin.right = ""
5515         iin.lno = iin.lno - 1
5516     }
5517     iin.Redraw();
5518     continue
5519 case GO_DOWN:
5520     cmd,ok := gsh.cmdStringInHistory(iin.lno+1)
5521     if ok {
5522         iin.line = cmd
5523         iin.right = ""
5524         iin.lno = iin.lno + 1
5525     }else{
5526         iin.line = ""
5527         iin.right = ""
5528         if iin.lno == iin.lastlno-1 {
5529             iin.lno = iin.lno + 1
5530         }
5531     }
5532     iin.Redraw();
5533     continue
5534 case GO_LEFT:
5535     if 0 < len(iin.line) {
5536         xline,tail := delTailChar(iin.line)
5537         iin.line = xline
5538         iin.right = tail + iin.right
5539     }
5540     iin.Redraw();
5541     continue;
5542 case GO_RIGHT:
5543     if( 0 < len(iin.right) && iin.right[0] != 0 ){
5544         xright,head := delHeadChar(iin.right)
5545         iin.right = xright
5546         iin.line += head
5547     }
5548     iin.Redraw();
5549     continue;
5550 case EOF:
5551     goto EXIT;
5552 case 'R'-0x40: // replace
5553     dst := convs(iin.line+iin.right);
5554     iin.line = dst
5555     iin.right = ""
5556     iin.Redraw();
5557     continue;
5558 case 'T'-0x40: // just show the result
5559     readDic();
5560     romkanmode = !romkanmode;
5561     iin.Redraw();
5562     continue;
5563 case 'L'-0x40:
5564     iin.Redraw();
5565     continue
5566 case 'K'-0x40:
5567     iin.right = ""
5568     iin.Redraw();
5569     continue
5570 case 'E'-0x40:
5571     iin.line += iin.right
5572     iin.right = ""
5573     iin.Redraw();
5574     continue
5575 case 'A'-0x40:
5576     iin.right = iin.line + iin.right
5577     iin.line = ""
5578     iin.Redraw();
5579     continue
5580 case 'U'-0x40:

```

```

5581         iin.line = ""
5582         iin.right = ""
5583         iin.clearline();
5584         iin.Redraw();
5585         continue;
5586     case DEL_RIGHT:
5587         if( 0 < len(iin.right) ){
5588             iin.right,_ = delHeadChar(iin.right)
5589             iin.Redraw();
5590         }
5591         continue;
5592     case 0x7F: // BS? not DEL
5593         if( 0 < len(iin.line) ){
5594             iin.line,_ = delTailChar(iin.line)
5595             iin.Redraw();
5596         }
5597         /*
5598         else
5599             if( 0 < len(iin.right) ){
5600                 iin.right,_ = delHeadChar(iin.right)
5601                 iin.Redraw();
5602             }
5603         */
5604         continue;
5605     case 'H'-0x40:
5606         if( 0 < len(iin.line) ){
5607             iin.line,_ = delTailChar(iin.line)
5608             iin.Redraw();
5609         }
5610         continue;
5611     }
5612     if( ch == '\n' || ch == '\r' ){
5613         iin.line += iin.right;
5614         iin.right = ""
5615         iin.Redraw();
5616         fputc(ch,stderr);
5617         AtConsoleLineTop = true
5618         break;
5619     }
5620     if MODE_CapsLock {
5621         if 'a' <= ch && ch <= 'z' {
5622             ch = ch+'A'-'a'
5623         }
5624     }
5625     if MODE_LowerLock {
5626         if 'A' <= ch && ch <= 'Z' {
5627             ch = ch+'a'-'A'
5628         }
5629     }
5630     iin.line += string(ch);
5631     iin.Redraw();
5632 }
5633 EXIT:
5634     return iin.line + iin.right;
5635 }
5636
5637 func getline_main(){
5638     line := Xgetline(0,"",nil)
5639     fprintf(stderr,"%s\n",line);
5640 /*
5641     dp = strchr(line,"\r\n");
5642     if( dp != NULL ){
5643         *dp = 0;
5644     }
5645
5646     if( 0 ){
5647         fprintf(stderr,"\n(%d)\n",int(strlen(line)));
5648     }
5649     if( lseek(3,0,0) == 0 ){
5650         if( romkanmode ){
5651             var buf [8*1024]byte;
5652             convs(line,buf);
5653             strcpy(line,buf);
5654         }
5655         write(3,line,strlen(line));
5656         ftruncate(3,lseek(3,0,SEEK_CUR));
5657         //fprintf(stderr,"outsize=%d\n",(int)lseek(3,0,SEEK_END));
5658         lseek(3,0,SEEK_SET);
5659         close(3);
5660     }else{
5661         fprintf(stderr,"\r\ngetline: ");
5662         trans(line);
5663         //printf("%s\n",line);
5664         printf("\n");
5665     }
5666 */
5667 }
5668 //== end ===== getline
5669 //
5670 // $USERHOME/.gsh/
5671 // gsh-rc.txt, or gsh-configure.txt
5672 // gsh-history.txt
5673 // gsh-aliases.txt // should be conditional?
5674 //
5675 func (gshCtx *GshContext)gshSetupHomedir()(bool) {
5676     homedir,found := userHomeDir()
5677     if !found {
5678         fmt.Printf("--E-- You have no UserHomeDir\n")
5679         return true
5680     }
5681     gshhome := homedir + "/" + GSH_HOME
5682     _, err2 := os.Stat(gshhome)
5683     if err2 != nil {
5684         err3 := os.Mkdir(gshhome,0700)
5685         if err3 != nil {
5686             fmt.Printf("--E-- Could not Create %s (%s)\n",
5687                 gshhome,err3)
5688             return true
5689         }
5690     }
5691     fmt.Printf("--I-- Created %s\n",gshhome)
5692 }
5693 gshCtx.GshHomeDir = gshhome
5694 return false
5695 }
5696 func setupGshContext()(GshContext,bool){
5697     gshPA := syscall.ProcAttr {
5698         "", // the staring directory
5699         os.Environ(), // environ[]
5700         []uintptr{os.Stdin.Fd(),os.Stdout.Fd(),os.Stderr.Fd()},
5701         nil, // OS specific
5702     }
5703     cwd,_ := os.Getwd()
5704     gshCtx := GshContext {

```

```

5705     cwd, // StartDir
5706     "", // GetLine
5707     []GChdirHistory { {cwd,time.Now(),0} }, // ChdirHistory
5708     gshP,
5709     []GCommandHistory{}, //something for invokation?
5710     GCommandHistory{}, // CmdCurrent
5711     false,
5712     []int{},
5713     syscall.Rusage{},
5714     "", // GshHomeDir
5715     Ttyid(),
5716     false,
5717     false,
5718     []PluginInfo{},
5719     []string{},
5720     "",
5721     "v",
5722     ValueStack{},
5723     GServer{"", ""}, // LastServer
5724     // RSERV
5725     cwd, // RWD
5726     CheckSum(),
5727 }
5728 err := gshCtx.gshSetupHomedir()
5729 return gshCtx, err
5730 }
5731 func (gsh*GshContext)gshelllh(gline string)(bool){
5732     ghist := gsh.CmdCurrent
5733     ghist.WorkDir,_ = os.Getwd()
5734     ghist.WorkDirX = len(gsh.ChdirHistory)-1
5735     //fmt.Printf("--D--ChdirHistory(%#d)\n",len(gsh.ChdirHistory))
5736     ghist.StartAt = time.Now()
5737     rusagev1 := Getrusagev()
5738     gsh.CmdCurrent.FoundFile = []string{}
5739     fin := gsh.tgshellh(gline)
5740     rusagev2 := Getrusagev()
5741     ghist.Rusagev = RusageSubv(rusagev2,rusagev1)
5742     ghist.EndAt = time.Now()
5743     ghist.CmdLine = gline
5744     ghist.FoundFile = gsh.CmdCurrent.FoundFile
5745
5746     /* record it but not show in list by default
5747     if len(gline) == 0 {
5748         continue
5749     }
5750     if gline == "hi" || gline == "history" { // don't record it
5751         continue
5752     }
5753     */
5754     gsh.CommandHistory = append(gsh.CommandHistory, ghist)
5755     return fin
5756 }
5757 // <a name="main">Main loop</a>
5758 func script(gshCtxGiven *GshContext) (_ GshContext) {
5759     gshCtxBuf,err0 := setupGshContext()
5760     if err0 {
5761         return gshCtxBuf;
5762     }
5763     gshCtx := *gshCtxBuf
5764
5765     //fmt.Printf("--I-- GSH_HOME=%s\n",gshCtx.GshHomeDir)
5766     //resmap()
5767
5768     /*
5769     if false {
5770         gsh_getlinev, with_exgetline :=
5771             which("PATH",[]string{"which","gsh-getline","-s"})
5772         if with_exgetline {
5773             gsh_getlinev[0] = toFullpath(gsh_getlinev[0])
5774             gshCtx.GetLine = toFullpath(gsh_getlinev[0])
5775         }else{
5776             fmt.Printf("--W-- No gsh-getline found. Using internal getline.\n");
5777         }
5778     }
5779     */
5780
5781     ghist0 := gshCtx.CmdCurrent // something special, or gshrc script, or permanent history
5782     gshCtx.CommandHistory = append(gshCtx.CommandHistory,ghist0)
5783
5784     prevline := ""
5785     skipping := false
5786     for hix := len(gshCtx.CommandHistory); ; {
5787         gline := gshCtx.getline(hix,skipping,prevline)
5788         if skipping {
5789             if strings.Index(gline,"fi") == 0 {
5790                 fmt.Printf("fi\n");
5791                 skipping = false;
5792             }else{
5793                 //fmt.Printf("%s\n",gline);
5794             }
5795             continue
5796         }
5797         if strings.Index(gline,"if") == 0 {
5798             //fmt.Printf("--D-- if start: %s\n",gline);
5799             skipping = true;
5800             continue
5801         }
5802         if false {
5803             os.Stdout.Write([]byte("gotline:"))
5804             os.Stdout.Write([]byte(gline))
5805             os.Stdout.Write([]byte("\n"))
5806         }
5807         gline = strsubst(gshCtx,gline,true)
5808         if false {
5809             fmt.Printf("fmt.Printf %v - %v\n",gline)
5810             fmt.Printf("fmt.Printf %s - %s\n",gline)
5811             fmt.Printf("fmt.Printf %x - %s\n",gline)
5812             fmt.Printf("fmt.Printf %U - %s\n",gline)
5813             fmt.Printf("Stoutt.Write -")
5814             os.Stdout.Write([]byte(gline))
5815             fmt.Printf("\n")
5816         }
5817         /*
5818         // should be cared in substitution ?
5819         if 0 < len(gline) && gline[0] == '!' {
5820             xgline, set, err := searchHistory(gshCtx,gline)
5821             if err {
5822                 continue
5823             }
5824             if set {
5825                 // set the line in command line editor
5826             }
5827             gline = xgline
5828         }

```



```

5953 "CeWGHQpzdW1vbW8J44GZ44KC44KCCnN1bW9tb21vCeOBmeOCguOCguOCggptb21vCeahgwpt"+
5954 "b21vbW8J5gD44KCCiwsCeOAgQouLgnjg11KFC90ZxH0YXJ1YT4K"
5955 //</span>
5956
5957 var SijimiDic = //<span id="gsh-sijimi-dic">
5958 "data:text/dic;base64,"+
5959 "Pgl1dEGy2hcnNldD0iVVRGLTgiPgo8dGV4dGFyZWV29scz04M0B3b3dzPTQwPgovL3Zl"+
5960 "cglHU2h1bGxoc01NRVxzZjJlZGl1bGVmFyeVxzZm9yXHNtTaGlqWlpxHmVlL1xzMjYyMC0wODMw"+
5961 "CnNpCe0BlWpzaGkx44GXCmpCpCe0BmApAptAqNjg8KbMzE44GqCmp1Ce0Bm00ChOp4eXUJ44KF"+
5962 "CuU44G0Cm55Ce0BqWprbwnjg2MKYnUJ44G2Cm55Ce0Ckwpubwnjga4KY2hpCe0Bo0p0aQnJ"+
5963 "gaEka2EJ44GCLnJhCe0CiQosLAnjgIeKLi4J44CCnHuYw5hCeS4gwp4anV1CeWNgOp4bmkJ"+
5964 "5LqCmteVAnlGIsKa29xCeWaiwprb3gu5YCLCm55hmfqdxVuaXgJNzIKbmFuYwP1dW5pHgJ"+
5965 "77yX77yScm55hmfqdxVuaVgJ77yX77yScuS4g+WNgeS6jHgJNzIKa29idW55CeWai+WIhgp0"+
5966 "awthcmFxCe0Boe0Bi+OCiQp0awthcmEJ5YqbcMNoawthcmEJ5YqbcjvvdGV4dGFyZWV2eWpCg="
5967 //</span>
5968
5969 var JA_JKLDic = //<span id="gsh-ja-jkl-dic">
5970 "data:text/dic;base64,"+
5971 "Ly92Z2Jsc015SUIFamRyY2ptb3ZzZwPKQWpKS0w0mJyAYMGowODE5KShelV4pL1NhdG94SVRT"+
5972 "CmtqamprbGta2tsa2ps10S4lueVJApgamtqamwJ44G0CmtqbAnjgYQKa2tqbAnjgYKamtq"+
5973 "amwJ44G0Cmtq2trbAnjgYKa2pra2wJ44G0CmpzamttrbAnjgYQKa2trawJ44G0Cmpzamps"+
5974 "Ce0B0pamprbAnjgZKamtq2psCe0Bl0pamtq2wJ44G0CmpqamtqbAnjgZKka2pqamts"+
5975 "Ce0BmwpamprbAnjgZ0KamtscE0Bnwpr2prbAnjgaEka2pqa2wJ44G0Cmtqa2pqbAnjgYK"+
5976 "a2tqa2tsCe0BqApramtsCe0Bqgppa2prbAnjgKa2tra2wJ44G0Cmpga2psCe0B0p2pqa2p"+
5977 "bAnjga4Kamt2wJ44G0Cmpqa2tqbAnjgIKampra2wJ44G1CmtsCe0BuApqa2tsCe0Buwppq"+
5978 "a2tqbAnjg4Ka2tqa2psCe0BwvppqAnjgaKamtra2psCe0CgOpqa2tqa2wJ44K0CmtqamwJ"+
5979 "44K0Cmp2pqbAnjgYKampsCe0CiApra2tsCe0Ci0pamtsCe0Ci0pqa2pqa2wJ44K0Cmpq"+
5980 "amwJ44K0Cmtqa2psCe0Cj0pqa2psCe0CjwpramtramwJ44K0CmtqamtrbAnjgYKa2pqa2p"+
5981 "44K0Cmtqa2prbAnjgYKa2pqa2psCe0DvApra2wJ44K0CmtramprbAnjgYKa2pamtqbAnj"+
5982 "gIEK";
5983 //</span>
5984
5985 //</span>
5986 /*
5987 <style id="gsh-references-style">
5988 #references details { font-family:Georgia; }
5989 #references a { font-family:Georgia; }
5990 .wrap { white-space:normal; }
5991 </style>
5992 <details id="references"><summary>References</summary><div class="gsh-src">
5993 Web technology
5994 <a href="https://html.spec.whatwg.org">HTML: The Living Standard</a> (September 2020)
5995 <a href="https://html.spec.whatwg.org/dev/">Developer Version</a>
5996
5997 <a href="https://drafts.csswg.org">CSS Working Group Editor Drafts</a> (September 2020)
5998
5999 <a href="https://developer.mozilla.org/ja/docs/Web">MDN web docs</a>
6000 <a href="https://developer.mozilla.org/ja/docs/Web/HTML/Element">HTML</a>
6001 <a href="https://developer.mozilla.org/en-US/docs/Web/CSS">CSS</a> : <span class="wrap">
6002 <a href="https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors">selectors</a>
6003 <a href="https://developer.mozilla.org/en-US/docs/Web/CSS/background-repeat">repeat</a></span>
6004 <a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript">JavaScript</a>
6005 <a href="https://developer.mozilla.org/en-US/docs/Web/HTTP">HTTP</a>
6006 <a href="https://mdn-web-dna.s3-us-west-2.amazonaws.com/MDN-Browser-Compatibility-Report-2020.pdf">MDN Browser Compatibility Report (2020)</a>
6007
6008 Go language (August 2020 / Go 1.15)
6009 <a href="https://golang.org">The Go Programming Language</a>
6010 <a href="https://golang.org/pkg/">Packages</a>
6011 <a href="https://godoc.org/golang.org/x/net/websocket">WebSocket</a>
6012
6013 <a href="https://stackoverflow.com/">Stackoverflow</a>
6014 <!--
6015 <iframe src="https://golang.org" width="100%" height="300"></iframe>
6016 -->
6017 </div></details>
6018 */
6019 /*
6020 <details id="html-src" onclick="frame_open();"><summary>Raw Source</summary><div>
6021
6022 <!-- h2>The full of this HTML including the Go code is here.</h2 -->
6023 <details id="gsh-whole-view"><summary>Whole file</summary>
6024 <a name="whole-src-view"></a>
6025 <span id="src-frame"></span><!-- a window to show source code -->
6026 </details>
6027
6028 <details id="gsh-style-frame" onclick="fill_CSSView()"><summary>CSS part</summary>
6029 <a name="style-src-view"></a>
6030 <span id="gsh-style-view"></span>
6031 </details>
6032
6033 <details id="gsh-script-frame" onclick="fill_JavaScriptView()"><summary>JavaScript part</summary>
6034 <a name="script-src-view"></a>
6035 <span id="gsh-script-view"></span>
6036 </details>
6037
6038 <details id="gsh-data-frame" onclick="fill_DataView()"><summary>Builtin data part</summary>
6039 <a name="gsh-data-frame"></a>
6040 <span id="gsh-data-view"></span>
6041 </details>
6042
6043 </div></details>
6044 */
6045
6046 /*
6047 <div id="GshFooter0"></div>
6048 <!-- 2020-09-17 SatoxITS, visible script { -- >
6049 <details><summary>GJScript</summary>
6050 <style>.gjscript { font-family:Georgia; }</style>
6051 <pre id="gjscript_1" class="gjscript"> function gjttest1(){ alert('Hello GJScript!'); }
6052 gjttest1()
6053 </pre>
6054 <script>
6055 gjs = document.getElementById('gjscript_1');
6056 //eval(gjs.innerHTML);
6057 //gjs.outerHTML = ""
6058 </script>
6059 </details><!-- ----- END-OF-VISIBLE-PART ----- } -->
6060
6061 <!--
6062 // 2020-0906 added,
6063 https://developer.mozilla.org/en-US/docs/Web/CSS/z-index
6064 https://developer.mozilla.org/en-US/docs/Web/CSS/position
6065 -->
6066 <span id="GshGrid">(^_^)</small><small>[Hit j k l h]</small></span>
6067
6068 <span id="GStat"><br>
6069 </span>
6070 <span id="GMenu" onclick="GShellMenu(this)" draggable="true"></span>
6071 <span id="GTop"></span>
6072 <div id="GShellPlane" onclick="showGShellPlane();"></div>
6073 <div id="RawTextViewer"></div>
6074 <div id="RawTextViewerClose" onclick="hideRawTextViewer()"> CLOSE </div>
6075
6076 <style id="GshStyleDef">

```

```

6077 #LineNumbered table,tr,td {
6078     margin:0;
6079     padding:4px;
6080     spacing:0;
6081     border:12px;
6082 }
6083 textarea.LineNumber {
6084     font-size:12px;
6085     font-family:monospace,Courier New;
6086     color:#282;
6087     padding:4px;
6088     text-align:right;
6089 }
6090 textarea.LineNumbered {
6091     font-size:12px;
6092     font-family:monospace,Courier New;
6093     padding:4px;
6094     wrap:off;
6095 }
6096 #RawTextViewer{
6097     z-index:0;
6098     position:fixed; top:0px; left:0px;
6099     width:100%; xxxheight:50px; xheight:0px;
6100     overflow:auto;
6101     color:#fff; background-color:rgba(128,128,256,0.2);
6102     font-size:12px;
6103     spellcheck:false;
6104 }
6105 #RawTextViewerClose{
6106     z-index:0;
6107     position:fixed; top:-100px; left:-100px;
6108     color:#fff; background-color:rgba(128,128,256,0.2);
6109     font-size:20px; font-family:Georgia;
6110     white-space:pre;
6111 }
6112 #xxxGShellPlane{
6113     z-index:0;
6114     position:fixed; top:0px; left:0px;
6115     width:100%; height:50px;
6116     overflow:auto;
6117     color:#fff; background-color:rgba(128,128,256,0.3);
6118     font-size:12px;
6119 }
6120 #xxxGTop{
6121     z-index:9;
6122     opacity:1.0;
6123     position:fixed; top:0px; left:0px;
6124     width:320px; height:20px;
6125     color:#fff; background-color:rgba(32,32,160,0.15);
6126     color:#fff; font-size:12px;
6127 }
6128 #xxxGPos{
6129     z-index:12;
6130     position:fixed; top:0px; left:0px;
6131     opacity:1.0;
6132     width:640px; height:30px;
6133     color:#fff; background-color:rgba(0,0,0,0.2);
6134     color:#fff; font-size:12px;
6135 }
6136 #GMenu{
6137     z-index:100000000;
6138     position:fixed; top:250px; left:0px;
6139     opacity:1.0;
6140     width:100px; height:100px;
6141     color:#fff;
6142     color:#fff; background-color:rgba(0,0,0,0.0);
6143     color:#fff; font-size:16px; font-family:Georgia;
6144     background-repeat:no-repeat;
6145 }
6146 #xxxGStat{
6147     z-index:8;
6148     xopacity:0.0;
6149     position:fixed; top:20px; left:0px;
6150     xwidth:640px;
6151     width:100%; height:90px;
6152     color:#fff; background-color:rgba(0,0,128,0.04);
6153     font-size:20px; font-family:Georgia;
6154 }
6155 #GLog{
6156     z-index:10;
6157     position:fixed; top:50px; left:0px;
6158     opacity:1.0;
6159     width:640px; height:60px;
6160     color:#fff; background-color:rgba(0,0,128,0.10);
6161     font-size:12px;
6162 }
6163 #GshGrid {
6164     z-index:11;
6165     xopacity:0.0;
6166     position:fixed; top:0px; left:0px;
6167     width:320px; height:30px;
6168     color:#9f9; font-size:16px;
6169 }
6170 xbody {display:none;}
6171 .gsh-link{color:green;}
6172 #gsh {border-width:1;margin:0;padding:0;}
6173 #gsh {font-family:monospace,Courier New;color:#ddf;font-size:8px;}
6174 #gsh header{height:100px;}
6175 #xgsh header{height:100px;background-image:url(GShell-Logo00.png);}
6176 #GshMenu{font-size:14pt;color:#c44;}
6177 .GshMenu{
6178     font-size:14pt;color:#2a2;padding:4px; text-align:right;
6179 }
6180 .GshMenu: hover{
6181     font-size:14pt;color:#fff;font-weight:bold;background-color:#2a2;
6182 }
6183 #GshFooter{height:100px;background-size:80px;background-repeat:no-repeat;}
6184 #gsh note{color:#000;font-size:10pt;}
6185 #gsh h2{color:#24a;font-family:Georgia;font-size:18pt;}
6186 #gsh h3{color:#24a;font-family:Georgia;font-size:16pt;}
6187 #gsh details{color:#888;background-color:#fff;font-family:monospace;}
6188 #gsh summary{font-size:16pt;color:#fff;background-color:#8af;height:30px;}
6189 #gsh summary{font-size:16pt;color:#fff;
6190     xxx-background-color:#8af;
6191     background-color:#6881AD;xxx-PBlue;
6192     height:30px;}
6193 }
6194 #gsh pre{font-size:11pt;color:#223;background-color:#fafff;}
6195 #gsh a{color:#24a;}
6196 #gsh a[name]{color:#24a;font-size:16pt;}
6197 #gsh .gsh-src{white-space:pre;font-family:monospace,Courier New;font-size:11pt;}
6198 #gsh .gsh-src{background-color:#Eafff;color:#223;}
6199 #gsh-src-src{spellcheck:false}
6200 #SrcTextarea{white-space:pre;font-family:Courier New;font-size:10pt;}

```

```

6201 #SrcTextarea{background-color:#fafff;color:#223;}
6202 .gsh-code {white-space:pre;font-family:Courier New !important;}
6203 .gsh-code {color:#024;font-size:11pt; background-color:#fafff;}
6204 .gsh-golang-data {display:none;}
6205 #gsh-winId {color:#000;font-size:14pt;}
6206
6207 .gsh-document {font-size:11pt;background-color:#fff;font-family:Georgia;}
6208 .gsh-document {color:#000;background-color:#fff !important;}
6209 .gsh-document > h2{color:#000;background-color:#fff !important;}
6210 .gsh-document details{color:#000;background-color:#fff;font-family:Georgia;}
6211 .gsh-document p{max-width:550pt;color:#000;background-color:#fff;font-family:Georgia;}
6212 .gsh-document address{width:500pt;color:#000;background-color:#fff;font-family:Georgia;}
6213
6214 @media print {
6215 #gsh pre{font-size:11pt !important;}
6216 }
6217 </style>
6218
6219 <!--
6220 // Logo image should be drawn by JavaScript from a meta-font.
6221 // CSS seems not follow line-splittd URL
6222 -->
6223 <script id="gsh-data">
6224 //GSellLogo="QR-ITS-more.jp.png"
6225 GsellLogo="data:image/png;base64,\
6226 iVBORw0KGgoAAAANSUHEUgAAQEAAB/CAYAAADvs3f4AAAAAXNSR0IArs4c6QAAAhLWElM\
6227 TU0AKGAAAABAAEAAUAAABAAAAPgEBAUAAUAAAABAAARgEoAAUAAAABAAIAADpAAQAAAB\
6228 AAAATGAAAAAAABIAAAAAQAAAEgAAAAAAOQADAAAAAQAABAACgAEAAAAAQAAAGgAAE\
6229 AAAAAQAAH8AAAAAY1BhgAAAAWfSL2AAALwAACXMBAAJcGAAAF3RJRFEUeAHTcQUFNWZ\
6230 x++t7ukZ3iCggO/jY60sb8WgZAvn7uG4+blSTR7YnQxdQCKGj2aWLD2MSLRkeUaPnoCdu\
6231 4iuJx7jrY25D0GmF2VqTBEiSggCoIMMA+mu+vu//2MD9U1da6a2aUbuV91GKrq3vvdv6/g\
6232 fnVxdx8tBA8SIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAES\
6233 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIA\
6234 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIA\
6235 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESI1FP14A8dLP2\
6236 zeXs9H9+ftSk5dHxs1c2qqdE7YusS+1qaa1KfnY5YsoKHWEPtdK4MQzP25UeExLbLYSAU15\
6237 mp0LlKXBClFLRM53J5Ug9SgqU06i+2kK3Stu0NY5rEeGKJ7Qw7m0vKec2Toq0LzwoLjhfS\
6238 jBOVICstMR3JUSUJ6Hfu7DsdmFb2+uX4vWwFFXBpMeZUIAE/hcK6gab66eK60llykh56PC\
6239 hX2UVBKRgk3h3gUeKlYdaof0N560Kd1f655wmcwOQlyPzi0NDdMLTPK/60p2P/P3xv\
6240 N8mFM+/NWJgnJw9KqOTolVGSPt2z2R1lgn3iJ0vK7YsoWwMzEuVfP1RKvdf0ak2LRSB0q\
6241 zrWocCOG6EhvgRacJ/dktj3q7dXh4gKN6AR50zpzergS6RAo2DQqE79SKTRKHu+/eFN\
6242 L66as88pU/PN1pNtLQJKSC73dPXSR20ur7iWpC8QbhnCynHUIlry0TYp5fjvqL7jx\
6243 +cNHjB5jgRyDLJHy39o84D0H2QtX8THaPeFuIOU+wLc+Knyh5FGEV0WGGAE883eXMoYU\
6244 iRkb9gHEP52VgQ14h89FU6KjYfYbBqbnzLJg4zFiesNDHWUoeIQV00b/5C9F9D1UueOH\
6245 +zghU9nSgOqgrm0Ugurk19RpjBD4Y6uQcQD5TU0W63zD3MHesy14V49isbdKyxgHlCpFR\
6246 UJ6toAOC7F9V58NBFDHT0MbaE74Ent+eWrrWr+Lz/Q7w6AD87BJUjps/OA7COoBNBCeMUZ\
6247 ttcU/coG28FLpvKELTPFV8juRasEahbHvXaR1guoeBpYfUD04+OfEBdyg8L4t29XeSXFAMOC\
6248 bgGovgov1zGgGw4JF392xnHhdC+MwF3J7Jfnt22yC1YJBYXNU55KIkyck1sXRld6BmcvN\
6249 aJovry/VbacmevqEP46/2lnj+jt9x17VL53215Mtvap1QLNHw5pDqQxyNTQ1Z2b8nCGMZ2V\
6250 q0oFJSYvV0AZzdfayidv6FJ35CS4jX2K9h1r7e27zm6p38ThLjpkYicJpV1HtK/DJFU4Jw1\
6251 1ImhXsIR9fzrzgRkx4w/C+HQSPe+krb1yR3qEPTNahsHaLds2xh5Q5Nc0PVDGepgcbm/8e\
6252 7/2doA0tPag/mlKJ77U0V0GxybTdx/Ex/PTfa/l7r7Ku+cSoiCxtWrohUxV16wEV9H+ccVg1\
6253 pd/CFU42AK2IUP1vTKL/sjJyE5PVHq728NzvFuzvDODGy9GooouhmMLNfctX48YHL2GH\
6254 f/8hpXVU/43rOg9xtg6fctv1XDC3fmdQn9nb21e7wE1b0K651cBu0Eghd3Taw82dwKpUw\
6255 hXaU66c6m6jE6n1Ji9/PTeH+LoMaOp+NI3JL548d91Pa\
6256 NNW4+P9y9jXuPEDL/HXnzagTsvaeLD2vsWHI9mu5rvvX9fSo4V/LfmgdEIPhDGLfh2cUg\
6257 gJ1y2wOEPaZ3fEicvld+2YNClNYrYhyhGa8JR0JTAImRiQcJrNw5FpT++r7wdh4SiUv\
6258 BLVWbfffLcRF04qazRD7176/rBjkyLD5pbiz25i4wQu7tikPBeC0puW+Kj0sp8GHNoAZuLw\
6259 iOzuYv0h9zBr2XDrXQVF15QxxH60wVjRKAAM46pvt+RxAUvLjw7vY9/+CeURMk168/rPQn\
6260 mCufKzaldFN/y18gA5iwc3dKIKshyVzuCYSVG/KhcwFWRDKAMcD8EKX+rHF12A9bt2d172\
6261 2qNzOvzCYdmfEtNy0oQDXWIKAIQ7c0Q2chyADWnerN5xVtXtJsdGp20twqmWU7A+EH7\
6262 yhbUgml1X1f7K1DwaRyUFN42F1uXNDdVtAmL65YCR926Tvz8aW2px8Nfmehz3EM+mgsoL\
6263 d3ZNBGEIXPGUWzXg1Yc5eW5/zBcy54aW0gWkFNWbqptceVFT4Bwvov32gwe8DLzDTMaj\
6264 aurg7t/bMXX+yw/egJGkOtkSy2d+qFbb9v0dvX5BL2TOR+wjfyb0p6U0XGOYNgR/qua3vB\
6265 Fgeua6gv2dvn8dFDV3r1dBw34GSPg910DG9h5XWkh9kAaMlyJ6dkLPzZmtD3cnu77vtw5C\
6266 h/YrG1p7Wxp/VvuRduc+wsq54ymm+8zzOgyRSPra4IKoZ1i86ytagcEPm9v/m09CUATz\
6267 Jow6tVnPCmXhZj+sNnPhsCJYa6csrRsmYRgk1wF4I5Uio1l1LRW7fmLEx3z2+gFW1LU2\
6268 Y57268EazkYoPctJi15Q1nJyLdrFRUzpl/3pmku/yN9gAoGYMTf7neV1vx/6CHUgh11uh/\
6269 F9Uvo+g7O3q7zrFL8xQ+zW+/8F6Pw6FV7xShiNlayvWdz2X1ULM/4u1lpMNoA5UGedoL9\
6270 zFA6egoxzhG6Q41NR5Doj9xuv1cy+rFbcuYVsnLkKvOCefphUbICLRMv1+9KPAvngH6G62\
6271 NCgMS1CSnKfxe+mtF1BwuxdfBoZqT/194225Y3TczrPQWhThG2zHra0/yb0khdpanzq\
6272 GXPF66/0cb5AHebzdznpuUjeG6Fow1geHmtgNCDeKzTlXuc3L4kyTfepuq5tgSWfKkAd\
6273 ufs9MF61G3mCz3EXORwVesqyPzTedh04/1kXGfawuXp66cG11+zme2K91bo\
6274 o2Ulms2WJEB0+f6k0DftNxfwZU9x70/bqzct5z0Poi0+vdpjDcdxr3D4U9XcHrLoSktt3uq\
6275 AcwtK009Fzn+qWtdKds60dcFodrAxneOCFRXWUS93PBXN7+ve+qvr506/2041XngLdrC\
6276 76HgrdvtH2W1MYVVqgm5zTTP5+7v0LRR/zJ1OYlx+8oh0Eeb+CV/0TUSic3NGfjKs30Mz\
6277 rFUt1l+Yi4fyAcwkjzqpb6HlgJebwpGLYxoo9/j8k//WW3xS32gQpHrV5aMTP1IDFN0p6\
6278 Ez5yF4HfMDX+/Buy4NVU73yEfB0K65icot+2jP+8qf4JKy1TnGRtB/gST0zMKAcQ18jPGL\
6279 A4PCXnPKKOTjREv84HpyOsws/BsqyTZRGZ6zr10gA9sBhEP46hsP2ratm0JeRugWBDB2Pw\
6280 NYD1B405TMcmcds2E/GG2zVrF7Uejsqy/7A7gUeH6Kyy19q3fpQ0vXtg4dz+Ueg+Lmy5v\
6281 bjjYt0+b5LSgq5Nz6nbwFPhudaYgem2y4aplz5dlbByA3NCTC4F3RKYfOTkAU9XryOLW\
6282 sDMC/H29v0GTNV1C+iZhTuz27rgAebkb4+8H3P553q00yu/WHj21Zwbd7z2XLU4f41qm0SV\
6283 2GML+6KmhovvaQWgnellYz/gLlX+IBNcn2Q7F9Y5XQFN/qUa+Hr3UrAG1MTRLg3bfpYpTp\
6284 m645oyCzeJmzXnQ2jAqgbByMxSL9vzQ8gBfXbjHJbpbXzM+V+KueRBRiotE/Bw8ogf/LI2hY\
6285 9/TCnsb6817dTgnQR8LEvt2z9eWt58jF71FSz0V1yftLVqT0b62ctccbr011HeS68SvY\
6286 20zUdeqWmRTW57ng7dKrVrLzr2toMPBK73na4Yrd2fm+5DzsymYmaHnCl0kvPOVHG5FRQ8\
6287 wC76Rw09DKx5MU9wQXMaX+ePguLw8/dvif6ULLPvsBpPxpOn1QwagE1sm9ggnXct0E01v3\
6288 TBBBJAdHMPdY0/g/1r1wbf4455cNKGKwAg7DsuJ2h16C28bk+1u2u78FYXWk1Q4/gZ2X\
6289 rYvjA8BoyW6zwc9/Ojwz7PvtvLLpNG2ULo8RPO0M1uovvDTLjYxzcNWEHjQW5yKfCP8\
6290 2h1f4Lc1cXQp6nms54YsKe1le0C95+XwEj3m5mc+jMe5b+1yHzYELGj3RmDvY/HMK08\
6291 aP7Mds3PueU7z8DWnd0vszXVF/xsFe+Lpz/wjQ09eiiH4ZwqV62+CUh31McnJSHfXoZHE\
6292 wKz9gPw1rTCRJwJwh5+/ocSLQ1zG52BwITg+wOpqXRVwEwcaRfdrbSG5cb/PySxHhPak\
6293 qzXy4110uAB4k5w8qdsH06+h0nWzjFYAUvY16Ece0017SA2kx0UgxtZB9RcaVYxx\
6294 2CMBjAdTcrUwWyKriwy4myTH9zt3R93/8X1j0ESWety7qF1j1odwKAmhFEA2K6D1wNEuH\
6295 H52HuWwIaLQHOQUZwr6yznTLs7rgu40YBJq4JWBjCayRHrTyeYx48/xCw+rUS9L5yc50A+W\
6296 8v0wN2ZAxw7VADPzceDpXpdsLX0KefrWEM+yj47aEa7yXzjXm+61FzUL46ch7eO46Q/m\
6297 wncf9BTvXbs6Z3hNXPvmlKJhJUBTFKRrbagLQCWiwbu1iPtykLhHwZaq8YKoeMcji9Y9L\
6298 Pwk79U/55Bk75fSXMcwhj7Y935x77q8YspvTbqSG+55hdjnj6YS6ErfyQVOL2xoeLrbmW\
6299 YwkG5S2p1OK5dZjgs+2LB1B426/g+u0sa6yUWOY1jzcCuoG41lqxVQ0Yep1wulXUL4pPR\
6300 zD3GL6w1VE4JA35xePk1NLSubb/34RcwB6JGzGz6rfLBBjBbJH71LWbGDRvdb4ieXppPbhN\
6301 NQT31qMHz7ETHvRxn45r8FpfQWRNdiqVf2qB1xEP16+rqDLV82CTnVYids2JfBPwMJP\
6302 aW3rYBgm9qXMLmChjCnvUN5fKMRc2LbzJbk8mU55cn4z/2rLDJQznjtkYyu01pdqccfMz\
6303 gKpG/ahfXooVi+JTofImZuJyn87QHMhAmdAaeUx6c7F07SUUkyg5Oz33v/20C7b+sch\
6304 Ltnp1tH3YeH84ipGt4JWAnu7Pn5xwvjxB41MabC3Q8rflZPCJfTcOSF0b8NabzSFwgfHbU\
6305 nmlDj1THGh3eSRt+42Mk5KwctXpMe35RJTvorF3rmn49VM0gFf80id191X61dVbXmkqjvb\
6306 MydY9m8Mw2MLKZeSL/VzqSKDPzcdiCyTe71q/B4XFKQANeK3M447r29fQl/ga+/vrBD\
6307 qPTX0U9UWKVME19MYLLzjVpzzxu0fP00/pTedh04/1kXGfawuXp66cG11+zme2K91bo\
6308 0xuU119fObLaKqG0hafa5NVPhxjK7X0Lu0Rnm+JAFefsnnaKzLhRXLyf5ed1WdK1/wD7\
6309 fd+JL72vEtDPEIggWkZj6zFP/d5duzt+2HihxfKlnhs7umT011AJkyVScenp1WALAAMCzAE\
6310 dqV2Sx/+/nLn0dPelXVtd/SkUr+JL5/9VsbL75z+bVNS8Q2EuQn/Oa3x1/FJZS/VZ30EGCq\
6311 ePdATCYORCKr3q6vL0p07fXfXvDaVzcgjOE2C56YcYmzC/7CyuWar21IN2xK4NOC075/\
6312 4yMTRK3XuyfGJgmxT/xdpbt8uSRi7F11luoFJT0m3U17cKkfyqWsdvVpVg9RPAeh07FRv\
6313 hUL4693puU1Yn+FXOC+CyOvr1WxzYlh/w3n7fiiibreUtsVURMitjpkWRVYmPKZMHZ2FciM\
6314 dMf1f6+ew10/651MmCDDY2F12dFycqj38ARabQSPGX1sCGUCaKrD0UysauzvgcZx6zATfE\
6315 LLGqFLXPjYjYlthCkphr+c+n+r76LoLJ1d3d451+sndv9Yr4veCWg9+SrXtc6G/areZLXB4WX\
6316 tgzv7Wk4n+28f/FFzZUKtA3ky5ULmo9C8BN3HglLin151SRny32hsXoRnTBmbWmiP9zT7o3\
6317 j0g8vnn3zcecgY1gCmlw2/fvICjoYxtleoL0xRvGHmYn21/IJtL6Ww3j5y8j711dyU57\
6318 xLJDmJm+xOQgtrucgEUTDVPfCnovWafZKAeVArG5T3tjBQGT+5rCIU+U1BzPPIPJumpRPV\
6319 4YE2w9wP9xlfv/0ppuyxDp9uNpYih91/XNkovNS4dGG8C8wms1CzfrkCQUTCZSHj+wm8G\
6320 JV7X3eXm6WjLSr6LV668toExtHjJ/4Cdw24+uzFvsJrSt11RkFOOALtznFzdf2SD12QrO\
6321 8Y588Pdsb0VhRLQD6exvrEOj9y49DQPKC52mjjyz021LDv7yb3zL8qmsDMOFARTVFC3D\
6322 N1NQGW1JebavQMrZ78D2ZvFemHcPFCU086nbfBB5zKFLFPMRHE6F0S0At0vm/d8Vv8km7D\
6323 C5YrseFVLvsLpbx79z64erdNyulK1leJdaluA7j0or315x+Y9CbQBDP/cK7JKHdB\
6324 E5sg69OKMH9pRdJdv3v9vYbdQucocSLVM9No/QaPF3KZ1ve8zWcmJjK30kX+30RQ86K1wN\

```

```

6325 b1xafe29JqBL8of8GKam6n5P9mdGP5bmUikpmc22t7RbHskj jP0kmCktCf /KAMlS0JtEjK\
6326 v7q+ /OzmZbn/ Z5IoHT3+NPgzn2eyx7ui20JDm9xoyTcZBT0Ya+vnvdqW3URPJ jYxbm0e1 /au\
6327 zq4BcYggsImpGdLlKxcmLwksKzB6W9A90stveB+5f714i1K30i05Mkod+r9 /ZVx809P8C\
6328 xp7KQYUwTqmcct0+NeY/ 99v33xh71zib0k3cnot1j4fCzknzaspSDW /vDg4XPKCnb8+W9p\
6329 Szdzkuz2Q3fev05llytqt omo30pzk9Ecs5H0W+FXV150r6zr55HkDFMGAAQ3QyA09y4fDfj\
6330 pp5fJkq6grNyl3DfYk15h40Kj1a2ehBJ9NWTfBAGv9luIawS2xVtAhf8S50dfTpseEaP\
6331 mRiFlX0mXm4xnp /fBy6aVg2ft5y5SkWno2MPF5F3sGc304UGGSj /wI548wVlfVvab720b\
6332 Xx/MrwglE9zrXP0mBx5C1Af jIHyXjhsR7BkkmfG8mL+D3CdfJF2god1vNN3V3d60w7hyf\
6333 koSV0pEpK2FeqJwQtLd70c6dnp1H7z10z933h0LHWYU1ReH27ptxeVe69XWH+3Jdsamt6o\
6334 lEwSv1G5j8EaJ2NR0adga71EvOR2LBSccV8Z0u5UelJbspxVGHecusJRkKYLW0VSSUinTmW\
6335 LaycFxpHswIKQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1\
6336 QA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1\
6337 QA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1\
6338 QA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1\
6339 QA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1\
6340 QA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1KQA1\
6341 m38w0ncAAAAASUVORK5CYII=";
6342
6343 GShellInsideIcon="data: image/png;base64, "+
6344 iVBORw0KGgoAAAANSUHUegAAAQAAAAAYAAABjXid/gAAAAAXNSR0Iars4c6QAAAHh1WE1m\
6345 TU0AKGAAAGABAEAAUAAAAAANAAPFgEdAUAUAAAAAABAAARgEoAAMAABAAIAAIdpAAQAABA\
6346 AAAATgAAAAAABIAAAAAQAAAAEgAAAAABAAOgAQAQAAAAAQAABAAcAgAEAAAAAQAAYKgwAwE\
6347 AAAAQAAADgAAAAA2CVJowAAAlwSFlZAAALEWAACMBAJqCGAAADJREJFUEAhtWt0VMUz\
6348 npm75BtEKqRlLlVf0VQ0kG5LsJvJax2Kf61vU4qtInkFqR9XJkUz0mFvVpSri8EV\
6349 hGStkN0Qj4D0StIm0AEKkIKY08du9Mv7nuZuWYXZLABg1wz7k7d/7555+2b/5/hv/MenL0\
6350 isoCNRkR5Dk6VrwyC22wnqRRehol /DSBKfAyhBKRKARLJEOMRDwGog2Bv5ayJ/A2geQ0Ugt\
6351 oaxWCFPHhdj053TDbw1m5esmaJL5kP5HJAM2dW/prwGqGQXgHf+AdhXeroGQT5WSzIGIZ\
6352 JwWqFqPnE8DOvX9GnWm8z5dYvcMnwak9+fh4q/AdY/Fb4plGE0VoJge84Wvh9AB24N1\
6353 B6hhNDP0UAVIdZLuPaUz0 /5TubvyyeQAjgZ4xndBZtMqJhIBDKRk1VORfEITborb515c+m\
6354 7Y52I6H5fbTmNGUimsAnYx9BpNfh+T+j8vdl8LE+jXaaU120ApudUngzhvJyveq1RcUYW\
6355 Q50Wdy270fwwRlHTKykKpKexq4EuFciiIGevsAFfbnUNX6Ucco/Eqd0AxxnQ5Y3r2o2wR0\
6356 16uMikUrCmlrolDXqIpIz6o4nzF204Reg9dLOX+gOD+18mALiRqqjhyvNkHuaHjYMF1Da9\
6357 umL8xKadrWBP57982pr45r6t12PsnY2ytMA4esSd11x2o0UENKDP91WeyvyiANo4APVX4pz\
6358 ba8eAGv+znyJ7+plQ8beqQ5CFceBwF5w5f5Y /7prt10mBAZQXqh3vR1GzsdR5Vh53b0Ha\
6359 ouguWbi71fgL8+eVx1T11iXDWlHqXLP+cmP9ad+hoQo0v0VMZ5W8hcyN1+h0rFkz4ujuF\
6360 9gbe1KfrnZd1+!+EssT4LdoN5U+nreLkVbsNag20Z0Qx4nM4E8X59me7EohvZnH4rTeh+ES\
6361 mwr+h2JYsyr02IcQD0yP2PRY4VElBtXKLAszoQLumom25UuwkbuKlPbUK11+2Xk1WNNYPg\
6362 wKaSRkLJJUa7x6/cn9BHMS3MRZqULWKEz130S10UqY1dRjgj23F1hh/PS8uuHWNKQSDrJT\
6363 0Zz49Q0U3/wds53PHLw7K3U0dtORUjgYkmbCj1SaBaFyQ1cEB70U0s0K8rB\
6364 b7SLJQTH8x3yWwY7f8MHUuzHkZLLJq+GteBca+i+J7zBe8tSSG100dH0yGScFuacL/x8\
6365 MH0/3T8ncstV/3UB8PphuvMnr0iUvUv15WpK/IDzuzvS9htTqcm20CNKMrh4p4En+CFZfht\
6366 Pg/17U7c7qx4mUjVJbczn00xi23h60FASbZ11SaytZHSu6d+3ZQ7KfCS431AyrUmjTYh\
6367 afpvgz0EftunV4vgVno3FeQ080tGoYL9HhHxunB6t2JWVFu47VgVoKlMq+0zV6kH3GUSkdi\
6368 3Ek0510/Y7Tna07PrZKovN+W5NNmCUGWMSIewGrWLoGpANTW7zt/58p6UA/uFHGMCH3N1U\
6369 Qg15V/Omr /rHnRWWHlqM3E5mb76kukVC5cVpux0QFRNk6n6WBAfGnRnLymzHoYea1aD\
6370 M+B5tThvdF8v1DOANURkYw6FT51h5kGvOY3e1h2raG0gbi4YshFq0wxh9Bczro0LTOxbx\
6371 4Bc161bmSm4ThfWhgJCEf1g9ZEqb1h906C6YsbJLDXzU7J6bdEdvkh0ky/wEdiWLVJZazgz\
6372 tLaA065kG502du4hgZHeAYPG /lBgMwkoXyojF5J5XpFOYXK+3+05mMaIDQBN0/evUWLS\
6373 sxySyf4z3jEcrfLzlbIPgPkU6Lfg05fbBEW930Y/DBStB84xGrAh8HH1ByoSmEk72epkh0\
6374 knzu1UQh2leFsqBte0Oq6HxVLDc/5boj8PrGmL231UNMGEBE75CAwJLcZ9i+hDa9KYm6\
6375 0U1AGRMBUjEXCaDOfJLnBsB2x2jPpEoBpGM0DU7FhoFnRWGPALG7oolU8SS3tHRakjJHs/\
6376 MiX9hdgK/ fs2T1TUsBrKCMVA+FksMLDcc6Pbq1XchUNLaogdaV8ouIdQ1Q5S54vd6UFdQw9\
6377 Q/E88btg/mvcurXQAMr9JDSUpwN2TzB4wh4EfyJ0urBu1cF09wle4Epw0e+EGE13F20U\
6378 r3JNaVY89nTKfKVj5H65WnNm7HQvnpKGAhynB8qUPHXuyy1K0gj1D2syk1EV713CeVl\
6379 nPN00000000000000000000000000000000000000000000000000000000000000\
6380 Hb/mbuy7QD24FR3Zcd1TpoFv8XwctntP61e1cFZVMZCKYXoch8L0ZyV0KfY9hncS\
6381 jYnFRRkhdFqpfwKbYD/qtQ1v0p95adY/BSN17laf2LkTfFO/dLV9whldpJ5btDh1hAYM\
6382 nCMY3RkStz3K2PnbDd/xlwbHg79XLZ4agz/tYJo0oFPQ02B/0h4aeiUQ9JINKd3j8hDt2\
6383 Bx1mAppP+PL49VQ4GHE61vHZFOhQ+ZmaVYXESH9dNjjsW5PEB5pflK5zFlXrBrVjDk3CtU\
6384 OX5Gogv8oaGfUrme0L3Rv508EgFtjv8r5iCQxQ4RfVxmkqNv3mrSrc7Pe8AikqghxiWU\
6385 Y06RuyBlts2YnQsoX+D2k5WZid5AWockCmCLasLwt4dkB6C6qL0nFhhWpT6FV0EE32K\
6386 PpP5LczyG1IE+Q2bKfJdQHP2xR8hWY7D11id0GkwYt9ymgoI03j3Gj0Wx+cA9vbW1gfaZt\
6387 ZYIhrTaC3tqsl1nDCYApDVR0NHBX5JLW1W8pWHnHDR0Bn1U0y/S1GHM13hb23K4V5Mvbus\
6388 rxwAe619U31g+AKvBU8tUgqDNH6q2XomEedj1buXn9dN4Lep52EVreT/IAJ9raJgsBJZ1\
6389 goR08DbaEzEFf1Ppsb8Bkv9zt/LPKnTY9AZFLlB134DA2NRF84cI4rdkHqK1wRo14uN\
6390 PVYXQzFuqD4sXLcU/HKf69s9Pt9FspuhDgdL2DFPhobBBaupfWoaYaz9qiy/eUumvtf4\
6391 eTbLS20uceZWTAEbJc0K386rVpmZ1Ysyo600s0oogGbuUazgQUH8g255F54k93u1e\
6392 k/Wt+anFaspPh/yodkGR6DJPkC15tjBSDOTORkGmycNcB10CCar9zvQrda04DJTLdYzjrfL\
6393 9/MMRRQvXlpap94NPWWjs4mdZ5ieB52RLjmgEco11hmlZXFAMs0dyKoNEtgBpGgBUz6W\
6394 X4F67eK1sYs94vL0yrktRo6932be+13JCGdqNpKhwQ09iDjUelqRcbEetJdUkjevXjJVV\
6395 XDMAMEwPBPv1L1LsmwSukQcxceZgTfWuzrjHEMy6dyp+L5567yUGagCKf0bB1JK8Gw9\
6396 zV2f8hR0Utd/lnj8pY86z1zscwKN11K1kIDM341Q0GHRjgg1AEZwGdhh+m1IAASSBE\
6397 QKcXcYEMPF370mCt1FbP8z1JodN64F5f5kz2+nTv8BMD+8Fh310bU\
6398 mXkQwUvPUDdM1QXQVNG2f80jnb15ca58RSuKhOgi2xjNwG11EV2nH+b2tLSR1b14xCuV\
6399 d6CAqDqWqCzq7FU2m0xWNFLzqH4q5F+SYkrZb7MiZUj9Whf4iAu1N8Y52Qa52Bgo3jG8C\
6400 wj35kMhKf6p240XALahvEd8z2daZafhuwE2nm3Fbhkg0T/DnwFSO/wTncW2h9W8GSHSD\
6401 i81Pbr9kwhLkVdLav+8j3VID+rBUoPVhXYD+zKMBmbac9z+caefAku/CR0s/DogyWl912d\
6402 eqwxRyEgweJ3GEa6Gf5s7Lb120pxW2wBcINER21rfvffzqx/P/MDpeJ+073S64yy+oWXX\
6403 f+rYHZLV1ErIjg/7paY9Gzflfbz1iURX6So0jtB18GTYMORXOPNjYVK7oXcDkBrAke1lDSae\
6404 jXRWA06szhdVz4RTA8rPlhpk1/EsENF2dVnStJR5+OCGTOXhky/ArX425UYXBBKRECljw9j\
6405 z9N+5j+qpeG1Bb65rsrTRHJDL5pFBfTmq9/dB+pmHz1yPNPCyDx1SRk53t5B5011jX1M5EC\
6406 I+YkpbJIGJpq4BDYpJxnhKceAR+27Mqx2EC/ganWzW8JD71VCJ13GyR91NMC/38E9AJE0\
6407 m5yzs+osgCmvIE0NB6zK7BRQYSTPOeDAz8ZJ3wp54C9p8r9rjyK06mteAzm4Ndy35a3yVA\
6408 ZW5TzReppm4PeE2rZEkngY0JozKtMJ08uuw5jV2XXNtFsQUHYMG8UqVQhEAX6GLuaUJ7D\
6409 KY61+y02yYui10T9ctCiH51c/i1xdBY9t0waagLhOUX280DdehRv6BNhYDQ7HrSgqS3S1\
6410 Pd3W0CGHWD48gD1xPpYQeQMSBUQtMeyLT19N37LC7F1DTunY1UzB+vwvIFBa5/gz88DgpQ\
6411 CzjSehAPx/eHvWJ3/tv3Xb692pwC8J9MANw0ex9t7ANZ1V6dR3f3U+aECVceBy0pUHYg4jD\
6412 y0UrBfJ0UacOASX5zWz1/5ksu2JuxJyA17X1B4YVq78y7yKagotQAVQX1/wXj3cG5Yd0D1+G\
6413 z2+5nvgv1LrZ0MKMRzsz6G1cCpH9b1j8gqC+hD7Jm2L278g00Cru1d9P9w6h14exkR\
6414 gtd1Z80WYyE1/pzBfKfWmB99d8PvJqHNOvevCYkppwJ5Vw1y8Kh9/k7R0kD0z8kd\
6415 Qm58uAuQzxpLRwv8QyXz/g91Vb859RLM+PQ7ozXMMAlrb6dXu/sMeB/ogrCO3nANks5b\
6416 DfHqLANOQ0sa0pxn9K3VFGuD00XMS40E2La+2j9W065IrvoXfngPgoT82LkNhV3S038C\
6417 xvKNOCGoppBkNevLH4qtZcWdK91t9srL4110xj1e1XbZ3ums8ulcPmPpC1ahdeeVwS\
6418 z731+zfa2UnvuvHwAeA0a0N0mHnLMD8dpXuE10U/Qryyft+9U6T736P+kARQ6m9qdE\
6419 mnxBQRkUMt7WpPwtD3IT1123z3zF3h1VCKfSfBFCN6+/qjD0qz6rUT9VEofa7vq7m5tI/d\
6420 295Gh25YkHSDyq9t4XKHq03sYf/3Eha+AxrWmBhEc6uq6XUuBAKSDcOVX4G7eJOYsm2H\
6421 REMNEKF9WdhLS18hly/wA+RiDRJHOJ221j+pl10A5DH4OyaQkOcuVz1B14jRwHH8CD+7tA\
6422 0FuxPWSASoCgmV4sZzHw5rXGyecX1wmy7+E/+1qFeUK/R89kneGSW7wAAAABJRU5ERkJg\
6423 gg=";
6424
6425 GShellFavIcon="data: image/png;base64, \
6426 iVBORw0KGgoAAAANSUHUegAAAQAAAAAYAAABjXid/gAAAAAXNSR0Iars4c6QAAAHh1WE1m\
6427 TU0AKGAAAGABAEAAUAAAAAANAAPFgEBAUAAAAAABAAARgEoAAMAABAAIAAIdpAAQAABA\
6428 AAAATgAAAAAABIAAAAAQAAAAEgAAAAABAAOgAQAQAAAAAQAABAAcAgAEAAAAAQAAYKgwAwE\
6429 AAAAQAAADgAAAAA2CVJowAAAlwSFlZAAALEWAACMBAJqCGAAADJREJFUEAhtWt0VMUz\
6430 x9/b21z+1YKCK1IKamW1j /jH6K8stfEFT7h11GPRWdStQoKdumtFw2nFq01YTL1atimZ0\
6431 am8Aq6j3YOK17KqglarVv743BQPKvBAJj3e3r94WcJpe93cmcbj784kd/v7723+3nE\
6432 ff+nx8S1AESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAE\
6433 IAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAESIAE\
6434 XiEuf65tAhEwaD81mP2wLTwTadyBmzrT+42pzRSrd3peQvpKSmr.rhgNcN8fwhXFUap+zyH\
6435 2UASIAESIAESK0II+LPR9dsk5ELvxdKBZt1hpQ0kIEfLe+8Jan8AzkmVA/67BKiek+pmQ\
6436 hsF7VweE6PyDz+oM6JmAxwznn6REVCGS4SOXwhL18xtFFcuWqH7AMRgIKQA1KQA1KQA1\
6437 nbqem3098iHOOS8sa50440umz+EzcEAE/FWHXfnj5oFLd/YP802NJKRALvAWlqEgG4/BSG\
6438 rgK0AM0/d3uCU1WjYiYvnsIY0KAQ8FVeYnmsYwYJRSF3cgmUmtL6v/fDD1QAV487EBpZyQ\
6439 65pV5cdd246QncHyziL1LzEdtK7m51Aww2ymTmXNDa8cLepLjLW01oLU1/s305692znGp\
6440 2DBH8ZLDxp+28KXICyGj9Fve6Eh5QVCINOVQeOKFL1Ka7gpKvWbUnKnFoods8i4tKyWaGPT\
6441 e0Lc2e8+3+ralP162LEVYnPs6SfapwSgmV0Kf8upDWGzLSbaWPFUdretUtyYrEPWH9e\
6442 faWfI9Q0t4Ccz7Y0r0BVF9CrE/2qnEo/El1F4dDqHalosCU4rPzSLGNJ56Z10qVrT\
6443 HrDxjvvn8YmysWPTq6UEZEBGJeS2Ewmp14skJ8SVQAt/zSeLluz5aemLHZ1rvkdhvAHO\
6444 P6R5ZafF8510gmVortLSeXg/OtLB9s+r5h8u0ihsYfz191flp2cN8ytL1A2//wU0J8aEK\
6445 iX87zhymPtTb3oc1bxXa/DKX3bypoeH686jCSEXCvgXALY+CVN0808Mm19BU00h+oIH\
6446 ZFN7phGbb3Ck00JF09R2rRgVn3d1QNRtG4570Ts1hkYsJ80k6478h1pR6e14D4m4U7\
6447 4tXw1BP1u1BE6u0Gwz+T9jPfnKN7Wbzm5W9pJG7K13801ulcApeIwCLADauzqEoc1Ys4\
6448 1B7b03bKUEQ4panzx+9y0NN9ANsdFQm4XosNokzgn7+f0cANUSnm3XJnG40v1ZsuC6\

```



```

6573 "21tp2loJw/+2rP1ZiY2cwi0a7KV60JRqoTbcLY/pXG30ZampCeRX22RpjvzoqHCGewntdHRU"+
6574 "eVj9N7wshZ06vtH1/01rZVCUMhmbGqB5084DC7hF05RRaT1o1hgptHrUvyOEMdoFmQERBbu"+
6575 "QnFsgRwLVR4NSUeieKk7m3vtChpPC1h41kG4HF04g2mqTQ:RhmikJt4F0j/QIG6GablEswkK"+
6576 "jNFzP3mHMaOhlGsgovZjP243F6TMR/ihYgnAj3CLtGABjPLVxx+qY7c5Sg7efflo5zko"+
6577 "Z6xdkERKwtiQOPp91rZEMjJ7x+o8a9U4m/ppFMRpPUCDzJzCLRqCkGoaiqUEakYEB6rROB58"+
6578 "No5QE61hs884PHs/jOPEmmao/9NtXvgetyMdkzLnfnvnVQUsjClwRtEanMtGuv4wHA9"+
6579 "iyoNqLkCk6x2TquCnpqOxULecJTWnz12mfRr:JsbJVV7jeWk6Ug19hX88Lx5XLznoq3n2"+
6580 "me8nyoI13kBJtASGuvrrWp0JdfJcwr9HwLNUU+PplgCoIPAM7fmFmuHJ4doEretI6mWkvrV"+
6581 "7PCu6/KphjzRkRqeQ/6FS1NBUEWDAkMdQ0QABtIANKTooyx7+72oJOUTx3vk7Zgpwx4Fh1"+
6582 "u2mZT34Qm23uHzBf+oDK8d4m4e1jz6nsZmmLkKhJh7MC657N23sqiAbRlPpE21AgLsM"+
6583 "DzATP868r7jxog7xxgJQE7E2M6NT465GFHxmFOVWZMzmeUe2iwRfj3K4DbTt/jOpChzRmi9"+
6584 "GhqpX70zeGvhlmlnLpoeCioBrSlgnc30VnWwIoq3LhA4b/ImUEYt90xvYZT3crT1tasGbm"+
6585 "MkpoFyQ5f9A8e9/Vi3+9Mc94juda9f1p0b2ZCmRAXqxoep/5SYG2QHUKmKsooNyJlGCKS1sw"+
6586 "mK6QtN84QoqShuCXlOKAgplUq+WcBp30LNSBQEFt++q78bPL0ndxeWnG470tPgU8IgrLp1Tj"+
6587 "odqoLNDc8jz6hibRjhpFqw3irVjRr:jnJJuzeZj9i+brGgXJWDGWNpGkZn3j1QbwKLMWnr"+
6588 "+068Z63ndVpSWzn5mNJVIRuXz3F9C3B2MKX/22kgf/B3gfw1/NQbVQWVG6VWQyLntEoWkz"+
6589 "SgWlGRFf65OM2hKjJhXtZw6j6H6S5gmKCR3MJV2YPMhF9bXRFHXSasWOCJ292pN1w2wrjdm"+
6590 "1z27+svrjVmjrlKhCfXnEUFEDHLLosuNPrdfWxDUt/7Baac4Cv9lyFMFLW42zo2cLh1DOAX9"+
6591 "eA02H3GRqPLuZrZz/bYFKSAhSeXb99ei1MZL19azEYDVHLaAj9iNFqCmJGf1h/iCdHzPLYu"+
6592 "SOT01bV8903yfrNjrtDjYObYIyFD7b04RpeQ1YRBdW1zLcqmG/+CLWgzbVHzmXSWfajj"+
6593 "xwBRBzFwF0d8QAgC7/69dml1FXYyGS8wQ1zEzB7BewrtAe1tpkEbW0FP/KuH6gVNmJFK"+
6594 "iJtW5OC3xrrYmUE5LzRgnan9qIYfvkjbqPLHfyzZPpcDz5Uil+zT6T4sNNf7b960tBBFY"+
6595 "S0NT3eU2YdyHMsDp27xY9YRHEi:R7+dl6Ql8XPag/tWpBubplav83zAgQ4dPOH2ibxfzpu"+
6596 "HYjh101/LacjBEZnicd/rj6IvzChwey82ImLGENvIYgs+rohEaM7Knu7DumGsyALNARlmq"+
6597 "qzZG0xjPULj6y8SGbGANcZ3G0MeaV2MxM18cacAGngcUUPSvBbeHxWAsEbJcuoP8Hj5g09P"+
6598 "mrzXqpf6CqeQ7fzb6m9pdBP8X1xdw7/7AW9UQTJcWpt0L3K9E51VFdtxgi890GDx64"+
6599 "JQ+zqfFmo/errScxiYef7zqf4AAVCAhXEY1f+Lk6jHYPS0H74VUKEHf0mPEzOk/jAg19"+
6600 "KsP7/2nUHNoStw5y1jPxoEYVMdvwEaEY6JLJoG7vUan365OguXzM8w7d/WY6nv2B0vMW"+
6601 "A4/3B7F7/gnuPfltcOPZV44ZdcljGPKXdxhYkYxUzGBNdprLmpLaas75yYeuc18HlIkX"+
6602 "Tuneq7gjhXQkvbfu40H6Lxvu2afNPF24cb0c8r5bhqmMdpkKzocGAh6I/4Qwr8aTpgP5"+
6603 "MlKa+a47GxUjWxyeMaK51S6yup3j/eUohjF86qvWypHuV6d14VN+f+e/PsCA9a6chkjKb"+
6604 "1XKteIjHZX1Rwd5hhSAbKzCzEp5/5LZxh7jxX0XD13HW/fXYRddm4vz9WmyBCHRScLPfz"+
6605 "r50w670Gw/cq07uxL2QyO/q7Ybft8Eq/tvn2GSOJWR/vDzh1D05GYTXj9Mx0MkAw5fXRE"+
6606 "uColFLe8CbZbfekmyzivtWzE13M85NhlJyTWyLSScTAhG/+1Ejdn5Sbs7dq8zf0o+mP4W4"+
6607 "tAH8c6Ubd66Gp9cN8uk2yTL7Nhd21XD0yYGDUEkM16PcCxEs54md37/d/km30BEKQ3"+
6608 "62eYO0BJPNiUeW8q61mLm/152pjFIUTRj+6zrp47dx2B9fMxnXLMsotIZBjG/WgrpnIa"+
6609 "25kAWsVqcf3VUDQVflpjrLXcn169AjX0XQD6bnJyGFByL+GHHEebEO/Jmpac7mcbjKTKz"+
6610 "fivcNlz+DlJdr/Sf1+5H4zDxmmp0Jh1xXF4ec/mI4llmPdaxK03PMCO5UeCOFP3gDdel"+
6611 "nsgVWA2YAaVier/s78JnJx02jirdg8j98vmJZEUH33j7aZnx/Hawqf1fhjQRI3AhInEpi"+
6612 "WlFseaFKnxnoQdZmMK3YU0u1jONL5PnfAJlHn39y8No4xf40r9Hkntv8BwQNapoM+Tbdt"+
6613 "4eg8pjxclS6xMxzLLo2ouPnuLNVXmewFtry/gkpCzRHxHzLpdY/6+rqk5BvmPnRnMJX"+
6614 "01k:5xYx3766OMU942Jdp8L7B0XRQRHvXmtGd3791Evdur8Uq7h0UMZVv4+SVQAd30xy"+
6615 "swBEOuAcoZ2MIP84YXJuuXupplal2w6JdrPhOrdTEp4v7Qd+5nxc1c9B0Wnps7oot1ey6hG"+
6616 "V6A+5aJ3kXBlfHbOUB4j425K46aJkealKSX8YkTh7hobswILYxiK3X3q0AvZzY8a5+mlE"+
6617 "YoePjCvPskdogi9i2m1+maokDLGGgWBDH/KFN8MLzkn0MOMJlvhgqdf8M6V4E6yFQTVWO"+
6618 "v/3N0DR+MzSV3wzpzP2GxsaGL3+iLLAL+6owv8da8M5GmgXnsBrfCI52A2xCTH8JQWP4"+
6619 "FfdvTE33A3vnrKmyZLy8tOWgQKupjByVhdHb1+L4as06jwhnhlegq3d0nXufystZbqu"+
6620 "kgC2fFyH1jQWmtbdk3+dVYH3W+G5N8YLLVyaXp106yve7ZFO85tnfDwbgu8Ewk4T10/zA"+
6621 "LsyzfIEBemBzVnVOR7bpaZahUMrikydY+mq655Lm8Lvg6NiAYQs1TnGzBNXQeCkCPYwq"+
6622 "XcB0tRr+NtoXNQITEVjLaCy/HOWWVxshF7V4maojOpqpfJjJHoYx2Y89zggSEH04r4"+
6623 "i74GwVYKSMR6icARoQsMSPa0P9F5UitR36ned9DA2BWTCThytOTn6MkxuIKpt+UtuIMzxxqj"+
6624 "tCywrEjJGXWBrJRI6uvMtyGsurU03AiQoTXG32S8K/CTC8mo689CT5FbU+JLZAYikk+g8E"+
6625 "f+TzE7tu4l/VV1En8ShhhNF4R/BK1iajEDYb1VTV6xDHsOyRbXWJVF5IFjDY+rz0Q/USKAY"+
6626 "Xwcd/s3LlK6sPtRCP1v3UH1Crtyps/ns/BCUfMtF1Hbf5/P/D94D1z5t1uE3AAAAAAE1FTksu"+
6627 "qmc";
6628 </script>
6629
6630 <div id="GJFactory_1" class="xxxGJFactory" style=""></div>
6631 <!--
6632 https://developer.mozilla.org/en-US/docs/Web/CSS/line-height
6633 -->
6634 <style>
6635 .GJFactory{
6636   resize:both; overflow:scroll;
6637   position:static;
6638   border:1.2px dashed #282; xborder-radius:2px;
6639   margin:0px; padding:10px !important;
6640   width:340px; height:340px;
6641   flex-wrap: wrap;
6642   color:#fff; background-color:rgba(0,0,0,0.0);
6643   line-height:0.0;
6644   xxxcolor:#22a !important;
6645   text-shadow:2px 2px #ddf;
6646 }
6647 .GJFactory h1,h2,h3,h4 {
6648   xxxcolor:#22a !important;
6649 }
6650 xxxinput {
6651   border:1px dashed #0f0; border-radius:0px;
6652 }
6653 .GJWin:hover{
6654   color:#df8 !important;
6655   background-color:rgba(32,32,160,0.8) !important;
6656   line-height:0.0;
6657 }
6658 .GJWin:active{
6659   color:#df8 !important;
6660   background-color:rgba(224,32,32,0.8) !important;
6661   line-height:0.0;
6662 }
6663 .GJWin:focus{
6664   color:#df8 !important;
6665   background-color:rgba(32,32,32,1.0) !important;
6666   line-height:0.0;
6667 }
6668 .GJWin{
6669   z-index:10000;
6670   display:inline;
6671   position:relative;
6672   flex-wrap: wrap;
6673   top:0; left:0px;
6674   width:285px !important; height:205px !important;
6675   border:1px solid #eea; border-radius:2px;
6676   margin:0px; padding:0px;
6677   font-size:8pt;
6678   line-height:0.0;
6679   color:#fff; background-color:rgba(0,0,64,0.1) !important;
6680 }
6681 .GJTab{
6682   display:inline;
6683   position:relative;
6684   top:0px; left:0px;
6685   margin:0px; padding:2px;
6686   border:0px solid #000; border-radius:2px;
6687   width:90px; height:20px;
6688   font-family:Georgia;
6689   font-size:9pt;
6690   line-height:1.0;
6691   white-space:nowrap;
6692   color:#fff; background-color:rgba(0,0,64,0.7);
6693   text-align:center;
6694   vertical-align:middle;
6695 }
6696 .GJStat:focus{

```

```

6697     color:#df8 !important;
6698     background-color:rgba(32,32,32,1.0) !important;
6699     line-height:1.0;
6700 }
6701 .GJStat{
6702     display:inline;
6703     position:relative;
6704     top:0px; left:0px;
6705     margin:0px; padding:2px;
6706     border:0px solid #00f; border-radius:2px;
6707     width:166px; height:20px;
6708     font-family:monospace;
6709     font-size:9pt;
6710     line-height:1.0;
6711     color:#fff; background-color:rgba(0,0,64,0.2);
6712     text-align:center;
6713     vertical-align:middle;
6714 }
6715 .GJIcon{
6716     display:inline;
6717     position:relative;
6718     top:0px; left:1px;
6719     border:2px solid #44a;
6720     margin:0px; padding:1px;
6721     width:13.2; height:13.2px;
6722     border-radius:2px;
6723     font-family:Georgia;
6724     font-size:13.2px;
6725     line-height:1.0;
6726     white-space:nowrap;
6727     color:#fff; background-color:rgba(32,32,160,0.8);
6728     text-align:center;
6729     vertical-align:middle;
6730     text-shadow:0px 0px;
6731 }
6732 .GJText:focus{
6733     color:#fff !important;
6734     background-color:rgba(32,32,160,0.8) !important;
6735     line-height:1.0;
6736 }
6737 .GJText{
6738     display:inline;
6739     position:relative;
6740     top:0px; left:0px;
6741     border:0px solid #000; margin:0px; padding:0px;
6742     width:280px; height:160px;
6743     border:0px;
6744     font-family:Courier New,monospace !important;
6745     font-size:8pt;
6746     line-height:1.0;
6747     white-space:pre;
6748     color:#fff; xbackground-color:rgba(0,0,64,0.5);
6749     background-color:rgba(32,32,128,0.8) !important;
6750 }
6751 .GJMode{
6752     display:inline;
6753     position:relative;
6754     top:0px; left:0px;
6755     border:0px solid #000; border-radius:0px;
6756     margin:0px; padding:0px;
6757     width:280px; height:20px;
6758     font-size:9pt;
6759     line-height:1.0;
6760     white-space:nowrap;
6761     color:#fff; background-color:rgba(0,0,64,0.7);
6762     text-align:left;
6763     vertical-align:middle;
6764 }
6765 </style>
6766
6767 <script id="gsh-script">
6768 // 2020-0909 added, permanent local storage
6769 // https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage
6770 var MyHistory = ""
6771 Permanent = localStorage;
6772 MyHistory = Permanent.getItem('MyHistory')
6773 if( MyHistory == null ){ MyHistory = "" }
6774 d = new Date()
6775 MyHistory = d.getTime()/1000+" "+document.URL+"\n" + MyHistory
6776 Permanent.setItem('MyHistory',MyHistory)
6777 //Permanent.setItem('MyWindow',window)
6778
6779 var GJLog_Win = null
6780 var GJLog_Tab = null
6781 var GJLog_Stat = null
6782 var GJLog_Text = null
6783 var GJWin_Mode = null
6784 var FProductInterval = 0
6785
6786 var GJ_FactoryID = -1
6787 var GJFactory = null
6788 if( e = document.getElementById('GJFactory_0') ){
6789     GJFactory_1.height = 0
6790     GJFactory = e
6791     e.setAttribute('class','GJFactory')
6792     var GJ_FactoryID = 0
6793 }else{
6794     GJFactory = GJFactory_1
6795     var GJ_FactoryID = 1
6796 }
6797
6798 function GJFactory_Destroy(){
6799     gjf = GJFactory
6800     //gjf = document.getElementById('GJFactory')
6801     //alert('gjf='+gjf)
6802     if( gjf != null ){
6803         if( gjf.childNodes != null ){
6804             for( i = 0; i < gjf.childNodes.length; i++ ){
6805                 gjf.removeChild(gjf.childNodes[i])
6806             }
6807         }
6808         gjf.innerHTML = ''
6809         gjf.style.width = 0
6810         gjf.style.height = 0
6811         gjf.removeAttribute('style')
6812         GJLog_Win = GJLog_Tab = GJLog_Stat = GJLog_Text = GJWin_Mode = null
6813         window.clearInterval(FProductInterval)
6814         return '-- Destroy: work product destroyed'
6815     }else{
6816         return '-- Destroy: work product not exist'
6817     }
6818 }
6819
6820 var TransMode = false

```



```

6821 var onKeyControl = false
6822 var onKeyShift = false
6823 var onKeyAlt = false
6824 var onKeyJ = false
6825 var onKeyK = false
6826 var onKeyL = false
6827
6828 function GJWin_OnKeyUp(ev){
6829     keycode = ev.code;
6830     if( keycode == 'ShiftLeft' ){
6831         onKeyShift = false
6832     }else
6833     if( keycode == 'ControlLeft' ){
6834         onKeyControl = false
6835     }else
6836     if( keycode == 'AltLeft' ){
6837         onKeyAlt = false
6838     }else
6839     if( keycode == 'KeyJ' ){ onKeyJ = false }else
6840     if( keycode == 'KeyK' ){ onKeyK = false }else
6841     if( keycode == 'KeyL' ){ onKeyL = false }else
6842     {
6843     }
6844     ev.preventDefault()
6845 }
6846 function and(a,b){ if(a){ if(b){ return true; } return false; } }
6847 function GJWin_OnKeyDown(ev){
6848     keycode = ev.code;
6849     mode = ''
6850     key = ''
6851     if( keycode == 'ControlLeft' ){
6852         onKeyControl = true
6853         ev.preventDefault()
6854         return;
6855     }else
6856     if( keycode == 'ShiftLeft' ){
6857         onKeyShift = true
6858         ev.preventDefault()
6859         return;
6860     }else
6861     if( keycode == 'AltLeft' ){
6862         ev.preventDefault()
6863         onKeyAlt = true
6864         return;
6865     }else
6866     if( keycode == 'Backquote' ){
6867         TransMode = !TransMode
6868         ev.preventDefault()
6869     }else
6870     if( and(keycode == 'Space', onKeyShift) ){
6871         TransMode = !TransMode
6872         ev.preventDefault()
6873     }else
6874     if( keycode == 'ShiftRight' ){
6875         TransMode = !TransMode
6876     }else
6877     if( keycode == 'Escape' ){
6878         TransMode = true
6879         ev.preventDefault()
6880     }else
6881     if( keycode == 'Enter' ){
6882         TransMode = false
6883         //ev.preventDefault()
6884     }
6885     if( keycode == 'KeyJ' ){ onKeyJ = true }else
6886     if( keycode == 'KeyK' ){ onKeyK = true }else
6887     if( keycode == 'KeyL' ){ onKeyL = true }else
6888     {
6889     }
6890
6891     if( ev.altKey ){ key += 'Alt+' }
6892     if( onKeyControl ){ key += 'Ctrl+' }
6893     if( onKeyShift ){ key += 'Shift+' }
6894     if( and(keycode != 'KeyJ', onKeyJ) ){ key += 'J+' }
6895     if( and(keycode != 'KeyK', onKeyK) ){ key += 'K+' }
6896     if( and(keycode != 'KeyL', onKeyL) ){ key += 'L+' }
6897     key += keycode
6898
6899     if( TransMode ){
6900         //mode = "[\343\201\202r]"
6901         mode = "[\343\201\202r]"
6902     }else{
6903         mode = '[---]'
6904     }
6905     ///// //gjmode.innerHTML = "[---]"
6906     GJWin_Mode.innerHTML = mode + ' ' + key
6907     //alert('Key:'+keycode)
6908     ev.stopPropagation()
6909     //ev.preventDefault()
6910 }
6911 function GJWin_OnScroll(ev){
6912     x = DragStartX = gsh.getBoundingClientRect().left.toFixed(0)
6913     y = DragStartY = gsh.getBoundingClientRect().top.toFixed(0)
6914     GJLog_append('OnScroll: x='+x+',y='+y)
6915 }
6916 document.addEventListener('scroll',GJWin_OnScroll)
6917 function GJWin_OnResize(ev){
6918     w = window.innerWidth
6919     h = window.innerHeight
6920     GJLog_append('OnResize: w='+w+',h='+h)
6921 }
6922 window.addEventListener('resize',GJWin_OnResize)
6923
6924 var DragStartX = 0
6925 var DragStartY = 0
6926 function GJWin_DragStart(ev){
6927     // maybe this is the grabbing position
6928     this.style.position = 'fixed'
6929     x = DragStartX = gsh.getBoundingClientRect().left.toFixed(0)
6930     y = DragStartY = this.getBoundingClientRect().top.toFixed(0)
6931     GJLog_Stat.value = 'DragStart: x='+x+',y='+y
6932 }
6933 function GJWin_Drag(ev){
6934     x = ev.clientX; y = ev.clientY // x = ev.pageX; y = ev.pageY
6935     this.style.left = x - DragStartX
6936     this.style.top = y - DragStartY
6937     this.style.zIndex = '30000'
6938     this.style.position = 'fixed'
6939     x = this.getBoundingClientRect().left.toFixed(0)
6940     y = this.getBoundingClientRect().top.toFixed(0)
6941     GJLog_Stat.value = 'x='+x+',y='+y
6942     ev.preventDefault()
6943     ev.stopPropagation()
6944 }

```



```

6945 function GJWin_DragEnd(ev){
6946     x = ev.clientX; y = ev.clientY
6947     //x = ev.pageX; y = ev.pageY
6948     this.style.left = x - DragStartX
6949     this.style.top = y - DragStartY
6950     this.style.zIndex = '30000'
6951     this.style.position = 'fixed'
6952     if( true ){
6953         console.log("Dropped: "+this.nodeName+'#'+this.id+' x='+x+' y='+y
6954             +' parent='+this.parentNode.id)
6955     }
6956     x = this.getBoundingClientRect().left.toFixed(0)
6957     y = this.getBoundingClientRect().top.toFixed(0)
6958     GJLog_Stat.value = 'x='+x+',y='+y
6959     ev.preventDefault()
6960     ev.stopPropagation()
6961 }
6962 function GJWin_DragIgnore(ev){
6963     ev.preventDefault()
6964     ev.stopPropagation()
6965 }
6966 // 2020-09-15 let every object have console view!
6967 var GJ_ConsoleID = 0
6968 var PrevReport = new Date()
6969 function GJLog_StatUpdate(){
6970     txa = GJLog_Stat;
6971     if( txa == null ){
6972         return;
6973     }
6974     tmLap0 = new Date();
6975     p = txa.parentNode;
6976     pw = txa.getBoundingClientRect().width;
6977     ph = txa.getBoundingClientRect().height;
6978     //txa.value += '#'+p.id+' pw='+pw+' ph='+ph+'\n';
6979     tx1 = '#'+p.id+' pw='+pw+' ph='+ph+'\n';
6980
6981     w = txa.getBoundingClientRect().width;
6982     h = txa.getBoundingClientRect().height;
6983     //txa.value += 'w='+w+', h='+h+'\n';
6984     tx1 += 'w='+w+', h='+h+'\n';
6985
6986     //txa.value += '\n';
6987     //txa.value += DateShort() + '\n';
6988     tx1 += '\n';
6989     tx1 += DateShort() + '\n';
6990     tmLap1 = new Date();
6991
6992     txa.value += tx1;
6993     tmLap2 = new Date();
6994
6995     // vertical centering of the last line
6996     sHeight = txa.scrollHeight - 30; // depends on the font-size
6997     tmLap3 = new Date();
6998
6999     txa.scrollTop = sHeight; // depends on the font-size
7000     tmLap4 = new Date();
7001
7002     now = tmLap0.getTime();
7003     if( PrevReport == 0 || 10000 <= now-PrevReport ){
7004         PrevReport = now;
7005         console.log('StatBarUpdate:
7006             + ' leng=' + txa.value.length + ' byte, '
7007             + ' time=' + (tmLap4 -tmLap0) + ' ms { '
7008             + ' tadd=' + (tmLap2 -tmLap1) + ', '
7009             + ' hcal=' + (tmLap3 -tmLap2) + ', '
7010             + ' scrl=' + (tmLap4 -tmLap3) + ' }'
7011         );
7012     }
7013 }
7014 GJWin_StatUpdate = GJLog_StatUpdate;
7015 function GJ_showTime1(wid){
7016     //e = document.getElementById(wid);
7017     //console.log(wid.id+' .value.length='+wid.value.length)
7018     if( e != null ){
7019         //e.value = DateShort();
7020     }else{
7021         // should remove the Listener
7022     }
7023 }
7024 function GJWin_OnResizeTextarea(ev){
7025     this.value += 'resized:' + '\n'
7026 }
7027 function GJ_NewConsole(wname){
7028     wid = wname + '_' + GJ_ConsoleID
7029     GJ_ConsoleID += 1
7030
7031     GJFactory.style.setProperty('width',360+'px'); //GJFSIZE
7032     GJFactory.style.setProperty('height',320+'px')
7033     e = GJFactory;
7034     console.log('GJFa #'+e.id+' from w='+e.style.width+', h='+e.style.height)
7035
7036     if( GJFactory.innerHTML == "" ){
7037         GJFactory.innerHTML = '<'+H3>GJ_Factory_' + GJ_FactoryID +'<'+/H3><'+hr>\n'
7038     }else{
7039         GJFactory.innerHTML += '<'+hr>\n'
7040     }
7041
7042     gjwin = GJLog_Win = document.createElement('span')
7043     gjwin.id = wid
7044     gjwin.setAttribute('class','GJWin')
7045     gjwin.setAttribute('draggable','true')
7046     gjwin.addEventListener('dragstart',GJWin_DragStart)
7047     gjwin.addEventListener('drag',GJWin_Drag)
7048     gjwin.addEventListener('dragend',GJWin_Drag)
7049     gjwin.addEventListener('dragover',GJWin_DragIgnore)
7050     gjwin.addEventListener('dragenter',GJWin_DragIgnore)
7051     gjwin.addEventListener('dragleave',GJWin_DragIgnore)
7052     gjwin.addEventListener('dragexit',GJWin_DragIgnore)
7053     gjwin.addEventListener('drop',GJWin_DragIgnore)
7054     gjwin.addEventListener('keydown',GJWin_OnKeyDown)
7055
7056     gjtab = GJLog_Tab = document.createElement('textarea')
7057     gjtab.addEventListener('keydown',GJWin_OnKeyDown)
7058     gjtab.style.readonly = true
7059     gjtab.contenteditable = false
7060     gjtab.value = wid
7061     gjtab.id = wid + '_Tab'
7062     gjtab.setAttribute('class','GJTab')
7063     gjtab.setAttribute('spellcheck','false')
7064     gjwin.appendChild(gjtab)
7065
7066     gjstat = GJLog_Stat = document.createElement('textarea')
7067     gjstat.addEventListener('keydown',GJWin_OnKeyDown)
7068     gjstat.id = wid + '_Stat'

```

```

7069 gjstat.value = DateShort()
7070 gjstat.setAttribute('class', 'GJStat')
7071 gjstat.setAttribute('spellcheck', 'false')
7072 gjwin.appendChild(gjstat)
7073
7074 gjicon = document.createElement('span')
7075 gjicon.addEventListener('keydown', GJWin_OnKeyDown)
7076 gjicon.id = wid + 'Icon'
7077 gjicon.innerHTML = '<G<font color="#f44">J</font>'
7078 gjicon.setAttribute('class', 'GJIcon')
7079 gjicon.setAttribute('spellcheck', 'false')
7080 gjwin.appendChild(gjicon)
7081
7082 gjtext = GJLog_Text = document.createElement('textarea')
7083 gjtext.addEventListener('keydown', GJWin_OnKeyDown)
7084 gjtext.addEventListener('keyup', GJWin_OnKeyUp)
7085 gjtext.addEventListener('resize', GJWin_OnResizeTextarea)
7086 gjtext.id = wid + '_Text'
7087 gjtext.setAttribute('class', 'GJText')
7088 gjtext.setAttribute('spellcheck', 'false')
7089 gjwin.appendChild(gjtext)
7090
7091
7092 // user's mode as of IME
7093 gjmode = GJWin_Mode = document.createElement('textarea')
7094 gjmode.addEventListener('keydown', GJWin_OnKeyDown)
7095 gjmode.addEventListener('keydown', GJWin_OnKeyDown)
7096 gjmode.id = wid + '_Mode'
7097 gjmode.setAttribute('class', 'GJMode')
7098 gjmode.setAttribute('spellcheck', 'false')
7099 gjmode.innerHTML = '[---]'
7100 gjwin.appendChild(gjmode)
7101
7102 gjwin.zIndex = 30000
7103 GJFactory.appendChild(gjwin)
7104
7105 gjtab.scrollTop = 0
7106 gjstat.scrollTop = 0
7107
7108 //x = gjwin.getBoundingClientRect().left.toFixed(0)
7109 //y = gjwin.getBoundingClientRect().top.toFixed(0)
7110 //gjwin.style.position = 'static'
7111 //gjwin.style.left = 0
7112 //gjwin.style.top = 0
7113
7114 //update = '{'+wid+'.value=DateShort()}',
7115 update = '{GJ_showTime1('+wid+')};',
7116 // 2020-09-19 this causes memory leaks
7117 //FProductInterval = window.setInterval(update,200)
7118 //FProductInterval = window.setInterval(GJWin_StatUpdate,200)
7119 //FProductInterval = window.setInterval(GJ_showTime1,200,wid);
7120 FProductInterval = window.setInterval(GJ_showTime1,200,gjstat);
7121 return update
7122 }
7123 function xxxGJF_StripeClass(){
7124 GJLog_Win.style.removeProperty('width')
7125 GJLog_Tab.style.removeProperty('width')
7126 GJLog_Stat.style.removeProperty('width')
7127 GJLog_Text.style.removeProperty('width')
7128 return "Stripped classes"
7129 }
7130 function isElem(id){
7131 return document.getElementById(id) != null
7132 }
7133 function GJLog_append(...args){
7134 txt = GJLog_Text;
7135 if( txt == null ){
7136 return; // maybe GJLog element is removed
7137 }
7138 logs = args.join(' ');
7139 txt.value += logs + '\n'
7140 txt.scrollTop = txt.scrollHeight
7141 //GJLog_Stat.value = DateShort()
7142 }
7143 //window.addEventListener('time', GJLog_StatUpdate)
7144 function test_GJ_Console(){
7145 window.setInterval(GJLog_StatUpdate,1000);
7146 GJ_NewConsole('GJ_Console')
7147 e = GJFactory;
7148 console.log('GJF0 #' + e.id + ' from w=' + e.style.width + ', h=' + e.style.height)
7149 e.style.width = 360; //GJFsize
7150 e.style.height = 320;
7151 console.log('GJF0 #' + e.id + ' to w=' + e.style.width + ', h=' + e.style.height)
7152 }
7153 /// test_GJ_Console();
7154
7155 var StopConsoleLog = true
7156 // 2020-09-15 added,
7157 // log should be saved to permanent memory
7158 // const px = new Proxy(console.log, { alert() })
7159 __console_log = console.log
7160 __console_info = console.info
7161 __console_warn = console.warn
7162 __console_error = console.error
7163 __console_exception = console.exception
7164 // should pop callstack info.
7165 console.exception = function(...args){
7166 __console_exception(...args)
7167 alert('-- got console.exception(""+args+"")')
7168 }
7169 console.error = function(...args){
7170 __console_error(...args)
7171 alert('-- got console.error(""+args+"")')
7172 }
7173 console.warn = function(...args){
7174 __console_warn(...args)
7175 alert('-- got console.warn(""+args+"")')
7176 }
7177 console.info = function(...args){
7178 alert('-- got console.info(""+args+"")')
7179 __console_info(...args)
7180 }
7181 console.xxxlog = function(...args){ // rewrite xxxlog to log to enable it
7182 __console_log(...args)
7183 if( StopConsoleLog ){
7184 return;
7185 }
7186 if( 0 <= args[0].indexOf('!') ){
7187 //alert('-- got console.log(""+args+"")')
7188 }
7189 GJLog_append(...args)
7190 }
7191
7192 //document.getElementById('GshFaviconURL').href = GShellFavicon

```

```

7193 //document.getElementById('GshFaviconURL').href = GShellInsideIcon
7194 //document.getElementById('GshFaviconURL').href = ITSmoreQR
7195 //document.getElementById('GshFaviconURL').href = GSellLogo
7196
7197 // id of GShell HTML elemets
7198 var E_BANNER = "GshBanner" // banner element in HTML
7199 var E_FOOTER = "GshFooter" // footer element in HTML
7200 var E_GINDEX = "gsh-gindex" // index of Golang code of GShell
7201 var E_GOCODE = "gsh-gocode" // Golang code of GShell
7202 var E_TODO = "gsh-todo" // TODO of GShell
7203 var E_DICT = "gsh-dict" // Dictionary of GShell
7204
7205 function bannerElem(){ return document.getElementById(E_BANNER); }
7206 function bannerStyleFunc(){ return bannerElem().style; }
7207 var bannerStyle = bannerStyleFunc()
7208 function GshSetImages(){
7209     document.getElementById('GshFaviconURL').href = GShellInsideIcon
7210     bannerStyle.backgroundImage = "url("+GSellLogo+")";
7211     //bannerStyle.backgroundImage = "url("+GShellInsideIcon+")";
7212     //bannerStyle.backgroundImage = "url("+GShellFavicon+")";
7213     GMenu.style.backgroundImage = "url("+GShellInsideIcon+")";
7214     showFooter();
7215 }
7216
7217 function footerElem(){ return document.getElementById(E_FOOTER); }
7218 function footerStyle(){ return footerElem().style; }
7219 //footerElem().style.backgroundImage="url("+ITSmoreQR+")";
7220 //footerStyle().backgroundImage = "url("+ITSmoreQR+")";
7221
7222 function html_fold(e){
7223     if( e.innerHTML == "Fold" ){
7224         e.innerHTML = "Unfold"
7225         document.getElementById('gsh-menu-exit').innerHTML=""
7226         document.getElementById('GshStatement').open=false
7227         GshFeatures.open = false
7228         document.getElementById('html-src').open=false
7229         document.getElementById(E_GINDEX).open=false
7230         document.getElementById(E_GOCODE).open=false
7231         document.getElementById(E_TODO).open=false
7232         document.getElementById('references').open=false
7233     }else{
7234         e.innerHTML = "Fold"
7235         document.getElementById('GshStatement').open=true
7236         GshFeatures.open = true
7237         document.getElementById(E_GINDEX).open=true
7238         document.getElementById(E_GOCODE).open=true
7239         document.getElementById(E_TODO).open=true
7240         document.getElementById('references').open=true
7241     }
7242 }
7243 function html_pure(e){
7244     if( e.innerHTML == "Pure" ){
7245         document.getElementById('gsh').style.display=true
7246         //document.style.display = false
7247         e.innerHTML = "Unpure"
7248     }else{
7249         document.getElementById('gsh').style.display=false
7250         //document.style.display = true
7251         e.innerHTML = "Pure"
7252     }
7253 }
7254
7255 var bannerIsStopping = false
7256 //NOTE: .com/JSREF/prop_style_backgroundposition.asp
7257 function shiftBG(){
7258     bannerIsStopping = !bannerIsStopping
7259     bannerStyle.backgroundPosition = "0 0";
7260 }
7261 // status should be inherited on Window Fork(), so use the status in DOM
7262 function html_stop(e,toggle){
7263     if( toggle ){
7264         if( e.innerHTML == "Stop" ){
7265             bannerIsStopping = true
7266             e.innerHTML = "Start"
7267         }else{
7268             bannerIsStopping = false
7269             e.innerHTML = "Stop"
7270         }
7271     }else{
7272         // update JavaScript variable from DOM status
7273         if( e.innerHTML == "Stop" ){ // shown if it's running
7274             bannerIsStopping = false
7275         }else{
7276             bannerIsStopping = true
7277         }
7278     }
7279 }
7280 html_stop(document.getElementById('GshMenuStop'),false) // onInit.
7281 //html_stop(bannerElem(),false) // onInit.
7282
7283 //https://www.w3schools.com/jsref/met_win_setinterval.asp
7284 function shiftBanner(){
7285     var now = new Date().getTime();
7286     //console.log("now="+now%10)
7287     if( !bannerIsStopping ){
7288         bannerStyle.backgroundPosition = ((now/10)%100000)+" 0";
7289     }
7290 }
7291 function Banner_init(){
7292     window.setInterval(shiftBanner,10); // onInit.
7293 }
7294
7295 // <a href="https://developer.mozilla.org/ja/docs/Web/API/Window/open">window.open()</a>
7296 // from embedded html to standalone page
7297 var MyChildren = 0
7298 function html_fork(){
7299     ResetPerFon();
7300     ResetAffView();
7301     Reset_ShadingCanvas();
7302     GJFactory_Destroy();
7303     MyChildren += 1
7304     WinId = document.getElementById('gsh-WinId').innerHTML + "." + MyChildren;
7305     newwin = window.open("",WinId,"");
7306     src = document.getElementById("gsh");
7307     srchtml = src.outerHTML
7308     newwin.document.write("/*<<"+html>\n");
7309     newwin.document.write(srchtml);
7310     newwin.document.write("<"+"/html>\n");
7311     newwin.document.getElementById('gsh-menu-exit').innerHTML = "Close";
7312     newwin.document.getElementById('gsh-WinId').innerHTML = WinId;
7313     newwin.document.close();
7314     newwin.focus();
7315 }
7316 function html_close(){

```

```

7317     window.close()
7318 }
7319 function win_jump(win){
7320 //win = window.top;
7321 win = window.opener; // https://developer.mozilla.org/ja/docs/Web/API/window.opener
7322 if( win == null ){
7323     console.log("jump to window.opener("+win+") (Error)\n")
7324 }else{
7325     console.log("jump to window.opener("+win+")\n")
7326     win.focus();
7327 }
7328 }
7329
7330 // 0.2.9 2020-0902 created chekosum of HTML
7331 CRC32UNIX = 0x04C11DB7 // Unix cksun
7332 function byteCRC32add(bigcrc,octstr,octlen){
7333     var crc = new Uint32Array(1)
7334     crc[0] = bigcrc
7335
7336     let oi = 0
7337     for( ; oi < octlen; oi++){
7338         var oct = new Uint8Array(1)
7339         oct[0] = octstr[oi]
7340         for( bi = 0; bi < 8; bi++){
7341             //console.log("--CRC32 "+crc[0]+" "+oct[0].toString(16)+" ["+oi+"."+bi+"]\n")
7342             ovf1 = crc[0] < 0 ? 1 : 0
7343             ovf2 = oct[0] < 0 ? 1 : 0
7344             ovf = ovf1 ^ ovf2
7345             oct[0] <= 1
7346             crc[0] <= 1
7347             if( ovf ){ crc[0] ^= CRC32UNIX }
7348         }
7349     }
7350     //console.log("--CRC32 byteAdd return crc="+crc[0]+" "+oi+"/"+"octlen+"\n")
7351     return crc[0];
7352 }
7353 function strCRC32add(bigcrc,stri,strlen){
7354     var crc = new Uint32Array(1)
7355     crc[0] = bigcrc
7356     var code = new Uint8Array(strlen);
7357     for( i = 0; i < strlen; i++){
7358         code[i] = stri.charCodeAtAt(i) // not charAt() !!!!
7359         //console.log("=== "+code[i].toString(16)+" <<== "+stri[i+"\n")
7360     }
7361     crc[0] = byteCRC32add(crc,code,strlen)
7362     //console.log("--CRC32 strAdd return crc="+crc[0]+"+"\n")
7363     return crc[0]
7364 }
7365 function byteCRC32end(bigcrc,len){
7366     var crc = new Uint32Array(1)
7367     crc[0] = bigcrc
7368     var slen = new Uint8Array(4)
7369     let li = 0
7370     for( ; li < 4; ){
7371         slen[li] = len
7372         li += 1
7373         len >= 8
7374         if( len == 0 ){
7375             break
7376         }
7377     }
7378     crc[0] = byteCRC32add(crc[0],slen,li)
7379     crc[0] ^= 0xFFFFFFFF
7380     return crc[0]
7381 }
7382 function strCRC32(stri,len){
7383     var crc = new Uint32Array(1)
7384     crc[0] = 0
7385     crc[0] = strCRC32add(0,stri,len)
7386     crc[0] = byteCRC32end(crc[0],len)
7387     //console.log("--CRC32 "+crc[0]+" "+len+"\n")
7388     return crc[0]
7389 }
7390
7391 DestroyGJLink = null; // to be replaced
7392 DestroyFooter = null; // to be defined
7393 DestroyEventSharingCodeview = function dummy(){
7394 DestroyWirtualDesktop = function(){
7395 DestroyNaviButtons = function(){
7396
7397 function getSourceText(){
7398     if( DestroyFooter != null ) DestroyFooter();
7399     version = document.getElementById('GshVersion').innerHTML
7400     sfavico = document.getElementById('GshFaviconURL').href;
7401     sbanner = document.getElementById('GshBanner').style.backgroundImage;
7402     spositi = document.getElementById('GshBanner').style.backgroundPosition;
7403
7404     if( document.getElementById('GJC_1') != null ){ GJC_1.remove() }
7405     if( DestroyGJLink != null ) DestroyGJLink();
7406     DestroyEventSharingCodeview();
7407     DestroyWirtualDesktop();
7408     GshTopbar.innerHTML = "";
7409     DestroyIndexBar();
7410     DestroyNaviButtons();
7411     ResetPerfMon();
7412     ResetAffView();
7413     Reset_ShadingCanvas();
7414
7415     // these should be removed by CSS selector or class, after sevaed to non-printed attribute
7416     GshBanner.removeAttribute('style');
7417     document.getElementById('GshMenuSign').removeAttribute("style");
7418     styleGMenu = GMenu.getAttribute("style")
7419     GMenu.removeAttribute("style");
7420     styleGStat = GStat.getAttribute("style")
7421     GStat.removeAttribute("style");
7422     styleGTop = GTop.getAttribute("style")
7423     GTop.removeAttribute("style");
7424     styleGshGrid = GshGrid.getAttribute("style")
7425     GshGrid.removeAttribute("style");
7426     //styleGPos = GPos.getAttribute("style");
7427     //GPos.removeAttribute("style");
7428     //GPos.innerHTML = "";
7429     //styleGLog = GLog.getAttribute("style");
7430     //GLog.removeAttribute("style");
7431     //GLog.innerHTML = "";
7432     styleGShellPlane = GShellPlane.getAttribute("style")
7433     GShellPlane.removeAttribute("style")
7434     styleRawTextViewer = RawTextViewer.getAttribute("style")
7435     RawTextViewer.removeAttribute("style")
7436     styleRawTextViewerClose = RawTextViewerClose.getAttribute("style")
7437     RawTextViewerClose.removeAttribute("style")
7438
7439     GshFaviconURL.href = "";
7440     if( iselem('ConfigIcon') ) ConfigIcon.src = "";

```

```

7441
7442 //it seems that interHTML and outerHTML generate style="" for these (??)
7443 //GshBanner.removeAttribute('style');
7444 //GshFooter.removeAttribute('style');
7445 //GshMenuSign.removeAttribute('style');
7446 GshBanner.style=""
7447 GshMenuSign.style=""
7448
7449 textarea = document.createElement("textarea")
7450 srchtml = document.getElementById("gsh").outerHTML;
7451 //textarea = document.createElement("textarea")
7452 // 2020-0910 ?? ... this causes inserting style="" to Banner and Footer,
7453 // with Chromium?/ after reloading from file:///
7454 textarea.innerHTML = srchtml
7455 // <a href="https://stackoverflow.com/questions/5796718/html-entity-decode">Thanks</a>
7456 var rawtext = textarea.value
7457 //textarea.destroy()
7458 //rawtext = gsh.textContent // this removes #include <FILENAME> too
7459 var orgtext = ""
7460 + "/*<+>html>\n" // lost preamble text
7461 + rawtext
7462 + "<+>/html>\n" // lost trail text
7463 ;
7464
7465 tlen = orgtext.length
7466 //console.log("getSourceText: length="+tlen+"\n")
7467 document.getElementById('GshFaviconURL').href = sfavico;
7468
7469 document.getElementById('GshBanner').style.backgroundImage = sbanner;
7470 document.getElementById('GshBanner').style.backgroundPosition = spositi;
7471
7472 GStat.setAttribute("style",styleGStat)
7473 GMenu.setAttribute("style",styleGMenu)
7474 GTop.setAttribute("style",styleGTop)
7475 //GLog.setAttribute("style",styleGLog)
7476 //GPos.setAttribute("style",styleGPos)
7477 GshGrid.setAttribute("style",styleGshGrid)
7478 GShellPlane.setAttribute("style",styleGshellPlane)
7479 RawTextViewer.setAttribute("style",styleRawTextViewer)
7480 RawTextViewerClose.setAttribute("style",styleRawTextViewerClose)
7481 canontext = orgtext.replace(' style=""','')
7482 // open="" too
7483 return canontext
7484 }
7485 function getDigest(){
7486 var text = ""
7487 text = getSourceText()
7488 var digest = ""
7489 tlen = text.length
7490 digest = strCRC32(text,tlen) + " " + tlen
7491 return { text, digest }
7492 }
7493 function html_digest(){
7494 version = document.getElementById('GshVersion').innerHTML
7495 let {text, digest} = getDigest()
7496 alert("cksum: " + digest + " " + version)
7497 }
7498 function charsin(stri,char){
7499 ln = 0;
7500 for( i = 0; i < stri.length; i++){
7501 if( stri.charCodeAt(i) == char.charCodeAt(0) )
7502 ln++;
7503 }
7504 return ln;
7505 }
7506
7507 //class digestElement extends HTMLElement { }
7508 //< script>customElements.define('digest',digestElement)< /script>
7509 function showDigest(e){
7510 result = 'version=' + GshVersion.innerHTML + '\n'
7511 result += 'lines=' + e.dataset.lines + '\n'
7512 + 'length=' + e.dataset.length + '\n'
7513 + 'crc32u=' + e.dataset.crc32u + '\n'
7514 + 'time=' + e.dataset.time + '\n';
7515
7516 alert(result)
7517 }
7518
7519 function html_sign(e){
7520 if( RawTextViewer.style.zIndex == 1000 ){
7521 hideRawTextViewer()
7522 return
7523 }
7524 GshTopbar.innerHTML = "";
7525 ResetPerfMon();
7526 ResetAffView();
7527 Reset_ShadingCanvas();
7528 DestroyIndexBar();
7529 DestroyNavButtons();
7530 DestroyEventSharingCodeview();
7531 Destroy_VirtualDesktop();
7532 GJFactory.Destroy();
7533 if( DestroyGJLink != null ) DestroyGJLink();
7534 //gsh_digest_innerHTML = "";
7535 text = getSourceText() // the original text
7536 tlen = text.length
7537 digest = strCRC32(text,tlen)
7538 //gsh_digest_innerHTML = digest + " " + tlen
7539 //text = getSourceText() // the text with its digest
7540 Lines = charsin(text,'\n')
7541
7542 name = "gsh"
7543 sid = name + "--digest"
7544 d = new Date()
7545 signedAt = d.getTime()
7546
7547 sign = '/'+'*<+>span\n'
7548 + ' id="'+ sid + '"\n'
7549 + ' class=" digest "\n'
7550 + ' data-target-id="'+name+"\n'
7551 + ' data-crc32u="'+ digest + '"\n'
7552 + ' data-length="'+ tlen + '"\n'
7553 + ' data-lines="'+ Lines + '"\n'
7554 + ' data-time="'+ signedAt + '"\n'
7555 + ' ><+> /span>\n'+'\n'
7556
7557 text = sign + text
7558
7559 txhtml = '<+> table id="LineNumbered"><+> <tr><+> <td>'
7560 + '<+> <+> textarea cols=5 rows=' + Lines + ' class="LineNumber">'
7561 for( i = 1; i <= Lines; i++){
7562 txhtml += i.toString() + '\n'
7563 }
7564 txhtml += ""

```

```

7565     + '<' + '/textarea>'
7566     + '<' + '/td<' + 'td>'
7567     + '<' + 'textarea cols=150 rows=' + Lines + 'spellcheck="false"'
7568     + ' class="LineNumbers">'
7569     + text + '<' + '/textarea>'
7570     + '<' + '/td<' + '/tr<' + '/table>'
7571
7572     for( i = 1; i <= 30; i++ ){
7573         txhtml += '<br>\n'
7574     }
7575     RawTextViewer.innerHTML = txhtml
7576     RawTextViewer.spellcheck = false // (spellcheck above seems ineffective)
7577
7578     btn = e
7579     e.style.color = "rgba(128,128,255,0.9)";
7580     y = e.getBoundingClientRect().top.toFixed(0)
7581     //h = e.getBoundingClientRect().height.toFixed(0)
7582     RawTextViewer.style.top = Number(y) + 30
7583     RawTextViewer.style.left = 100;
7584     RawTextViewer.style.height = window.innerHeight - 20;
7585     //RawTextViewer.style.Opacity = 1.0;
7586     //RawTextViewer.style.backgroundColor = "rgba(0,0,0,0.0)";
7587     RawTextViewer.style.backgroundColor = "rgba(255,255,255,0.8)";
7588     RawTextViewer.style.zIndex = 1000;
7589     RawTextViewer.style.display = true;
7590
7591     if( RawTextViewerClose.style == null ){
7592         RawTextViewerClose.style = "";
7593     }
7594     RawTextViewerClose.style.top = Number(y) + 10
7595     RawTextViewerClose.style.left = 100;
7596     RawTextViewerClose.style.zIndex = 1001;
7597
7598     ScrollToElement(CurElement,RawTextViewerClose)
7599 }
7600 function hideRawTextViewer(){
7601     RawTextViewer.style.left = 10000;
7602     RawTextViewer.style.zIndex = -100;
7603     RawTextViewer.style.Opacity = 0.0;
7604     RawTextViewer.style = null
7605     RawTextViewer.innerHTML = "";
7606
7607     GshMenuSign.style.color = "rgba(255,128,128,1.0)";
7608     RawTextViewerClose.style.top = 0;
7609     RawTextViewerClose.style = null
7610 }
7611
7612 // source code view
7613 function frame_close(){
7614     srcframe = document.getElementById("src-frame");
7615     srcframe.innterHTML = "";
7616     //srcframe.style.cols = 1;
7617     srcframe.style.rows = 1;
7618     srcframe.style.height = 0;
7619     srcframe.style.display = false;
7620     src = document.getElementById("SrcTextarea");
7621     src.innerHTML = ""
7622     //src.cols = 0
7623     src.rows = 0
7624     src.display = false
7625     //alert("--closed--")
7626 }
7627 //<!-- | <span onclick="html_view();">Source</span> -->
7628 //<!-- | <span onclick="frame_close();">SourceClose</span> -->
7629 //<!--| <span>Download</span> -->
7630 function frame_open(){
7631     GshTopbar.innerHTML = "";
7632     ResetPerfMon();
7633     ResetAffView();
7634     Reset_ShadingCanvas();
7635     DestroyIndexBar();
7636     DestroyNavButtons();
7637     if( DestroyFooter != null ) DestroyFooter();
7638     document.getElementById('GshFaviconURL').href = "";
7639     oldsrc = document.getElementById("GENSRC");
7640     if( oldsrc != null ){
7641         //alert("--I--(erasing old text)")
7642         oldsrc.innterHTML = "";
7643         return
7644     }else{
7645         //alert("--I--(no old text)")
7646     }
7647     styleBanner = GshBanner.getAttribute("style")
7648     GshBanner.removeAttribute("style")
7649     if( document.getElementById('GJC_1') ){ GJC_1.remove() }
7650
7651     GshFaviconURL.href = "";
7652     if( iselem('ConfigIcon') ) ConfigIcon.src = "";
7653     GStat.removeAttribute('style')
7654     GshGrid.removeAttribute('style')
7655     GshMenuSign.removeAttribute('style')
7656     //GPos.removeAttribute('style')
7657     //GPos.innerHTML = "";
7658     //GLog.removeAttribute('style')
7659     //GLog.innerHTML = "";
7660     GMenu.removeAttribute('style')
7661     GTop.removeAttribute('style')
7662     GShellPlane.removeAttribute('style')
7663     RawTextViewer.removeAttribute('style')
7664     RawTextViewerClose.removeAttribute('style')
7665
7666     if( DestroyGJLink != null ) DestroyGJLink();
7667     GJFactory_Destroy()
7668     Destroy_WirtualDesktop();
7669     DestroyEventSharingCodeview();
7670
7671     src = document.getElementById("gsh");
7672     srhtml = src.outerHTML
7673     srcframe = document.getElementById("src-frame");
7674     srcframe.innerHTML = ""
7675     + "<"+<cite id='\<GENSRC'\>\n"
7676     + "<"+<style>\n"
7677     + "#GENSRC textarea{tab-size:4;}\n"
7678     + "#GENSRC textarea{-o-tab-size:4;}\n"
7679     + "#GENSRC textarea{-moz-tab-size:4;}\n"
7680     + "#GENSRC textarea{spellcheck:false;}\n"
7681     + "</"+<style>\n"
7682     + "<"+<textarea id="SrcTextarea" cols=100 rows=20 class="gsh-code" spellcheck="false">"
7683     + "/*"+<html>\n" // lost preamble text
7684     + srhtml
7685     + "<"+</html>\n" // lost trail text
7686     + "</"+<textarea>\n"
7687     + "</"+<cite><!-- GENSRC -->\n";
7688

```

```

7689 //srcframe.style.cols = 80;
7690 //srcframe.style.rows = 80;
7691
7692 GshBanner.setAttribute('style',styleBanner)
7693 }
7694 function fill_CSSView(){
7695     part = document.getElementById('GshStyleDef')
7696     view = document.getElementById('gsh-style-view')
7697     view.innerHTML = ""
7698     + "<"+'textarea cols=100 rows=20 class="gsh-code">'
7699     + part.innerHTML
7700     + "<"+'/textarea>"
7701 }
7702 function fill_JavaScriptView(){
7703     jspart = document.getElementById('gsh-script')
7704     view = document.getElementById('gsh-script-view')
7705     view.innerHTML = ""
7706     + "<"+'textarea cols=100 rows=20 class="gsh-code">'
7707     + jspart.innerHTML
7708     + "<"+'/textarea>"
7709 }
7710 function fill_DataView(){
7711     part = document.getElementById('gsh-data')
7712     view = document.getElementById('gsh-data-view')
7713     view.innerHTML = ""
7714     + "<"+'textarea cols=100 rows=20 class="gsh-code">'
7715     + part.innerHTML
7716     + "<"+'/textarea>"
7717 }
7718 function jumpto_StyleView(){
7719     jsview = document.getElementById('html-src')
7720     jsview.open = true
7721     jsview = document.getElementById('gsh-style-frame')
7722     jsview.open = true
7723     fill_CSSView()
7724 }
7725 function jumpto_JavaScriptView(){
7726     jsview = document.getElementById('html-src')
7727     jsview.open = true
7728     jsview = document.getElementById('gsh-script-frame')
7729     jsview.open = true
7730     fill_JavaScriptView()
7731 }
7732 function jumpto_DataView(){
7733     jsview = document.getElementById('html-src')
7734     jsview.open = true
7735     jsview = document.getElementById('gsh-data-frame')
7736     jsview.open = true
7737     fill_DataView()
7738 }
7739 function jumpto_WholeView(){
7740     jsview = document.getElementById('html-src')
7741     jsview.open = true
7742     jsview = document.getElementById('gsh-whole-view')
7743     jsview.open = true
7744     frame_open()
7745 }
7746 function html_view(){
7747     html_stop();
7748
7749     banner = document.getElementById('GshBanner').style.backgroundImage;
7750     footer = document.getElementById('GshFooter').style.backgroundImage;
7751     document.getElementById('GshBanner').style.backgroundImage = "";
7752     document.getElementById('GshBanner').style.backgroundPosition = "";
7753     document.getElementById('GshFooter').style.backgroundImage = "";
7754
7755     //srcwin = window.open("", "CodeView2", "");
7756     srcwin = window.open("", "", "");
7757     srcwin.document.write("<span id=\"gsh\">\n");
7758
7759     src = document.getElementById("gsh");
7760     srcwin.document.write("<"+'style>\n");
7761     srcwin.document.write("<"+'textarea{tab-size:4;}\n");
7762     srcwin.document.write("<"+'textarea(-o-tab-size:4;)\n");
7763     srcwin.document.write("<"+'textarea(-moz-tab-size:4;)\n");
7764     srcwin.document.write("<"+'style>\n");
7765     srcwin.document.write("<"+'h2>\n");
7766     srcwin.document.write("<"+'span onclick=\"window.close();\n>Close</span> | \n");
7767     //srcwin.document.write("<"+'span onclick=\"html_stop();\n>Run</span>\n");
7768     srcwin.document.write("<"+'h2>\n");
7769     srcwin.document.write("<"+'textarea id=\"gsh-src-src\" cols=100 rows=60>\n");
7770     srcwin.document.write("<"+'html>\n");
7771     srcwin.document.write("<"+'span id=\"gsh\">");
7772     srcwin.document.write(src.innerHTML);
7773     srcwin.document.write("<"+'span><"+'html>\n");
7774     srcwin.document.write("<"+'textarea>\n");
7775
7776     document.getElementById('GshBanner').style.backgroundImage = banner;
7777     document.getElementById('GshFooter').style.backgroundImage = footer
7778
7779     sty = document.getElementById("GshStyleDef");
7780     srcwin.document.write("<"+'style>\n");
7781     srcwin.document.write(sty.innerHTML);
7782     srcwin.document.write("<"+'style>\n");
7783
7784     run = document.getElementById("gsh-script");
7785     srcwin.document.write("<"+'script>\n");
7786     srcwin.document.write(run.innerHTML);
7787     srcwin.document.write("<"+'script>\n");
7788
7789     srcwin.document.write("<"+'span><"+'html>\n"); // gsh span
7790     srcwin.document.close();
7791     srcwin.focus();
7792 }
7793 GSH = document.getElementById("gsh")
7794
7795 //GSH.onclick = "alert('Ouch!')"
7796 //GSH.css = "{background-color:#eef;}"
7797 //GSH.style = "background-color:#eef;"
7798 //GSH.style.display = false;
7799 //alert('Ouch0!')
7800 //GSH.style.display = true;
7801
7802 // 2020-0904 created, tentative
7803 //document.addEventListener('keydown',jgshCommand);
7804 //CurElement = GshStatement
7805 CurElement = GshMenu
7806 MemElement = GshMenu
7807
7808 function nextSib(e){
7809     n = e.nextSibling;
7810     for( i = 0; i < 100; i++ ){
7811         if( n == null ){
7812             break;

```

```

7813     }
7814     if( n.nodeName == "DETAILS" ){
7815         return n;
7816     }
7817     n = n.nextSibling;
7818 }
7819 return null;
7820 }
7821 function prevSib(e){
7822     n = e.previousSibling;
7823     for( i = 0; i < 100; i++ ){
7824         if( n == null ){
7825             break;
7826         }
7827         if( n.nodeName == "DETAILS" ){
7828             return n;
7829         }
7830         n = n.previousSibling;
7831     }
7832     return null;
7833 }
7834 function setColor(e,eName,eColor){
7835     if( e.hasChildNodes() ){
7836         s = e.childNodes;
7837         if( s != null ){
7838             for( ci = 0; ci < s.length; ci++ ){
7839                 if( s[ci].nodeName == eName ){
7840                     s[ci].style.color = eColor;
7841                     //s[ci].style.backgroundColor = eColor;
7842                     break;
7843                 }
7844             }
7845         }
7846     }
7847 }
7848 // https://docs.microsoft.com/en-us/previous-versions//hh781509(v=vs.85)
7849 function showCurElementPosition(ev){
7850     // if( document.getElementById("GPos") == null ){
7851         return;
7852     // }
7853     // if( GPos == null ){
7854         return;
7855     // }
7856     e = CurElement
7857     y = e.getBoundingClientRect().top.toFixed(0)
7858     x = e.getBoundingClientRect().left.toFixed(0)
7859
7860     h = ev + " "
7861     h += 'y='+y+", "+ 'x='+x+" -- "
7862     h += "w=" + window.innerWidth + ", h=" + window.innerHeight + " -- "
7863     //GPos.test = h
7864     //GPos.innerHTML = h
7865     //GPos.innerHTML = h
7866 }
7867 }
7868
7869 function zero2(n){
7870     if( n < 10 ){
7871         return '0' + n;
7872     }else{
7873         return n;
7874     }
7875 }
7876 function DateHourMin(){
7877     d = new Date();
7878     //return '%02d:%02d'.sprintf(d.getHours(),d.getMinutes())
7879     return zero2(d.getHours()) + ":" + zero2(d.getMinutes());
7880 }
7881 function DateShort0(d){
7882     return d.getFullYear()
7883         + '/' + zero2(d.getMonth())
7884         + '/' + zero2(d.getDate())
7885         + ' ' + zero2(d.getHours())
7886         + ':' + zero2(d.getMinutes())
7887         + ':' + zero2(d.getSeconds())
7888 }
7889 function DateShort(){
7890     return DateShort0(new Date());
7891 }
7892 function DateLong0(ms){
7893     d = new Date();
7894     d.setTime(ms);
7895     return DateShort0(d)
7896         + '.' + d.getMilliseconds()
7897         + ' ' + d.getTimezoneOffset()/60
7898         + ' ' + d.getTime() + '.' + d.getMilliseconds()
7899 }
7900 function DateLong(){
7901     return DateLong0(new Date());
7902 }
7903 function GShellMenu(e){
7904     //GLog.innerHTML = "Hello, World! (" + DateLong() + ")"
7905     //showGShellPlane()
7906     ConfigClick();
7907 }
7908 // placements of planes
7909 function GShellResizeX(ev){
7910     //if( document.getElementById("GMenu") != null ){
7911         GMenu.style.left = window.innerWidth - 100
7912         GMenu.style.top = window.innerHeight - 90 - 200
7913         //console.log("place GMENU "+GMenu.style.left+" "+GMenu.style.top)
7914     //}
7915     //}
7916     GStat.style.width = window.innerWidth
7917     //if( document.getElementById("GPos") != null ){
7918         //GPos.style.width = window.innerWidth
7919         //GPos.style.top = window.innerHeight - 30; //GPos.style.height
7920     //}
7921     //if( document.getElementById("GLog") != null ){
7922         //GLog.style.width = window.innerWidth
7923         //GLog.innerHTML = ""
7924     //}
7925     //if( document.getElementById("GLog") != null ){
7926         //GLog.innerHTML = "Resize: w=" + window.innerWidth +
7927         //", h=" + window.innerHeight
7928     //}
7929     showCurElementPosition(ev)
7930 }
7931 function GShellResize(){
7932     GShellResizeX("RESIZE")
7933 }
7934 window.onresize = GShellResize
7935 var prevNode = null
7936 var LogMouseMoveOverElement = false;

```



```

7937 function GJSH_OnMouseMove(ev){
7938   if( LogMouseMoveOverElement == false ){
7939     return;
7940   }
7941   x = ev.clientX
7942   y = ev.clientY
7943   d = new Date()
7944   t = d.getTime() / 1000
7945   if( document.elementFromPoint ){
7946     e = document.elementFromPoint(x,y)
7947     if( e != null ){
7948       if( e == prevNode ){
7949         }else{
7950           console.log('Mo-'+'t'+(''+x+', ''+y+' '
7951             +e.nodeType+' '+e.tagName+'#'+e.id)
7952           prevNode = e
7953         }
7954       }else{
7955         console.log(t+'(''+x+', ''+y+'') no element')
7956       }
7957     }else{
7958       console.log(t+'(''+x+', ''+y+'') no elementFromPoint')
7959     }
7960   }
7961   window.addEventListener('mousemove',GJSH_OnMouseMove);
7962 }
7963 function GJSH_OnMouseMoveScreen(ev){
7964   x = ev.screenX
7965   y = ev.screenY
7966   d = new Date()
7967   t = d.getTime() / 1000
7968   console.log(t+'(''+x+', ''+y+'') no elementFromPoint')
7969 }
7970 //screen.addEventListener('mousemove',GJSH_OnMouseMoveScreen);
7971 }
7972 function ScrollToElement(oe,ne){
7973   ne.scrollIntoView()
7974   ny = ne.getBoundingClientRect().top.toFixed(0)
7975   nx = ne.getBoundingClientRect().left.toFixed(0)
7976   //GLog.innerHTML = "["+ny+", "+nx+"]"
7977   //window.scrollTo(0,0)
7978 }
7979 GTop.style.backgroundColor = "rgba(0,0,0,0.0)"
7980 GshGrid.style.left = '250px';
7981 GshGrid.style.zIndex = 0
7982 if( false ){
7983   oy = oe.getBoundingClientRect().top.toFixed(0)
7984   ox = oe.getBoundingClientRect().left.toFixed(0)
7985   y = e.getBoundingClientRect().top.toFixed(0)
7986   x = e.getBoundingClientRect().left.toFixed(0)
7987   window.scrollTo(x,y)
7988   ny = e.getBoundingClientRect().top.toFixed(0)
7989   nx = e.getBoundingClientRect().left.toFixed(0)
7990   //GLog.innerHTML = "["+oy+", "+ox+"]->["+y+", "+x+"]->["+ny+", "+nx+"]"
7991 }
7992 }
7993 function showGShellPlane(){
7994   if( GShellPlane.style.zIndex == 0 ){
7995     GShellPlane.style.zIndex = 1000;
7996     GShellPlane.style.left = 30;
7997     GShellPlane.style.height = 320;
7998     GShellPlane.innerHTML = DateLong() + "<br>" +
7999       "-- History --<br>" + MyHistory;
8000   }else{
8001     GShellPlane.style.zIndex = 0;
8002     GShellPlane.style.left = 0;
8003     GShellPlane.style.height = 50;
8004     GShellPlane.innerHTML = "";
8005   }
8006 }
8007 var SuppressGJShell = false
8008 function jgshCommand(kevent){
8009   if( SuppressGJShell ){
8010     return
8011   }
8012   key = kevent
8013   keycode = key.code
8014   //GStat.style.width = window.innerWidth
8015   GStat.style.backgroundColor = "rgba(0,0,0,0.4)"
8016 }
8017 console.log("JSGsh-Key:"+keycode+"(^-)/")
8018 if( keycode == "Slash" ){
8019   console.log('(''+x+', ''+y+' ' )
8020   e = document.elementFromPoint(x,y)
8021   console.log('(''+x+', ''+y+' ' +e.nodeType+' '+e.tagName+'#'+e.id)
8022 }else
8023 if( keycode == "Digit0" ){ // fold side-bar
8024   // "Zero page"
8025   showGShellPlane();
8026 }else
8027 if( keycode == "Digit1" ){ // fold side-bar
8028   primary.style.width = "94%"
8029   secondary.style.width = "0%"
8030   secondary.style.opacity = 0
8031   GStat.innerHTML = "[Single Column View]"
8032 }else
8033 if( keycode == "Digit2" ){ // unfold side-bar
8034   primary.style.width = "58%"
8035   secondary.style.width = "36%"
8036   secondary.style.opacity = 1
8037   GStat.innerHTML = "[Double Column View]"
8038 }else
8039 if( keycode == "KeyU" ){ // fold/unfold all
8040   html_fold(GshMenuFold);
8041   location.href = "#"+CurElement.id;
8042 }else
8043 if( keycode == "KeyO" || keycode == "ArrowRight" ){ // fold the element
8044   CurElement.open = !CurElement.open;
8045 }else
8046 if( keycode == "ArrowRight" ){ // unfold the element
8047   CurElement.open = true
8048 }else
8049 if( keycode == "ArrowLeft" ){ // unfold the element
8050   CurElement.open = false
8051 }else
8052 if( keycode == "KeyI" ){ // inspect the element
8053   e = CurElement
8054   //GLog.innerHTML =
8055   GJLog.append("Current Element: " + e + "<br>"
8056     + "name="+e.nodeName + ", "
8057     + "id="+e.id + ", "
8058     + "children="+e.childNodes.length + ", "
8059     + "parent="+e.parentNode.id + "<br>"
8060     + "text="+e.textContent)

```

```

8061     GStat.style.backgroundColor = "rgba(0,0,0,0.8)"
8062     return
8063 }else
8064 if( keycode == "KeyM" ){ // memory the position
8065     MemElement = CurElement
8066 }else
8067 if( keycode == "KeyN" || keycode == "ArrowDown" ){ // next element
8068     e = nextSib(CurElement)
8069     if( e != null ){
8070         setColor(CurElement,"SUMMARY","#fff")
8071         setColor(e,"SUMMARY","#8f8") // should be complement ?
8072         oe = CurElement
8073         CurElement = e
8074         //location.href = "#"+e.id;
8075         ScrollToElement(oe,e)
8076     }
8077 }else
8078 if( keycode == "KeyP" || keycode == "ArrowUp" ){ // previous element
8079     oe = CurElement
8080     e = prevSib(CurElement)
8081     if( e != null ){
8082         setColor(CurElement,"SUMMARY","#fff")
8083         setColor(e,"SUMMARY","#8f8") // should be complement ?
8084         CurElement = e
8085         //location.href = "#"+e.id;
8086         ScrollToElement(oe,e)
8087     }else{
8088         e = document.getElementById("GshBanner")
8089         if( e != null ){
8090             setColor(CurElement,"SUMMARY","#fff")
8091             CurElement = e
8092             ScrollToElement(oe,e)
8093         }else{
8094             e = document.getElementById("primary")
8095             if( e != null ){
8096                 setColor(CurElement,"SUMMARY","#fff")
8097                 CurElement = e
8098                 ScrollToElement(oe,e)
8099             }
8100         }
8101     }
8102 }else
8103 if( keycode == "KeyR" ){
8104     location.reload()
8105 }else
8106 if( keycode == "KeyJ" ){
8107     GshGrid.style.top = '120px';
8108     GshGrid.innerHTML = '>_{Down}';
8109 }else
8110 if( keycode == "KeyK" ){
8111     GshGrid.style.top = '0px';
8112     GshGrid.innerHTML = '({Up}';
8113 }else
8114 if( keycode == "KeyH" ){
8115     GshGrid.style.left = '0px';
8116     GshGrid.innerHTML = "({Left}";
8117 }else
8118 if( keycode == "KeyL" ){
8119     //GLog.innerHTML +=
8120     GJLog_append(
8121         'screen='+screen.width+'px'+<br>'+
8122         'window='+window.innerWidth+'px'+<br>'
8123     )
8124     GshGrid.style.left = (document.documentElement.clientWidth-160).toString(10)+'px';
8125     GshGrid.innerHTML = '({Right}';
8126 }else
8127 if( keycode == "KeyS" ){
8128     html_stop(GshMenuStop,true)
8129 }else
8130 if( keycode == "KeyF" ){
8131     html_fork()
8132 }else
8133 if( keycode == "KeyC" ){
8134     window.close()
8135 }else
8136 if( keycode == "KeyD" ){
8137     html_digest()
8138 }else
8139 if( keycode == "KeyV" ){
8140     e = document.getElementById('gsh-digest')
8141     if( e != null ){
8142         showDigest(e)
8143     }
8144 }
8145 }
8146 showCurElementPosition("[+key.code+" --");
8147 //if( document.getElementById("GPos") != null ){
8148 //GPos.innerHTML += "[+key.code+" --"
8149 //}
8150 //GShellResizeX("[+key.code+" --");
8151 }
8152 var initGSKC = false;
8153 function GShell_initKeyCommands(){
8154     if( initGSKC ){ return; } initGSKC = true;
8155     GShellResizeX("[INIT]");
8156     DisplaySize = '-- Display: '
8157     + 'screen='+screen.width+'px, ' + 'window='+window.innerWidth+'px';
8158     let {text, digest} = getDigest()
8159     //GLog.innerHTML +=
8160     GJLog_append(
8161         '-- GShell: ' + GshVersion.innerHTML + '\n' +
8162         '-- Digest: ' + digest + '\n' +
8163         DisplaySize
8164         //+ "<br>" + "-- LastVisit:<br>" + MyHistory
8165     )
8166     GShellResizeX(null);
8167 }
8168 //GShell_initKeyCommands();
8169 }
8170 //GShell_initKeyCommands();
8171 }
8172 // <a href="https://www.w3.org/TR/WebCryptoAPI/">Web Cryptography API</a>
8173 //Convert a string into an ArrayBuffer
8174 //from https://developers.google.com/web/updates/2012/06/How-to-convert-ArrayBuffer-to-and-from-String
8175 function str2ab(str) {
8176     const buf = new ArrayBuffer(str.length);
8177     const bufView = new Uint8Array(buf);
8178     for (let i = 0, strLen = str.length; i < strLen; i++) {
8179         bufView[i] = str.charCodeAt(i);
8180     }
8181     return buf;
8182 }
8183 function importPrivateKey(pem) {
8184     const binaryDerString = window.atob(pemContents);

```

```

8185 const binaryDer = str2ab(binaryDerString);
8186 return window.crypto.subtle.importKey(
8187     "pkcs8",
8188     binaryDer,
8189     {
8190         name: "RSA-PSS",
8191         modulusLength: 2048,
8192         publicExponent: new Uint8Array([1, 0, 1]),
8193         hash: "SHA-256",
8194     },
8195     true,
8196     ["sign"]
8197 );
8198 }
8199 //importPrivateKey(ppem)
8200
8201 //key = {}
8202 //buf = "abc"
8203 //enc = "xyzxxxxxx"; //crypto.publicEncrypt(key,buf)
8204 //b64 = btoa(enc)
8205 //dec = atob(b64)
8206 //GLog.innerHTML = "enc:" + b64 + ", dec:" + dec
8207 </script>
8208 */
8209
8210 /*
8211 <!-- ----- GJConsole BEGIN { ----- -->
8212 <span id="gjc" data-title="GJConsole" data-author="sato@its-more.jp">
8213 <details><summary>GJ Console</summary>
8214 <p>
8215 <span id="GJE_RootNode0"></span>
8216 <span id="GJCl_Container"></span>
8217 </p>
8218 <style id="GJConsoleStyle">
8219 .GJConsole {
8220     z-index:1000;
8221     width:400; height:200px;
8222     margin:2px;
8223     color:#fff; background-color:#66a;
8224     font-size:12px; font-family:monospace,Courier New;
8225 }
8226 </style>
8227
8228 <script id="GJConsoleScript" class="GJConsole">
8229 var PS1 = "% "
8230 function GJC_KeyDown(keyevent){
8231     key = keyevent.code
8232     if( key == "Enter" ){
8233         GJC_Command(this)
8234         this.value += "\n" + PS1 // prompt
8235     }else
8236     if( key == "Escape"){
8237         SuppressGJShell = false
8238         GshMenu.focus() // should be previous focus
8239     }
8240 }
8241 var GJC_SessionId
8242 function GJC_SetSessionId(){
8243     var xd = new Date()
8244     GJC_SessionId = xd.getTime() / 1000
8245 }
8246 GJC_SetSessionId()
8247 function GJC_Memory(mem,args,text){
8248     argv = args.split(' ')
8249     cmd = argv[0]
8250     argv.shift()
8251     args = argv.join(' ')
8252     ret = ""
8253
8254     if( cmd == 'clear' ){
8255         Permanent.setItem(mem,'')
8256     }else
8257     if( cmd == 'read' ){
8258         ret = Permanent.getItem(mem)
8259     }else
8260     if( cmd == 'save' ){
8261         val = Permanent.getItem(mem)
8262         if( val == null ){ val = "" }
8263         d = new Date()
8264         val += d.getTime()/1000+ "+GJC_SessionId+ " +document.URL+ " +args+"\n"
8265         val += text.value
8266         Permanent.setItem(mem,val)
8267     }else
8268     if( cmd == 'write' ){
8269         val = Permanent.getItem(mem)
8270         if( val == null ){ val = "" }
8271         d = new Date()
8272         val += d.getTime()/1000+ "+GJC_SessionId+ " +document.URL+ " +args+"\n"
8273         Permanent.setItem(mem,val)
8274     }else{
8275         ret = "Commands: write | read | save | clear"
8276     }
8277     return ret
8278 }
8279 // -- 2020-09-14 added TableEditor
8280 var GJE_CurElement = null; //GJE_RootNode
8281 GJE_NodeSaved = null
8282 GJE_TableNo = 1
8283 function GJE_StyleKeyCommand(kev){
8284     keycode = kev.code
8285     console.log('GJE-Key: '+keycode)
8286     if( keycode == 'Escape' ){
8287         GJE_SetStyle(this);
8288     }
8289     kev.stopPropagation()
8290     // https://developer.mozilla.org/en-US/docs/Web/API/Event/stopPropagation
8291 }
8292 var GJE_CommandMode = false
8293 function GJE_TableKeyCommand(kev,tab){
8294     wasCmdMode = GJE_CommandMode
8295     key = kev.code
8296     if( key == 'Escape' ){
8297         console.log("To command mode: "+tab.nodeName+"#" +tab.id)
8298         //tab.setAttribute('contenteditable','false')
8299         tab.style.caretColor = "blue"
8300         GJE_CommandMode = true
8301     }else
8302     if( key == "KeyA" ){
8303         tab.style.caretColor = "red"
8304         GJE_CommandMode = false
8305     }else
8306     if( key == "KeyI" ){
8307         tab.style.caretColor = "red"
8308         GJE_CommandMode = false

```

```

8309     }else
8310     if( key == "KeyO" ){
8311         tab.style.caretColor = "red"
8312         GJE_CommandMode = false
8313     }else
8314     if( key == "KeyJ" ){
8315         console.log("ROW-DOWN")
8316     }else
8317     if( key == "KeyK" ){
8318         console.log("ROW-UP")
8319     }else
8320     if( key == "Keyw" ){
8321         console.log("COL-FORW")
8322     }else
8323     if( key == "Keyb" ){
8324         console.log("COL-BACK")
8325     }
8326
8327     kev.stopPropagation()
8328     if( wasCmdMode ){
8329         kev.preventDefault()
8330     }
8331 }
8332 function GJE_DragEvent(ev,elem){
8333     x = ev.clientX
8334     y = ev.clientY
8335     console.log("Dragged: "+this.nodeName+'#'+this.id+' x='+x+' y='+y)
8336 }
8337 // https://developer.mozilla.org/en-US/docs/Web/API/DragEvent
8338 // https://www.w3.org/TR/uevents/#events-mouseevents
8339 function GJE_DropEvent(ev,elem){
8340     x = ev.clientX
8341     y = ev.clientY
8342     this.style.x = x
8343     this.style.y = y
8344     this.style.position = 'absolute' // 'fixed'
8345     this.parentNode = gsh // just for test
8346     console.log("Dropped: "+this.nodeName+'#'+this.id+' x='+x+' y='+y
8347         + " parent="+this.parentNode.id)
8348 }
8349 function GJE_SetTableStyle(ev){
8350     this.innerHTML = this.value; // sync. for external representation?
8351     if(false){
8352         stid = this.parentNode.id+this.id
8353         // and remove "span" at the end
8354         e = document.getElementById(stid)
8355         //alert('SetTableStyle #' +e.id+'\n'+this.value)
8356         if( e != null ){
8357             e.innerHTML = this.value
8358         }else{
8359             console.log('Style Not found: '+stid)
8360         }
8361         //alert('event StopPropagation: '+ev)
8362     }
8363 }
8364 function setCSSofClass(cclass,cstyle){
8365     const ss = document.styleSheets[3]; // 0, 1, 2, 3, ... ?
8366     rlen = ss.cssRules.length;
8367     let tabrule = null;
8368     rulex = -1
8369
8370     // should skip white space at the top of cstyle
8371     sel = cstyle.charAt(0);
8372     selector = sel+cclass;
8373     console.log('-- search style rule for '+selector)
8374
8375     for(let i = 0; i < rlen; i++){
8376         cr = ss.cssRules[i];
8377         console.log("CSS rule ['+i+'/' +rlen+''] '+cr.selectorText);
8378         if( cr.selectorText === selector ){ // css class selector
8379             tabrule = ss.cssRules[i];
8380             console.log("CSS rule found for:['+i+'/' +rlen+''] '+selector);
8381             ss.deleteRule(i);
8382             //rlen = ss.cssRules.length;
8383             rulex = i
8384             // should search and replace the property here
8385         }
8386     }
8387     // https://developer.mozilla.org/en-US/docs/Web/API/CSSStyleSheet/insertRule
8388     if( tabrule == null ){
8389         console.log("CSS rule NOT found for:['+rlen+''] '+selector);
8390         ss.insertRule(cstyle,rlen);
8391         ss.insertRule(cstyle,0); // override by 0?
8392         console.log("CSS rule inserted:['+(rlen+1)']\n"+cstyle);
8393     }else{
8394         ss.insertRule(cstyle,rlen);
8395         ss.insertRule(cstyle,0);
8396         console.log("CSS rule replaced:['+(rlen+1)']\n"+cstyle);
8397     }
8398 }
8399 function GJE_SetStyle(te){
8400     console.log('Apply the style to:'+te.id+'\n');
8401     console.log('Apply the style to:'+te.parentNode.id+'\n');
8402     console.log('Apply the style to:'+te.parentNode.class+'\n');
8403     cclass = te.parentNode.class;
8404     setCSSofClass(cclass,te.value); // should get selector part from
8405         // selector { rules }
8406
8407     if(false){
8408         //console.log('Apply the style:')
8409         //stid = this.parentNode.id+this.id+"
8410         //stid = this.id+".style"
8411         css = te.value
8412         stid = te.parentNode.id+".style"
8413         e = document.getElementById(stid)
8414         if( e != null ){
8415             //console.log('Apply the style:'+e.id+'\n'+te.value);
8416             console.log('Apply the style:'+e.id+'\n'+css);
8417             // e.innerHTML = css; //te.value;
8418             //ncss = e.sheet;
8419             //ncss.insertRule(te.value,ncss.cssRules.length);
8420         }else{
8421             console.log('No element to Apply the style: '+stid)
8422         }
8423         tblid = te.parentNode.id+".table";
8424         e = document.getElementById(tblid);
8425         if( e != null ){
8426             //e.setAttribute('style',css);
8427             e.setProperty('style',css,'!important');
8428         }
8429     }
8430 }
8431 function makeTable(argv){
8432     //tid = ''

```

```

8433     //cwe = GJE_CurElement
8434     cwe = GJCl_Container;
8435     //cwd = GJFactory;
8436     tid = 'table_' + GJE_TableNo
8437
8438     nt = new Text('\n')
8439     cwe.appendChild(nt)
8440
8441     ne = document.createElement('span'); // the container
8442     cwe.appendChild(ne)
8443     ne.id = tid + '-span'
8444     ne.setAttribute('contenteditable',true)
8445
8446     htspan = document.createElement('span'); // html part
8447     //htspan.id = tid + '-html'
8448     //ne.innerHTML = '\n'
8449     nt = new Text('\n')
8450     ne.appendChild(nt)
8451     ne.appendChild(htspan)
8452
8453     htspan.id = tid
8454     htspan.setAttribute('class',tid)
8455
8456     ne.setAttribute('draggable','true')
8457     ne.addEventListener('drag',GJE_DragEvent);
8458     ne.addEventListener('dragend',GJE_DropEvent);
8459
8460     var col = 3
8461     var row = 2
8462     if( argv[0] != null ){
8463         col = argv[0]
8464         argv.shift()
8465     }
8466     if( argv[0] != null ){
8467         row = argv[0]
8468         argv.shift()
8469     }
8470
8471     //ne.setAttribute('class',tid)
8472     ht = "\n"
8473     //ht += '<'+ 'table ' + 'id="'+tid+'"' + ' class="'+tid+'"'
8474     ht += '<'+ 'table '
8475         + ' onkeydown="GJE_TableKeyCommand(event,this)"'
8476         + ' ondrag="GJE_DragEvent(event,this)"\n'
8477         + ' ondragend="GJE_DropEvent(event,this)"\n'
8478         + ' draggable="true"\n'
8479         + ' contenteditable="true"'
8480         + '>\n'
8481     ht += '<'+ 'tbody>\n';
8482     for( r = 0; r < row; r++ ){
8483         ht += "<"+ "tr>\n"
8484         for( c = 0; c < col; c++ ){
8485             ht += "<"+ "td>"
8486             ht += "ABCDEFGHIJKLMNOPQRSTUVWXYZ".charAt(c) + r
8487             ht += "<"+ "/td>\n"
8488         }
8489         ht += "<"+ "/tr>\n"
8490     }
8491     ht += '<'+ '/tbody>\n';
8492     ht += '<'+ '/table>\n';
8493     htspan.innerHTML = ht;
8494     nt = new Text('\n')
8495     ne.appendChild(nt)
8496
8497     st = '#'+tid+' *{\n' // # for instanse specific
8498     + '  '+ 'border:1px solid #aaa;\n'
8499     + '  '+ 'background-color:#efe;\n'
8500     + '  '+ 'color:#222;\n'
8501     + '  '+ 'font-size:#14pt !important;\n'
8502     + '  '+ 'font-family:monospace,Courier New !important;\n'
8503     + '}' //+ hit ESC to apply *+/\n'
8504
8505     // wish script to be included
8506     //nj = document.createElement('script')
8507     //ne.appendChild(nj)
8508     //ne.innerHTML = 'function SetStyle(e){}'
8509
8510     // selector seems lost in dynamic style appending
8511     if(false){
8512         ns = document.createElement('style')
8513         ne.appendChild(ns)
8514         ns.id = tid + '.style'
8515         ns.innerHTML = '\n'+st
8516         nt = new Text('\n')
8517         ne.appendChild(nt)
8518     }
8519     setCSSofClass(tid,st); // should be in JavaScript script?
8520
8521     nx = document.createElement('textarea')
8522     ne.appendChild(nx)
8523     nx.id = tid + '-style_def'
8524     nx.setAttribute('class','GJ_StyleEditor')
8525     nx.spellcheck = false
8526     nx.cols = 60
8527     nx.rows = 10
8528     nx.innerHTML = '\n'+st
8529     nx.addEventListener('change',GJE_SetTableStyle);
8530     nx.addEventListener('keydown',GJE_StyleKeyCommand);
8531     //nx.addEventListener('click',GJE_SetTableStyle);
8532
8533     nt = new Text('\n')
8534     cwe.appendChild(nt)
8535
8536     GJE_TableNo += 1
8537     return 'created TABLE id="'+tid+'"'
8538 }
8539 function GJE_NodeEedit(argv){
8540     cwe = GJE_CurElement
8541     cmd = argv[0]
8542     argv.shift()
8543     args = argv.join(' ')
8544     ret = ""
8545
8546     if( cmd == '.u' || cmd == '.un' || cmd == 'undo' ){
8547         if( GJE_NodeSaved != null ){
8548             xn = GJE_RootNode
8549             GJE_RootNode = GJE_NodeSaved
8550             GJE_NodeSaved = xn
8551             ret = '-- did undo'
8552         }else{
8553             ret = '-- could not undo'
8554         }
8555         return ret
8556     }

```

```

8557 GJE_NodeSaved = GJE_RootNode.cloneNode()
8558 if( cmd == '.c' || cmd == '.cd' || cmd == 'cd' ){
8559     if( argv[0] == null ){
8560         ne = GJE_RootNode
8561     }else
8562     if( argv[0] == '..' ){
8563         ne = cwe.parentNode
8564     }else{
8565         ne = document.getElementById(argv[0])
8566     }
8567     if( ne != null ){
8568         GJE_CurElement = ne
8569         ret = "-- current node: " + ne.id
8570     }else{
8571         ret = "-- not found: " + argv[0]
8572     }
8573 }else
8574 if( cmd == '.mkt' || cmd == '.mktable' ){
8575     makeTable(argv)
8576 }else
8577 if( cmd == '.m' || cmd == '.mk' || cmd == 'mk' ){
8578     ne = document.createElement(argv[0])
8579     //ne.id = argv[0]
8580     ret = "-- created " + ne + " under " + cwe.tagName + "#" + cwe.id
8581     cwe.appendChild(ne)
8582     if( cmd == '.m' || cmd == '.mk' ){
8583         GJE_CurElement = ne
8584     }
8585 }else
8586 if( cmd == '.n' || cmd == '.nm' || cmd == 'nm' ){
8587     cwe.id = argv[0]
8588 }else
8589 if( cmd == '.r' || cmd == '.rm' || cmd == 'rm' ){
8590 }else
8591 if( cmd == '.h' || cmd == '.sh' || cmd == 'sh' ){
8592     s = argv.join(' ')
8593     cwe.innerHTML = s
8594 }else
8595 if( cmd == '.a' || cmd == '.sa' || cmd == 'sa' ){
8596     cwe.setAttribute(argv[0],argv[1])
8597 }else
8598 if( cmd == '.l' ){
8599 }else
8600 if( cmd == '.i' || cmd == '.ih' || cmd == 'ih' ){
8601     ret = cwe.innerHTML
8602 }else
8603 if( cmd == '.p' || cmd == '.pw' || cmd == 'pw' ){
8604     ret = cwe.nodeType + " " + cwe.tagName + " " + cwe.id
8605     for( we = cwe.parentNode; we != null; ){
8606         ret += "\n" + " " + we.nodeType + " " + we.tagName + " " + we.id
8607         we = we.parentNode
8608     }
8609 }else
8610 {
8611     ret = "Command: mk | rm \n"
8612     ret += " pw -- print current node\n"
8613     ret += " mk type -- make node with name and type\n"
8614     ret += " nm name -- set the id #name of current node\n"
8615     ret += " rm name -- remove named node\n"
8616     ret += " cd name -- change current node\n"
8617 }
8618 //alert(ret)
8619 return ret
8620 }
8621 function GJC_Command(text){
8622     lines = text.value.split('\n')
8623     line = lines[lines.length-1]
8624     argv = line.split(' ')
8625     text.value += '\n'
8626     if( argv[0] == '%' ){ argv.shift() }
8627     args0 = argv.join(' ')
8628     cmd = argv[0]
8629     argv.shift()
8630     args = argv.join(' ')
8631 }
8632 if( cmd == 'nolog' ){
8633     StopConsoleLog = true
8634 }else
8635 if( cmd == 'new' ){
8636     if( argv[0] == 'table' ){
8637         argv.shift()
8638         console.log('argv='+argv)
8639         text.value += makeTable(argv)
8640     }else
8641     if( argv[0] == 'console' ){
8642         text.value += GJ_NewConsole('GJ_Console')
8643     }else{
8644         text.value += '-- new { console | table }'
8645     }
8646 }else
8647 if( cmd == 'strip' ){
8648     //text.value += GJF_StripClass()
8649 }else
8650 if( cmd == 'css' ){
8651     sel = '#table_1'
8652     if(argv[0]!='0')
8653     rule1 = sel+'{color:#000 !important; background-color:#fff !important;}';
8654     else
8655     rule1 = sel+'{color:#f00 !important; background-color:#eef !important;}';
8656     document.styleSheets[3].deleteRule(0);
8657     document.styleSheets[3].insertRule(rule1,0);
8658     text.value += 'CSS rule added: '+rule1
8659 }else
8660 if( cmd == 'print' ){
8661     e = null;
8662     if( e == null ){
8663         e = document.getElementById('GJFactory_0')
8664     }
8665     if( e == null ){
8666         e = document.getElementById('GJFactory_1')
8667     }
8668     if( argv[0] != null ){
8669         id = argv[0]
8670         if( id == 'f' ){
8671             //e = document.getElementById('GJE_RootNode');
8672         }else{
8673             e = document.getElementById(id)
8674         }
8675         if( e != null ){
8676             text.value += e.outerHTML
8677         }else{
8678             text.value += "Not found: " + id
8679         }
8680     }else{

```

```

8681         text.value += GJE_RootNode.outerHTML
8682         //text.value += e.innerHTML
8683     }
8684 }else
8685 if( cmd == 'destroy' ){
8686     text.value += GJFactory_Destroy()
8687 }else
8688 if( cmd == 'save' ){
8689     e = document.getElementById('GJFactory')
8690     Permanent.setItem('GJFactory-1',e.innerHTML)
8691     text.value += "-- Saved GJFactory"
8692 }else
8693 if( cmd == 'load' ){
8694     gjf = Permanent.getItem('GJFactory-1')
8695     e = document.getElementById('GJFactory')
8696     e.innerHTML = gjf
8697     // must restore EventListener
8698     text.value += "-- EventListener was not restored"
8699 }else
8700 if( cmd.charAt(0) == '.' ){
8701     argv0 = args0.split('.')
8702     text.value += GJE_NodeEdit(argv0)
8703 }else
8704 if( cmd == 'cont' ){
8705     bannerIsStopping = false
8706     GshMenuStop.innerHTML = "Stop"
8707 }else
8708 if( cmd == 'date' ){
8709     text.value += DateLong()
8710 }else
8711 if( cmd == 'echo' ){
8712     text.value += args
8713 }else
8714 if( cmd == 'fork' ){
8715     html_fork()
8716 }else
8717 if( cmd == 'last' ){
8718     text.value += MyHistory
8719     //h = document.createElement("span")
8720     //h.innerHTML = MyHistory
8721     //text.value += h.innerHTML
8722     //tx = MyHistory.replace("\n","")
8723     //text.value += tx.replace("<"+"br>","\n") + "xxxx<"+"br>yyyy"
8724 }else
8725 if( cmd == 'ne' ){
8726     text.value += GJE_NodeEdit(argv)
8727 }else
8728 if( cmd == 'reload' ){
8729     location.reload()
8730 }else
8731 if( cmd == 'mem' ){
8732     text.value += GJC_Memory('GJC_Storage',args,text)
8733 }else
8734 if( cmd == 'stop' ){
8735     bannerIsStopping = true
8736     GshMenuStop.innerHTML = "Start"
8737 }else
8738 if( cmd == 'who' ){
8739     text.value += "SessionId="+GJC_SessionId+" "+document.URL
8740 }else
8741 if( cmd == 'wall' ){
8742     text.value += GJC_Memory('GJC_Wall','write',text)
8743 }else
8744 {
8745     text.value += "Commands: help | echo | date | last \n"
8746     + '      new | save | load | mem \n'
8747     + '      who | wall | fork | nife'
8748 }
8749 }
8750
8751 function GJC_Input(){
8752     if( this.value.endsWith("\n") ){ // remove NL added by textarea
8753         this.value = this.value.slice(0,this.value.length-1)
8754     }
8755 }
8756
8757 var GCJ_Id = null
8758 function GJC_Resize(){
8759     GJC_Id.style.zIndex = 20000
8760     //GJC_Id.style.width = window.innerWidth - 16
8761     GJC_Id.style.width = '100%';
8762     GJC_Id.style.height = 300;
8763     GJC_Id.style.backgroundColor = "rgba(0,64,16,1.0)" // blackboard color
8764     GJC_Id.style.color = "rgba(255,255,255,1.0)"
8765 }
8766 function GJC_FocusIn(){
8767     this.spellcheck = false
8768     SuppressGJShell = true
8769     this.onkeydown = GJC_Keydown
8770     GJC_Resize()
8771 }
8772 function GJC_FocusOut(){
8773     SuppressGJShell = false
8774     this.removeEventListener('keydown',GJC_Keydown);
8775 }
8776 window.addEventListener('resize',GJC_Resize);
8777
8778 function GJC_OnStorage(e){
8779     //alert('Got Message')
8780     //GJC.value += "\n((ReceivedMessage))\n"
8781 }
8782 window.addEventListener('storage',GJC_OnStorage);
8783 //window.addEventListener('storage',()=>{alert('GotMessage')})
8784
8785 function GJC_Setup(gjcId){
8786     //gjcId.style.width = gsh.getBoundingClientRect().width
8787     gjcId.style.width = '100%';
8788     gjcId.value = "GJShell Console // " + GshVersion.innerHTML + "\n"
8789     //gjcId.value += "Date: " + DateLong() + "\n"
8790     gjcId.value += PS1
8791     gjcId.onfocus = GJC_FocusIn
8792     gjcId.addEventListener('input',GJC_Input);
8793     gjcId.addEventListener('focusout',GJC_FocusOut);
8794     GJC_Id = gjcId
8795 }
8796 function GJC_Clear(id){
8797 }
8798
8799 function GJConsole_initConsole(){
8800     if( document.getElementById("GJC_0") != null ){
8801         GJC_Setup(GJC_0)
8802     }else{
8803         GJC1_Container.innerHTML = '<'+
8804             '+textarea id="GJC_1" class="GJConsole"><'+'/textarea>';
8805         GJC_Setup(GJC_1)

```

```

8805     factory = document.createElement('span');
8806     gsh.appendChild(factory)
8807     GJE_RootNode = factory;
8808     GJE_CurElement = GJE_RootNode;
8809     }
8810 }
8811 var initGJCF = false;
8812 function GJConsole_initFactory(){
8813     if( initGJCF ){ return; } initGJCF = true;
8814     GShell_initKeyCommands();
8815     GJConsole_initConsole();
8816 }
8817 //GJConsole_initFactory();
8818 // TODO: focus handling
8819 </script>
8820 <style>
8821 .GJ_StyleEditor {
8822     font-size:9pt !important;
8823     font-family:Courier New, monospace !important;
8824 }
8825 </style>
8826 </details>
8827 </span>
8828 <!-- ----- GJConsole END } ----- -->
8829 */
8830 */
8831 */
8832 /*
8833 <span id="BlinderText">
8834 <style id="BlinderTextStyle">
8835 #GJLinkView {
8836     xxposition:absolute; z-index:5000;
8837     position:relative;
8838     display:block;
8839     left:8px;
8840     color:#fff;
8841     width:800px; height:300px; resize:both;
8842     margin:0px; padding:4px;
8843     background-color:rgba(200,200,200,0.5) !important;
8844 }
8845 .MsgText {
8846     width:578px !important;
8847     resize:both !important;
8848     color:#000 !important;
8849 }
8850 .GjNote {
8851     font-family:Georgia !important;
8852     font-size:13pt !important;
8853     color:#22a !important;
8854 }
8855 .textField {
8856     display:inline;
8857     border:0.5px solid #444;
8858     border-radius:3px;
8859     color:#000; background-color:#fff;
8860     width:106pt; height:18pt;
8861     margin:2px;
8862     padding:2px;
8863     resize:none;
8864     vertical-align:middle;
8865     font-size:10pt; font-family:Courier New;
8866 }
8867 .textLabel {
8868     border:0px solid #000 !important;
8869     background-color:rgba(0,0,0,0);
8870 }
8871 .textURL {
8872     width:300pt !important;
8873     border:0px solid #000 !important;
8874     background-color:rgba(0,0,0,0);
8875 }
8876 .VisibleText {
8877 }
8878 .BlinderText {
8879     color:#000; background-color:#eee;
8880 }
8881 .joinButton {
8882     font-family:Georgia !important;
8883     font-size:11pt;
8884     line-height:1.1;
8885     height:18pt;
8886     width:50pt;
8887     padding:3px !important;
8888     text-align:center !important;
8889     border-color:#aaa !important;
8890     border-radius:5px;
8891     color:#fff; background-color:#4a4 !important;
8892     vertical-align:middle !important;
8893 }
8894 .SendButton {
8895     vertical-align:top;
8896 }
8897 .ws0_log {
8898     font-size:10pt;
8899     color:#000 !important;
8900     line-height:1.0;
8901     background-color:rgba(255,255,255,0.7) !important;
8902     font-family:Courier New,monospace !important;
8903     width:99.3%;
8904     white-space:pre;
8905 }
8906 </style>
8907
8908 <!-- Form autofill test
8909 Location: <input id="xxserv" type="text" value="https://192.168.10.1/boafrm/FormLogin" size="80">
8910 <form method="POST" id="xxform" action="https://192.168.10.1/boafrm/FormLogin">
8911 dest? <input id="XDS" name="dest" type="text" size="80" value="/index_contents.html">
8912 -->
8913 <details><summary>Form Auto. Filling</summary>
8914 <style>
8915 .xxinput { width:260pt !important; line-height:1.1 !important; margin:1px;
8916     display:inline !important; font-size:10pt !important; padding:1px !important;
8917 }
8918 </style>
8919 <span style="font-family:Courier New;color:black;font-size:12pt;" onactive="">
8920 <form method="POST" id="xxform" action="https://192.168.10.1/" style="white-space:pre;">
8921 Location: <input id="xxserv" class="xxinput" type="text" value="https://192.168.10.1/" size="80">
8922 Username: <input id="xxuser" class="xxinput" name="user" type="text" autocomplete="on">
8923 Password: <input id="xxpass" class="xxinput" name="pass" type="password" autocomplete="on">
8924 SessionId:<input id="xxssid" class="xxinput" name="SESSION_ID" type="text" size="80">
8925 <input id="xxsubm" class="xxinput" type="submit" value="Submit"></form>
8926 </span>
8927 <script>
8928 function XXSetFormAction(){

```



```

8929     xxform.setAttribute('action',xxserv.value);
8930 }
8931 xxform.setAttribute('action',xxserv.value);
8932 xxserv.addEventListener('change',XXSetFormAction);
8933 //xxserv.value = location.href;
8934 </script>
8935 </details>
8936 */
8937
8938 /*
8939 <details id="BlinderTextClass" class="gsh-src"><summary>BlinderText</summary>
8940 <span id="BlinderTextScript">
8941 // https://w3c.github.io/uievents/#event-type-keydown
8942 //
8943 // 2020-09-21 class BlinderText - textarea element not to be readable
8944 //
8945 // BlinderText attributes
8946 // bl_plainText - null
8947 // bl_hideChecksum - [false]
8948 // bl_showLength - [false]
8949 // bl_visible - [false]
8950 // data-bl_config - []
8951 // - min. length
8952 // - max. length
8953 // - acceptable charset in generate text
8954 //
8955 function BlinderChecksum(text){
8956 plain = text.bl_plainText;
8957 return strCRC32(plain,plain.length).toFixed(0);
8958 }
8959 function BlinderKeydown(ev){
8960 pass = ev.target
8961 if( ev.code == 'Enter' ){
8962 ev.preventDefault();
8963 }
8964 ev.stopPropagation()
8965 }
8966 function BlinderKeyUpl(ev){
8967 blind = ev.target
8968 if( ev.code == 'Backspace' ){
8969 blind.bl_plainText = blind.bl_plainText.slice(0,blind.bl_plainText.length-1)
8970 }else
8971 if( and(ev.code == 'KeyV', ev.ctrlKey) ){
8972 blind.bl_visible = !blind.bl_visible;
8973 }else
8974 if( and(ev.code == 'KeyL', ev.ctrlKey) ){
8975 blind.bl_showLength = !blind.bl_showLength;
8976 }else
8977 if( and(ev.code == 'KeyU', ev.ctrlKey) ){
8978 blind.bl_plainText = "";
8979 }else
8980 if( and(ev.code == 'KeyR', ev.ctrlKey) ){
8981 checksum = BlinderChecksum(blind);
8982 blind.bl_plainText = checksum; //.toString(32);
8983 }else
8984 if( ev.code == 'Enter' ){
8985 ev.stopPropagation();
8986 ev.preventDefault();
8987 return;
8988 }else
8989 if( ev.key.length != 1 ){
8990 console.log('KeyUp: '+ev.code+'/'+ev.key);
8991 return;
8992 }else{
8993 blind.bl_plainText += ev.key;
8994 }
8995
8996 leng = blind.bl_plainText.length;
8997 //console.log('KeyUp: '+ev.code+'/'+blind.bl_plainText);
8998 checksum = BlinderChecksum(blind) % 10; // show last one digit only
8999
9000 visual = '';
9001 if( !blind.bl_hideCheckSum || blind.bl_showLength ){
9002 visual += '[';
9003 }
9004 if( !blind.bl_hideCheckSum ){
9005 visual += '#'+checksum.toString(10);
9006 }
9007 if( blind.bl_showLength ){
9008 visual += '/' + leng;
9009 }
9010 if( !blind.bl_hideCheckSum || blind.bl_showLength ){
9011 visual += ']' ;
9012 }
9013 if( blind.bl_visible ){
9014 visual += blind.bl_plainText;
9015 }else{
9016 visual += '*'.repeat(leng);
9017 }
9018 blind.value = visual;
9019 }
9020 function BlinderKeyUp(ev){
9021 BlinderKeyUpl(ev);
9022 ev.stopPropagation();
9023 }
9024 // https://w3c.github.io/uievents/#keyboardevent
9025 // https://w3c.github.io/uievents/#uievent
9026 // https://dom.spec.whatwg.org/#event
9027 function BlinderTextEvent(){
9028 ev = event;
9029 blind = ev.target;
9030 console.log('Event '+ev.type+'@'+blind.nodeName+'#'+blind.id)
9031 if( ev.type == 'keyup' ){
9032 BlinderKeyUp(ev);
9033 }else
9034 if( ev.type == 'keydown' ){
9035 BlinderKeydown(ev);
9036 }else{
9037 console.log('thru-event '+ev.type+'@'+blind.nodeName+'#'+blind.id)
9038 }
9039 }
9040 << textarea hidden id="BlinderTextClassDef" class="textField"
9041 // onkeydown="BlinderTextEvent()" onkeyup="BlinderTextEvent()"
9042 // spellcheck="false">< /textarea>
9043 << textarea hidden id="gj_pass1"
9044 // class="textField BlinderText"
9045 // placeholder="PassWord1"
9046 // onkeydown="BlinderTextEvent()"
9047 // onkeyup="BlinderTextEvent()"
9048 // spellcheck="false">< /textarea>
9049 function SetupBlinderText(parent,txa,phold){
9050 if( txa == null ){
9051 txa = document.createElement('textarea');
9052 //txa.id = id;

```

```

9053     }
9054     txa.setAttribute('class','textField BlinderText');
9055     txa.setAttribute('placeholder',phold);
9056     txa.setAttribute('onkeydown','BlinderTextEvent()');
9057     txa.setAttribute('onkeyup','BlinderTextEvent()');
9058     txa.setAttribute('spellcheck','false');
9059     //txa.setAttribute('bl_plainText','false');
9060     txa.bl_plainText = '';
9061     //parent.appendChild(txa);
9062 }
9063 function DestroyBlinderText(txa){
9064     txa.removeAttribute('class');
9065     txa.removeAttribute('placeholder');
9066     txa.removeAttribute('onkeydown');
9067     txa.removeAttribute('onkeyup');
9068     txa.removeAttribute('spellcheck');
9069     txa.bl_plainText = '';
9070 }
9071 //
9072 // visible textarea like Username
9073 //
9074 function VisibleTextEvent(){
9075     if( event.code == 'Enter' ){
9076         if( event.target.NoEnter ){
9077             event.preventDefault();
9078         }
9079     }
9080     event.stopPropagation();
9081 }
9082 function SetupVisibleText(parent,txa,phold){
9083     if( false ){
9084         txa.setAttribute('class','textField VisibleText');
9085     }else{
9086         newclass = txa.getAttribute('class');
9087         if( and(newclass != null, newclass != '') ){
9088             newclass += ' ';
9089         }
9090         newclass += 'VisibleText';
9091         txa.setAttribute('class',newclass);
9092     }
9093     //console.log('SetupVisibleText class='+txa.class);
9094     txa.setAttribute('placeholder',phold);
9095     txa.setAttribute('onkeydown','VisibleTextEvent()');
9096     txa.setAttribute('onkeyup','VisibleTextEvent()');
9097     txa.setAttribute('spellcheck','false');
9098     cols = txa.getAttribute('cols');
9099     if( cols != null ){
9100         txa.style.width = '580px';
9101         //console.log('VisualText#'+txa.id+' cols='+cols)
9102     }else{
9103         //console.log('VisualText#'+txa.id+' NO cols')
9104     }
9105     rows = txa.getAttribute('rows');
9106     if( rows != null ){
9107         txa.style.height = '30px';
9108         txa.style.resize = 'both';
9109         txa.NoEnter = false;
9110     }else{
9111         txa.NoEnter = true;
9112     }
9113 }
9114 function DestroyVisibleText(txa){
9115     txa.removeAttribute('class');
9116     txa.removeAttribute('placeholder');
9117     txa.removeAttribute('onkeydown');
9118     txa.removeAttribute('onkeyup');
9119     txa.removeAttribute('spellcheck');
9120     cols = txa.removeAttribute('cols');
9121 }
9122 </span>
9123 <script>
9124 js = document.getElementById('BlinderTextScript');
9125 eval(js.innerHTML);
9126 //js.outerHTML = ""
9127 </script>
9128
9129 </details>
9130 </span>
9131 */
9132 /*
9133 <script id="GJLinkScript">
9134 function gjkey_hash(text){
9135     return strCRC32(text,text.length) % 0x10000;
9136 }
9137
9138 function gj_addlog(e,msg){
9139     now = (new Date().getTime() / 1000).toFixed(3);
9140     tstp = '['+now+']';
9141     e.value += tstp + msg;
9142     e.scrollTop = e.scrollHeight;
9143 }
9144 function gj_addlog_cl(msg){
9145     ws0_log.value += '(console.log) ' + msg + '\n';
9146 }
9147 var GJ_Channel = null;
9148 var GJ_Log = null;
9149 var gjX; // the global variable
9150 function GJ_Join(){
9151     target = gj_join;
9152     if( target.value == 'Leave' ){
9153         GJ_Channel.close();
9154         GJ_Channel = null;
9155         target.value = 'Join';
9156         return;
9157     }
9158 }
9159 var ws0;
9160 var ws0_log;
9161
9162 sav_console_log = console.error
9163 console.error = gj_addlog_cl
9164 ws0 = new WebSocket(gj_serv.innerHTML);
9165 console.error = sav_console_log
9166
9167 GJ_Channel = ws0;
9168 ws0_log = document.getElementById('ws0_log');
9169 GJ_Log = ws0_log;
9170
9171 now = (new Date().getTime() / 1000).toFixed(3);
9172 const wsstats = ["CONNECTING","OPEN","CLOSING","CLOSED"];
9173 cst = wsstats[ws0.readyState];
9174 gj_addlog(ws0_log,'stat '+ws0.readyState+'('+cst+') : GJ Linked\n');
9175
9176 ws0.addEventListener('error', function(event){

```

```

9177     gj_addlog(ws0_log,'stat error : transport error?\n');
9178 });
9179 ws.addEventListener('open', function(event){
9180     GJLinkView.style.zIndex = 10000;
9181     //console.log('#'+GJLinkView.id+'.zIndex'+GJLinkView.style.zIndex);
9182     date1 = new Date().getTime();
9183     date2 = (date1 / 1000).toFixed(3);
9184     seed = date1.toString(16);
9185
9186     // user name and key
9187     user = document.getElementById('gj_user').value;
9188     if( user.length == 0 ){
9189         gj_user.value = 'nemo';
9190         user = 'nemo';
9191     }
9192     key1 = document.getElementById('gj_ukey').bl_plainText;
9193     ukey = gjkey_hash(seed+user+key1).toString(16);
9194
9195     // session name and key
9196     chan = document.getElementById('gj_chan').value;
9197     if( chan.length == 0 ){
9198         gj_chan.value = 'main';
9199         chan = 'main';
9200     }
9201     key2 = document.getElementById('gj_ckey').bl_plainText;
9202     ckey = gjkey_hash(seed+chan+key2).toString(16);
9203
9204     msg = date2 + ' JOIN ' + user + '|' + chan + ' ' + ukey + ':' + ckey;
9205     gj_addlog(ws0_log,'send '+msg+'\n');
9206     ws0.send(msg);
9207
9208     target.value = 'Leave';
9209     //console.log('[ '+date2+' ] #' +target.id+' '+target.value+'\n');
9210     //gj_addlog(ws0_log,'label '+target.value+'\n');
9211 });
9212 ws.addEventListener('message', function(event){
9213     now = (new Date().getTime() / 1000).toFixed(3);
9214     msg = event.data;
9215     gj_addlog(ws0_log,'recv '+msg+'\n');
9216
9217     argv = msg.split(' ');
9218     tstamp = argv[0];
9219     argv.shift();
9220     if( argv[0] == 'reload' ){
9221         location.reload()
9222     }
9223     argv.shift(); // command
9224     argv.shift(); // from/to
9225     if( argv[0] == 'auth' ){
9226         // doing authorization required
9227     }
9228     if( argv[0] == 'echo' ){
9229         now = (new Date().getTime() / 1000).toFixed(3);
9230         msg = now+ ' ' +RESP +argv.join(' ');
9231         gj_addlog(ws0_log,'send '+msg+'\n');
9232         ws0.send(msg);
9233     }
9234     if( argv[0] == 'eval' ){
9235         argv.shift();
9236         js = argv.join(' ');
9237         ret = eval(js); // <----- eval()
9238         gj_addlog(ws0_log,'eval '+js+' '+ret+'\n');
9239         now = (new Date().getTime() / 1000).toFixed(3);
9240         msg = now + ' ' + RESP + ret;
9241         ws0.send(msg);
9242         gj_addlog(ws0_log,'send '+msg+'\n')
9243     }
9244 });
9245 ws.addEventListener('close', function(event){
9246     if( GJ_Channel == null ){
9247         gj_addlog(ws0_log,'stat OK : GJ UnLinked\n');
9248         return;
9249     }
9250     GJ_Channel.close();
9251     GJ_Channel = null;
9252     target.value = 'Join';
9253     gj_addlog(ws0_log,'stat error : close : GJ UnLinked unexpectedly\n');
9254 });
9255 }
9256 function GJ_SendMessageUserPass(user,chan,msgbody){
9257     now = (new Date().getTime() / 1000).toFixed(3);
9258     msg = now + ' ISAY '+user+'|'+chan+' '+msgbody;
9259     gj_addlog(GJ_Log,'send '+msg+'\n');
9260     GJ_Channel.send(msg);
9261 }
9262 function GJ_SendMessage(msgbody){
9263     if( GJ_Channel == null ){
9264         gj_addlog(ws0_log,'stat error : send : GJ not Linked\n');
9265         return;
9266     }
9267     //target = event.target;
9268     user = document.getElementById('gj_user').value;
9269     chan = document.getElementById('gj_chan').value;
9270     GJ_SendMessageUserPass(user,chan,msgbody);
9271 }
9272 function GJ_Send(){
9273     msgbody = gj_sendText.value;
9274     GJ_SendMessage(msgbody);
9275 }
9276 </script>
9277
9278 <!-- ----- GJLINK ----- -->
9279 <!--
9280 - User can subscribe to a channel
9281 - A channel will be broadcasted
9282 - A channel can be a pattern (regular expression)
9283 - User is like From:(me) and channel is like To: or Recipient:
9284 - like VIABUS
9285 - watch message with SENDME, WATCH, CATCH, HEAR, or so
9286 - routing with path expression or name pattern (with routing with DNS like system)
9287 -->
9288 */
9289
9290 <<span id="GJLinkGolang">
9291 <<details id="GshWebSocket" class="gsh-src"><summary>Golang / JavaScript Link</summary>
9292 // 2020-0920 created
9293 // <a href="https://pkg.go.dev/golang.org/x/net/websocket">WS</a>
9294 // <a href="https://godoc.org/golang.org/x/net/websocket">WS</a>
9295 // INSTALL: go get golang.org/x/net/websocket
9296 // INSTALL: sudo {apt,yum} install git (if git is not installed yet)
9297 // import "golang.org/x/net/websocket"
9298 const gshws_origin = "http://localhost:9999"
9299 const gshws_server = "localhost:9999"
9300 const gshws_port = 9999

```

```

9301 const gshws_path = "glink1"
9302 const gshws_url = "ws://" + gshws_server + "/" + gshws_path
9303 const GSHWS_MSGSIZE = (8 * 1024)
9304 func fmtstring(fmts string, params ...interface{})(string){
9305     return fmt.Sprintf(fmts, params...)
9306 }
9307 func GSHWS_MARK(what string)(string){
9308     now := time.Now()
9309     us := fmtstring("%06d", now.UnixNano() / 1000)
9310     mark := ""
9311     if (!AtConsoleLineTop){
9312         mark += "\n"
9313         AtConsoleLineTop = true
9314     }
9315     mark += "[" + now.Format(time.Stamp) + "." + us + "] -GJ- " + what + ": "
9316     return mark
9317 }
9318 func gchk(what string, err error){
9319     if (err != nil){
9320         panic(GSHWS_MARK(what) + err.Error())
9321     }
9322 }
9323 func glog(what string, fmts string, params ...interface{})(
9324     fmt.Print(GSHWS_MARK(what))
9325     fmt.Printf(fmts + "\n", params...)
9326 }
9327
9328 var WSV = []*websocket.Conn{}
9329 func jsend(argv []string){
9330     if len(argv) <= 1 {
9331         fmt.Printf("--Ij %v [-m] command arguments\n", argv[0])
9332         return
9333     }
9334     argv = argv[1:]
9335     if (len(WSV) == 0){
9336         fmt.Printf("--Ej-- No link now\n")
9337         return
9338     }
9339     if (1 < len(WSV)){
9340         fmt.Printf("--Ij-- multiple links (%v)\n", len(WSV))
9341     }
9342
9343     multicast := false // should be filtered with regexp
9344     if (0 < len(argv) && argv[0] == "-m"){
9345         multicast = true
9346         argv = argv[1:]
9347     }
9348     args := strings.Join(argv, " ")
9349
9350     now := time.Now()
9351     msec := now.UnixNano() / 1000000
9352     tstamp := fmtstring("%.3f", float64(msec)/1000.0)
9353     msg := fmtstring("%v SEND gshell" * %v", tstamp, args)
9354
9355     if (multicast){
9356         for i, ws := range WSV {
9357             wn, werr := ws.Write([]byte(msg))
9358             if (werr != nil){
9359                 fmt.Printf("[%v] wn=%v, werr=%v\n", i, wn, werr)
9360             }
9361             glog("SQ", fmtstring("(%v) %v", wn, msg))
9362         }
9363     }else{
9364         i := 0
9365         ws := WSV[i]
9366         wn, werr := ws.Write([]byte(msg))
9367         if (werr != nil){
9368             fmt.Printf("[%v] wn=%v, werr=%v\n", i, wn, werr)
9369         }
9370         glog("SQ", fmtstring("(%v) %v", wn, msg))
9371     }
9372 }
9373
9374 func serv1(ws *websocket.Conn) {
9375     WSV = append(WSV, ws)
9376     //fmt.Print("\n")
9377     glog("CO", "accepted connections[%v]", len(WSV))
9378     //remoteAddr := ws.RemoteAddr
9379     //fmt.Printf("-- accepted %v\n", remoteAddr)
9380     //fmt.Printf("-- accepted %v\n", ws.Config())
9381     //fmt.Printf("-- accepted %v\n", ws.Config().Header)
9382     //fmt.Printf("-- accepted %v // %v\n", ws, serv1)
9383
9384     var reqb = make([]byte, GSHWS_MSGSIZE)
9385     for {
9386         rn, rerr := ws.Read(reqb)
9387         if (rerr != nil || rn < 0){
9388             glog("SQ", fmtstring("%v, %v", rn, rerr))
9389             break
9390         }
9391         req := string(reqb[0:rn])
9392         glog("SQ", fmtstring("(%v) %v", rn, req))
9393
9394         margv := strings.Split(req, " ")
9395         margv = margv[1:]
9396         if (0 < len(margv)){
9397             if (margv[0] == "RESP"){
9398                 // should forward to the destination
9399                 continue;
9400             }
9401         }
9402         now := time.Now()
9403         msec := now.UnixNano() / 1000000
9404         tstamp := fmtstring("%.3f", float64(msec)/1000.0)
9405         res := fmtstring("%v "+CAST+" %v", tstamp, req)
9406         wn, werr := ws.Write([]byte(res))
9407         gchk("SE", werr)
9408         glog("SR", fmtstring("(%v) %v", wn, string(res)))
9409     }
9410     glog("SF", "WS response finish")
9411
9412     wsv := []*websocket.Conn{}
9413     wsx := 0
9414     for i, v := range WSV {
9415         if (v != ws){
9416             wsx = i
9417             wsv = append(wsv, v)
9418         }
9419     }
9420     WSV = wsv
9421     //glog("CO", "closed %v", ws)
9422     glog("CO", "closed connection [%v/%v]", wsx+1, len(WSV)+1)
9423     ws.Close()
9424 }
9425 // url := [scheme://]host[:port][path]

```

```

9425 func decomp_URL(url string){
9426 }
9427 func full_wsURL(){
9428 }
9429 func gj_server(argv []string) {
9430     gjserv := gshws_url
9431     gjport := gshws_server
9432     gjpath := gshws_path
9433     gjscheme := "ws"
9434
9435     //cmd := argv[0]
9436     argv = argv[1:]
9437     if( 1 <= len(argv) ){
9438         serv := argv[0]
9439         if( 0 < strings.Index(serv,"/") ){
9440             schemev := strings.Split(serv,"://")
9441             gjscheme = schemev[0]
9442             serv = schemev[1]
9443         }
9444         if( 0 < strings.Index(serv,"/") ){
9445             pathv := strings.Split(serv,"/")
9446             serv = pathv[0]
9447             gjpath = pathv[1]
9448         }
9449         servv := strings.Split(serv,":")
9450         host := "localhost"
9451         port := 9999
9452         if( servv[0] != "" ){
9453             host = servv[0]
9454         }
9455         if( len(servv) == 2 ){
9456             fmt.Sprintf(servv[1],"%d",&port)
9457         }
9458         //glog("LC","hostport=%v (%v : %v)",servv,host,port)
9459         gjport = fmt.Sprintf("%v:%v",host,port)
9460         gjserv = gjscheme + "://" + gjport + "/" + gjpath
9461     }
9462     glog("LS",fmtstring("listening at %v",gjserv))
9463     http.Handle("/"+gjpath,websocket.Handler(serv))
9464     err := error(nil)
9465     if( gjscheme == "ws" ){
9466         // https://golang.org/pkg/net/http/#ListenAndServeTLS
9467         //err = http.ListenAndServeTLS(gjport,nil)
9468     }else{
9469         err = http.ListenAndServe(gjport,nil)
9470     }
9471     gchk("LE",err)
9472 }
9473
9474 func gj_client(argv []string) {
9475     glog("CS",fmtstring("connecting to %v",gshws_url))
9476     ws, err := websocket.Dial(gshws_url,"",gshws_origin)
9477     gchk("C",err)
9478
9479     var resb = make([]byte, GSHWS_MSGSIZE)
9480     for qi := 0; qi < 3; qi++ {
9481         req := fmtstring("Hello, GShell! (%v)",qi)
9482         wn, werr := ws.Write([]byte(req))
9483         glog("QM",fmtstring("(%v) %v",wn,req))
9484         gchk("QE",werr)
9485         rn, rerr := ws.Read(resb)
9486         gchk("RE",rerr)
9487         glog("RM",fmtstring("(%v) %v",rn,string(resb)))
9488     }
9489     glog("CF","WS request finish")
9490 }
9491 //</details></span>
9492
9493 /*
9494 <details><summary>GJ Link</summary>
9495 <span id="GJLinkView" class="GJLinkView">
9496 <p>
9497 <note class="GJNote">Execute command "gsh gj server" on the localhost and push the Join button:</note>
9498 </p>
9499
9500 <span id="GJLink_1">
9501 <div id="GJLink_ServerSet"></div>
9502
9503 <div>
9504 <input id="gj_join" type="button" class="joinButton" onclick="GJ_Join()" value="Join">
9505 <span id="GJLink_Account"></span>
9506 </div>
9507
9508 <div>
9509 <input id="gj_sendButton" type="button" class="joinButton SendButton" onclick="GJ_Send()" value="Send" data-bodyid="gj_sendText">
9510 <span id="GJLink_SendArea"></span>
9511 </div>
9512
9513 <div id="ws0_log_container"></div>
9514 </span>
9515 </span>
9516
9517 <script>
9518 function setupGJLinkArea(){
9519     GJLink_ServerSet.innerHTML = '<'+<span id="gj_serv_label"
9520 + ' class="textField textLabel">Server: <'+</span>
9521 + '<'+<span id="gj_serv" class="textField textURL" contenteditable><'+</span>';
9522
9523     GJLink_Account.innerHTML = '<'+<textarea id="gj_user" class="textField"><'+</textarea>
9524 + '<'+<textarea id="gj_ukey" class="textField"><'+</textarea>
9525 + '<'+<textarea id="gj_chan" class="textField"><'+</textarea>
9526 + '<'+<textarea id="gj_ckey" class="textField"><'+</textarea>';
9527
9528     GJLink_SendArea.innerHTML =
9529 '<'+<textarea id="gj_sendText" class="textField MsggText" cols=60 rows=2><'+</textarea>';
9530
9531     ws0_log_container.innerHTML = '<'+<textarea id="ws0_log" class="ws0_log"
9532 +', cols=100 rows=10 spellcheck="false"><'+</textarea>';
9533 }
9534 function clearGJLinkArea(){
9535     GJLink_ServerSet.innerHTML = "";
9536     GJLink_Account.innerHTML = "";
9537     GJLink_SendArea.innerHTML = "";
9538     ws0_log_container.innerHTML = "";
9539 }
9540 </script>
9541
9542 <script>
9543 function SetupGJLink(){
9544     setupGJLinkArea();
9545     SetupVisibleText(GJLink_1,gj_serv,'GJLinkSv');
9546     SetupVisibleText(GJLink_1,gj_user,'UserName');
9547     SetupBlinderText(GJLink_1,gj_ukey,'UserKey');
9548     SetupVisibleText(GJLink_1,gj_chan,'ChannelName');

```

```

9549 SetupBlinderText(GJLink_1,gj_ckey,'ChannelKey');
9550 SetupVisibleText(GJLink_1,gj_sendText,'Message');
9551 gj_serv.innerHTML = 'ws://localhost:9999/gjlink1'
9552 }
9553 function GJLink_init(){
9554     SetupGJLink();
9555 }
9556 function iselem(eid){
9557     return document.getElementById(eid);
9558 }
9559 function DestroyGJLink1(){
9560     clearGJLinkArea();
9561     if( !iselem('gj_user') ){
9562         return;
9563     }
9564     if( gj_serv_label.parentNode != gj_user ){
9565         return;
9566     }
9567     if( iselem('gj_serv_label') ) gj_user.parentNode.removeChild(gj_serv_label);
9568     if( iselem('gj_serv') ) gj_user.parentNode.removeChild(gj_serv);
9569     if( iselem('gj_user') ) gj_user.parentNode.removeChild(gj_user);
9570     if( iselem('gj_ukey') ) gj_ukey.parentNode.removeChild(gj_ukey);
9571     if( iselem('gj_chan') ) gj_chan.parentNode.removeChild(gj_chan);
9572     if( iselem('gj_ckey') ) gj_ckey.parentNode.removeChild(gj_ckey);
9573     if( iselem('gj_sendText') ) gj_sendText.parentNode.removeChild(gj_sendText);
9574     if( iselem('ws0_log') ) ws0_log.parentNode.removeChild(ws0_log);
9575 }
9576 DestroyGJLink = DestroyGJLink1;
9577 </script>
9578 </details>
9579 */
9580
9581 /*
9582 <script id="HtmlCodeview-script">
9583     function showNodeAsHtmlSource(otxa,code){
9584         txa = document.createElement('textarea');
9585         txa.id = otxa.id;
9586         txa.setAttribute('class','HtmlCodeviewText');
9587         otxa.parentNode.replaceChild(txa,otxa);
9588         txa.setAttribute('spellcheck','false');
9589         //txa.value = code.innerHTML;
9590         //txa.innerHTML = code.innerHTML;
9591         txa.innerHTML = code.outerHTML;
9592         txa.style.display = "block";
9593         txa.style.width = "100%";
9594         txa.style.height = "300px";
9595     }
9596     function showHtmlCode(otxa,code){
9597         if( event.target.value == 'ShowCode' ){
9598             showNodeAsHtmlSource(otxa,code);
9599             event.target.value = 'HideCode';
9600         }else{
9601             otxa.style.display = "none";
9602             event.target.value = 'ShowCode';
9603         }
9604     }
9605 </script>
9606 <style id="HtmlCodeview-style">
9607     .HtmlCodeviewText {
9608         font-size:10pt;
9609         font-family:Courier New;
9610         white-space:pre;
9611     }
9612     .HtmlCodeViewButton {
9613         padding:2pt !important;
9614         line-height:1.1 !important;
9615         border:2px inset #bbb !important;
9616         font-size:11pt !important;
9617         font-weight:normal !important;
9618         font-family:Georgia !important;
9619         border-radius:3px !important;
9620         color:#ddd; background-color:#228 !important;
9621     }
9622 </style>
9623 */
9624
9625 /*
9626 <details><summary>Live HTML Snapshot</summary>
9627 <span id="LiveHTML">
9628     <!-- ----- HTML Snapshot: Edit, save and load // 2020-0924 SatoxITS -->
9629     <input class="HtmlCodeViewButton" type="button" value="ShowCode" onclick="showLiveHTMLCode()" >
9630     <span id="LiveHTML_Codeview"></span>
9631     <script id="LiveHtmlScript">
9632     function showLiveHTMLCode(){
9633         showHtmlCode(LiveHTML_Codeview,LiveHTML);
9634     }
9635     var _editable = false;
9636     var savSuppressGJShell = false;
9637     function ToggleEditMode(){
9638         _editable = !_editable;
9639         if( _editable ){
9640             savSuppressGJShell = SuppressGJShell;
9641             SuppressGJShell = true;
9642             gsh.setAttribute('contenteditable','true');
9643             GshMenuEdit.innerHTML = 'Lock';
9644             GshMenuEdit.style.color = 'rgba(255,0,0,1)';
9645             GshMenuEdit.style.backgroundColor = 'rgba(255,255,255,1)';
9646         }else{
9647             SuppressGJShell = savSuppressGJShell;
9648             gsh.setAttribute('contenteditable','false');
9649             GshMenuEdit.innerHTML = 'Edit';
9650             GshMenuEdit.style.color = 'rgba(16,160,16,1)';
9651             GshMenuEdit.style.backgroundColor = 'rgba(255,255,255,1)';
9652         }
9653     }
9654     function html_edit(){
9655         ToggleEditMode();
9656     }
9657
9658     // Live HTML (DOM) Snapshot onto browser's localStorage
9659     // 2020-0923 SatoxITS
9660     var htRoot = gsh // -- Element-ID, should be selectable
9661     const snappedHTML = 'SnappedHTML'; // Item-ID of the HTML data in localStogate
9662     // -- should be a [map] of URL
9663     // -- should be with CSSOM as inline script
9664     const htVersionTag = 'VersionTag'; // VesionTag Element-ID in the HTML (in DOM)
9665     function showVersion(note,w,v,u,t){
9666         w.alert(note+' : ' + v + '\n'
9667             + '-- URL: ' + u + '\n'
9668             + '-- Time: ' + t + ' (' + DateLong0(t*1000) + ')');
9669     };
9670 }
9671 function html_save(){
9672     u = document.URL;

```

```

9673     t = new Date().getTime() / 1000;
9674     v = '<'+span id="+htVersionTag+" data-url="+u+" data-time="+t+"'>';
9675     v += '</span>\n';
9676     h += v + htRoot.outerHTML;
9677     localStorage.setItem(snappedHTML,h);
9678     showVersion("Saved",window,v,u,t);
9679 }
9680 function html_load(){
9681     h = localStorage.getItem(snappedHTML);
9682     if( h == null ){
9683         alert('No snapshot taken yet');
9684         return;
9685     }
9686     w = window.open('','');
9687     d = w.document;
9688     d.write(h);
9689     w.focus();
9690     html_ver1("Loaded",w,d);
9691 }
9692 function html_ver1(note,w,d){
9693     if( (v = d.getElementById(htVersionTag)) != null ){
9694         h = v.outerHTML;
9695         u = v.getAttribute('data-url');
9696         t = v.getAttribute('data-time');
9697     }else{
9698         h = 'No version info. in the page';
9699         u = '';
9700         t = 0;
9701     }
9702     showVersion(note,w,v,u,t);
9703 }
9704 function html_ver0(){
9705     html_ver1("Version",window,document);
9706 }
9707 </script>
9708 <!-- LiveHTML -->
9709 </span>
9710 </details>
9711 */
9712
9713 /*
9714 <details><summary>Event sharing</summary>
9715 <span id="EventSharingCodeSpan">
9716
9717 <!-- ----- Event sharing // 2020-0925 SatoxITS { -->
9718
9719 <div id="iftestTemplate" class="iftest" hidden="">
9720 <style> .iftestbody{ color:#f22;font-family:Georgia;font-size:10pt;} </style>
9721 <span id="frameBody" class="iftestbody" onclick="frameClick()"><script>
9722     function docadd(txt){
9723         document.body.append(txt);
9724         window.scrollTo(0,100000);
9725     }
9726     function frameClick(){
9727         xy = 'x'+event.x + ' y'+event.y+'';
9728         //docadd('Got Click on #' +event.target.id+ '+xy+ '\n');
9729         docadd('Got Click on #' +Fid.value+ ', '+xy+ '\n');
9730         window.scrollTo(0,100000);
9731         window.parent.postMessage('OnClick: '+xy, '*');
9732     }
9733     function frameMousemove(){
9734         if( false ){
9735             document.body.append('Mousemove on #' +event.target.id+ ' '
9736                 + 'x'+event.x + ' y'+event.y + '\n');
9737             peerWin = window.frames.iframe1;
9738             document.body.append('Send to peer #' +peerWin+ ' ' + '\n');
9739             window.scrollTo(0,100000);
9740             peerWin.postMessage('Hi!', '*');
9741         }
9742     }
9743     function frameKeydown(){
9744         msg = 'Got Keydown: #' +Fid.value+ ', (' +event.code+ ')';
9745         docadd(msg+ '\n');
9746         window.parent.postMessage(msg, '*');
9747     }
9748     function frameOnMessage(){
9749         docadd('Message ' + event.data + '\n');
9750         window.scrollTo(0,100000);
9751     }
9752     if( document.getElementById('Fid') ){
9753         frameBody.id = Fid.value;
9754         h = '';
9755         h += '<'+style>*{';
9756         h += 'font-size:10pt;white-space:pre-wrap;';
9757         h += 'font-family:Courier New;';
9758         h += '<'+style>';
9759         h += 'I am '+Fid.value+'\n';
9760         document.write(h);
9761         window.addEventListener('click', frameClick);
9762         window.addEventListener('keydown', frameKeydown);
9763         window.addEventListener('message', frameOnMessage);
9764         window.addEventListener('mousemove', frameMousemove);
9765         window.parent.postMessage('Hi parent, I am '+Fid.value, '*');
9766     }
9767 </script></span></div>
9768
9769 <style>.iframeTest{margin:3px;resize:both;width:370px;height:120px;}</style>
9770 <h2>Inter-window communicaiton</h2>
9771 <note>
9772 frame0 >>> frame1 and frame2<br>
9773 frame1 >>> frame0 and frame2<br>
9774 frame2 >>> frame0 and frame1<br>
9775 </note>
9776 <div id="iframe-test">
9777 <pre id="iframeHost" style="border:1px;font-family:Courier New;font-size:10pt;"></pre>
9778 <iframe id="iframe0" title="iframe0" class="iframeTest" style=""></iframe>
9779 <iframe id="iframe1" title="iframe1" class="iframeTest"></iframe>
9780 <iframe id="iframe2" title="iframe2" class="iframeTest"></iframe>
9781 </div>
9782
9783 <script id="if0-test-script">
9784     function InterFrameComm_init(){
9785         setupFrames0();
9786         setupFrames12();
9787     }
9788     function setFrameSrodoc(dst,src){
9789         if( true ){
9790             dst.contentWindow.document.write(src);
9791             // this makes browser waite close, and crash if accumulated !?
9792             // so it should be closed after write
9793             dst.contentWindow.document.close();
9794         }else{
9795             // to be erased before source dump
9796             // but should be set for live snapshot

```

```

9797     dst.srcdoc = src;
9798 }
9799 }
9800 function setupFrames0(){
9801     ibody = iframe0.contentWindow.document.body;
9802     iframe0.style.width = "755px";
9803     //iframeHost.innerHTML = "Message exchange at iframes' host:\n";
9804     window.addEventListener('message',messageFromChild);
9805
9806     if0 = '';
9807     if0 += '<'+pre style="font-family:Courier New;">';
9808     if0 += '<input id="Fid" value="iframe0">';
9809     if0 += iftestTemplate.innerHTML;
9810     setFrameSrcdoc(iframe0,if0);
9811
9812     function clickOnChild(){
9813         console.log('clickOn #' +this.id);
9814     }
9815     function moveOnChild(){
9816         console.log('moveOn #' +this.id);
9817     }
9818     iframe0.contentWindow.document.body.style.setProperty('white-space','pre');
9819     iframe0.contentWindow.document.body.style.setProperty('font-size','9pt');
9820 }
9821 function setupFrames12(){
9822     if1 = '<input id="Fid" value="iframe1">';
9823     if1 += iftestTemplate.innerHTML;
9824     setFrameSrcdoc(iframe1,if1);
9825     //iframe1.name = 'iframe1'; // this seems break contentWindow
9826
9827     if2 = '<input id="Fid" value="iframe2">';
9828     if2 += iftestTemplate.innerHTML;
9829     setFrameSrcdoc(iframe2,if2);
9830
9831     iframe1.addEventListener('message',messageFromChild);
9832     //iframe1.addEventListener('mouseover',moveOnChild);
9833     iframe2.addEventListener('message',messageFromChild);
9834     //iframe2.addEventListener('mouseover',moveOnChild);
9835     iframe1.contentWindow.postMessage(['parent0] Hi iframe1 -- from parent.','*');
9836     //iframe1.contentWindow.postMessage('Your peer is '+iframe2.contentWindow,'*');
9837     iframe2.contentWindow.postMessage(['parent0] Hi iframe2 -- from parent.','*');
9838     //iframe2.contentWindow.postMessage('Your peer is '+iframe1.contentWindow,'*');
9839 }
9840 function messageFromChild(){
9841     from = null;
9842     forw = null;
9843     if( event.source == iframe0.contentWindow ){
9844         from = 'iframe0' ;
9845         forw = 'iframe12';
9846     }else
9847     if( event.source == iframe1.contentWindow ){
9848         from = 'iframe1' ;
9849         forw = 'iframe2';
9850     }else
9851     if( event.source == iframe2.contentWindow ){
9852         from = 'iframe2' ;
9853         forw = 'iframe1';
9854     }else
9855     {
9856         iframeHost.innerHTML += 'Message [unknown] '
9857         + ' orig=' + event.origin
9858         + ' data=' + event.data
9859         //+ ' from=' + event.source
9860         ;
9861     }
9862     msglog1 = from + event.data + ' -- '
9863     + ' from=' + event.source
9864     + ' orig=' + event.origin
9865     + ' name=' + event.source.name
9866     //+ ' port=' + event.ports
9867     //+ ' evid=' + event.lastEventId
9868     + '\n'
9869     ;
9870     if( true ){
9871         if( forw == 'iframe1' || forw == 'iframe12' ){
9872             iframe1.contentWindow.postMessage(from+event.data);
9873         }
9874         if( forw == 'iframe2' || forw == 'iframe12' ){
9875             iframe2.contentWindow.postMessage(from+event.data);
9876         }
9877     }
9878     txtadd0(msglog1);
9879
9880     function txtadd0(txt){
9881         iframe0.contentWindow.document.body.append(txt);
9882         iframe0.contentWindow.scrollTo(0,100000);
9883     }
9884 }
9885 function es_ShowSelf(){
9886     iframe1.setAttribute('src',document.URL);
9887     iframe2.setAttribute('src',document.URL);
9888 }
9889 </script>
9890
9891 <input class="HtmlCodeViewButton" type="button" value="ShowSelf" onclick="es_ShowSelf()">
9892 <input class="HtmlCodeViewButton" type="button" value="ShowCode" onclick="es_showHtmlCode()">
9893 <span id="EventSharingCodeview"></span>
9894 <script id="EventShareingScript">
9895 function es_showHtmlCode(){
9896     showHtmlCode(EventSharingCodeview,EventSharingCodeSpan);
9897 }
9898 DestroyEventSharingCodeview = function(){
9899     //EventSharingCodeview.parentNode.removeChild(EventSharingCodeview);
9900     EventSharingCodeview.innerHTML = "";
9901     iframe0.style = "";
9902     //iframe0.srcdoc = "erased";
9903     //iframe1.srcdoc = "erased";
9904     //iframe2.srcdoc = "erased";
9905 }
9906 </script>
9907 <!-- EventSharing -->
9908 </span>
9909 </details>
9910 */
9911 /*
9912 /*
9913 <!-- ----- "GShell Inside" Notifaction { -->
9914 <script id="script-gshell-inside">
9915 var notices = 0;
9916 function noticeGShellInside(){
9917     ver = '';
9918     if( ver = document.getElementById('GshVersion') ){
9919         ver = ver.innerHTML;
9920     }

```



```

9921 console.log('GJShell Inside (^-^)' + ver);
9922 notices += 1;
9923 if( 2 <= notices ){
9924     document.removeEventListener('mousemove',noticeGShellInside);
9925 }
9926 }
9927 document.addEventListener('mousemove',noticeGShellInside);
9928 noticeGShellInside();
9929
9930 const FooterName = 'GshFooter'
9931 function DestroyFooter(){
9932     if( (footer = document.getElementById(FooterName)) != null ){
9933         //footer.parentNode.removeChild(footer);
9934         empty = document.createElement('div');
9935         empty.id = 'GshFooter0';
9936         footer.parentNode.replaceChild(empty,footer);
9937     }
9938 }
9939 function showFooter(){
9940     footer = document.createElement('div');
9941     footer.id = FooterName;
9942     footer.style.backgroundImage = "url("+ITSmoreQR+")";
9943     //GshFooter0.parentNode.appendChild(footer);
9944     GshFooter0.parentNode.replaceChild(footer,GshFooter0);
9945 }
9946 </script>
9947 <!-- } -->
9948
9949 <!--
9950     border:20px inset #888;
9951 -->
9952
9953 //<span id="WirtualDesktopCodeSpan">
9954 /*
9955 <details id="WirtualDesktopDetails"><summary>Wirtual Desktop</summary>
9956 <!-- ----- Web Wirtual Desktop // 2020-0927 SatoxITS { -->
9957 <style>
9958     .WirtualSpace {
9959         z-index:0;
9960         xwidth:1280px !important; xheight:720px !important;
9961         width:5120px; height:2880px;
9962         border-width:0px;
9963         xbackground-color:rgba(32,32,160,0.8);
9964         xbackground-image:url("WD-WallPaper03.png");
9965         xbackground-size:100% 100%;
9966         color:#22a;font-family:Georgia;font-size:10pt;
9967         xoverflow:scroll;
9968     }
9969     .WirtualGrid {
9970         z-index:0;
9971         position:absolute;
9972         width:800px; height:500px;
9973         border:1px inset #fff;
9974         color:rgba(192,255,192,0.8);
9975         font-family:Georgia, Courier New;
9976         text-align:right;
9977         vertical-align:middle;
9978         font-size:200px;
9979         text-shadow:4px 4px #ccf;
9980     }
9981     .WD_GridScroll {
9982         z-index:100000;
9983         background-color:rgba(200,200,200,0.1);
9984     }
9985     .WirtualDesktop {
9986         z-index:0;
9987         position:relative;
9988         resize:both !important;
9989         overflow:scroll;
9990         display:block;
9991         min-width:120px !important; min-height:60px !important;
9992         width:800px;
9993         height:500px;
9994         border:10px inset #228;
9995         border-width:30px; border-radius:20px;
9996         background-image:url("WD-WallPaper03.png");
9997         background-size:100% 100%;
9998         color:#22a;font-family:Georgia;font-size:10pt;
9999     }
10000 .comment {
10001     // overflow=scroll seems to bound childrens' view in the element span
10002     // specifying overflow seems fix the position of the element
10003 }
10004 .WirtualBrowserSpan {
10005     z-index:10;
10006     xxxborder:0.5px dashed #fff !important;
10007     border-color:rgba(255,255,255,0.5) !important;
10008     position:relative;
10009     left:100px;
10010     top:100px;
10011     display:block;
10012     resize:both !important;
10013     width:540px;
10014     height:320px;
10015     min-width:40px !important; min-height:20px !important;
10016     max-width:5120px !important; max-height:2880px !important;
10017     background-color:rgba(255,200,255,0.1);
10018     xoverflow:scroll;
10019 }
10020 .xWirtualBrowserLocationBar:focus {
10021     color:#f00;
10022     background-color:rgba(255,128,128,0.2);
10023 }
10024 .xWirtualBrowserLocationBar:active {
10025     color:#f00;
10026     background-color:rgba(128,255,128,0.2);
10027 }
10028a .WirtualBrowserLocation {
10029     color:#ccc !important;
10030     text-decoration:none !important;
10031 }
10032a .WirtualBrowserLocation:hover {
10033     color:#fff !important;
10034     text-decoration:underline;
10035 }
10036 .WirtualBrowserLocationBar {
10037     position:absolute;
10038     z-index:100000;
10039     display:block;
10040     width:400px;
10041     height:20px;
10042     padding-left:2px;
10043     line-height:1.1;
10044     vertical-align:middle;

```

```

10045 font-size:14px;
10046 color:#fff;
10047 background-color:rgba(128,128,128,0.2);
10048 font-family:Georgia;
10049}
10050.VirtualBrowserCommandBar {
10051 position:absolute;
10052 z-index:200000;
10053 xxxdisplay:inline;
10054 display:block;
10055 width:60px;
10056 height:20px;
10057 line-height:1.1;
10058 vertical-align:middle;
10059 font-size:14px;
10060 color:#fe4;
10061 background-color:rgba(128,128,128,0.1);
10062 font-family:Georgia;
10063 text-align:left;
10064 left:404px;
10065}
10066.VirtualBrowserFrame {
10067 xxposition:relative;
10068 position:absolute;
10069 xxxdisplay:inline;
10070 display:block;
10071 z-index:10;
10072 resize:both !important;
10073 width:480px; height:240px;
10074 min-width:60px; min-height:30px;
10075 max-width:5120px; max-height:2880px;
10076 border-radius:6px;
10077 background-color:rgba(255,255,255,0.9);
10078 border-top:20px solid;
10079 border-right:4px solid;
10080 border-bottom:10px solid;
10081}
10082.WinFavicon {
10083 width:16px;
10084 height:16px;
10085 margin:1px;
10086 margin-right:3px;
10087 vertical-align:middle;
10088 background-color:rgba(255,255,255,1.0);
10089}
10090.VirtualDesktopMenuBar {
10091 xposition:absolute;
10092 color:#fff;
10093 font-size:7pt;
10094 text-align:right;
10095 padding-right:4px;
10096 background-color:rgba(128,128,128,0.7);
10097}
10098.VirtualDesktopCalender {
10099 color:#fff;
10100 font-size:22pt;
10101 text-align:right;
10102 padding-right:4px;
10103 xbackground-color:rgba(255,255,255,0.2);
10104}
10105.xxxxinput { width:260pt !important; line-height:1.1 !important; margin:1px;
10106 display:inline !important; font-size:10pt !important; padding:1px !important;
10107}
10108.WD_Config {
10109 display:inline !important;
10110 padding:2px !important;
10111 font-size:10pt !important;
10112 width:60pt !important;
10113 height:12pt !important;
10114 line-height:1.0pt !important;
10115 height:15pt !important;
10116}
10117.WD_Button {
10118 display:inline !important;
10119 padding:2px !important;
10120 color:#fff !important;
10121 background-color:#228 !important;
10122 font-size:10pt !important;
10123 width:60pt !important;
10124 height:12pt !important;
10125 line-height:1.0pt !important;
10126 height:16pt !important;
10127 border:2px inset #44a !important;
10128}
10129.WD_Href {
10130 display:inline !important;
10131 padding:2px !important;
10132 font-size:9pt !important;
10133 width:120pt !important;
10134 height:12pt !important;
10135 line-height:1.0pt !important;
10136 height:15pt !important;
10137}
10138}
10139.LiveHtmlCodeviewText {
10140 font-size:10pt;
10141 font-family:Courier New;
10142 xwhite-space:pre;
10143}
10144}
10145.WD_Panel {
10146 x-index:100 !important;
10147 color:#000 !important;
10148 margin-left:25px !important;
10149 width:800px !important;
10150 padding:4px !important;
10151 border:1px solid #888 !important;
10152 border-radius:6px !important;
10153 background-color:rgba(220,220,220,0.9) !important;
10154 font-size:9pt;
10155 font-family:Courier New;
10156}
10157.WD_Help {
10158 font-size:10pt !important;
10159 font-family:Courier New;
10160 line-height:1.2 !important;
10161 color:#000 !important;
10162 width:100% !important;
10163 background-color:rgba(240,240,255,0.8) !important;
10164}
10165}
10166.WB_Zoom {
10167}
10168</style>

```

```

10169<h2>CosmoScreen 0.0.8</h2>
10170<menu>
10171<span id="WD_Help_1" class="WD_Help"><b>ScopeControl command keys</b><br>
10172 g ... grid on/off<br>
10173 i ... zoom in<br>
10174 o ... zoom out<br>
10175 s ... save current scope<br>
10176 r ... restore saved scope<br>
10177</span>
10178</menu>
10179<div class="WD_Panel" draggable="true">
10180<p><!-- should be on the frame of the WD -->
10181Monitor <input id="WD_Resize_1" class="WD_Button" type="button" value="Resize">
10182 < <input id="WD_Width_1" class="WD_Config" type="text">
10183 x <input id="WD_Height_1" class="WD_Config" type="text">
10184 wall-paper: <a href="WD-WallPaper03.png" class="WD_Href" contenteditable="true">WD-WallPaper03.png</a>
10185</p>
10186<p>
10187Desktop <input id="WS_Resize_1" class="WD_Button" type="button" value="Resize">
10188 < <input id="WS_1_Width" class="WD_Config" type="text">
10189 x <input id="WS_1_Height" class="WD_Config" type="text">
10190</p>
10191<p>
10192Display <input id="WD_Zoom_1" class="WD_Button" type="button" value="Zoom">
10193 < <input id="WD_Zoom_1_XY" class="WD_Config" type="text" value="1.0">
10194 x <input id="WD_Zoom_1_MAG" class="WD_Config" type="text" value="1.41421356">
10195 < <input id="WD_Zoom_1_OUT" class="WD_Button" type="button" value="ZoomOut">
10196 < <input id="WD_Zoom_1_IN" class="WD_Button" type="button" value="ZoomIn">
10197</p>
10198<p>
10199Content <input id="WD_Scroll_1" class="WD_Button" type="button" value="Scroll">
10200 X <input id="WD_Left_1" class="WD_Config" type="text" value="0">
10201 Y <input id="WD_Top_1" class="WD_Config" type="text" value="0">
10202shift+wheel for horizontal scroll
10203</p>
10204<p>
10205Scopexx <input id="WD_Zoom_1_Restore" class="WD_Button" type="button" value="Restore">
10206 < <input id="WD_Zoom_1_RestoreScope" class="WD_Config" type="text" value="Scope0">
10207 | <input id="WD_Zoom_1_Save" class="WD_Button" type="button" value="Save">
10208 > <input id="WD_Zoom_1_SaveScope" class="WD_Config" type="text" value="Scope0">
10209</p>
10210<p>
10211Scopeyy <input id="WD_Zoom_1_Forw" class="WD_Button" type="button" value="Forw">
10212 < <input id="WD_Zoom_1_ForwScope" class="WD_Config" type="text" value="Scope0">
10213 | <input id="WD_Zoom_1_Back" class="WD_Button" type="button" value="Back">
10214 > <input id="WD_Zoom_1_BackScope" class="WD_Config" type="text" value="Scope0">
10215</p>
10216<p>
10217Scopezz <input id="WD_Zoom_1_Push" class="WD_Button" type="button" value="Push">
10218 < <input id="WD_Zoom_1_PushScope" class="WD_Config" type="text" value="Scope0">
10219 | <input id="WD_Zoom_1_Pop" class="WD_Button" type="button" value="Pop">
10220 > <input id="WD_Zoom_1_PopScope" class="WD_Config" type="text" value="Scope0">
10221</p>
10222<p>
10223Display <input id="WD_Grid_1" class="WD_Button" type="button" value="GridOn">
10224</p>
10225<p>
10226Overflo <input id="WD_Overflow_1" class="WD_Button" type="button" value="scroll">
10227"scroll" imprisons windows inside the display
10228</p>
10229</div>
10230
10231<div id="WirtualDesktop_1" class="WirtualDesktop" draggable="true" contenteditable="true" style="">
10232<div id="WirtualDesktop_1_MenuBar" class="WirtualDesktopMenuBar" spellcheck="false">
10233<i>CosmoScreen 0.0.8</i><span id="WirtualDesktop_1_Clock"></span>
10234</div>
10235<div id="WirtualDesktop_1_Calender" class="WirtualDesktopCalender" >00:00</div>
10236<div align="right"><h1>WirtualSpace 1.</h1></div>
10237<div id="WirtualDesktop_1_Content" class="WirtualSpace">
10238
10239<div id="WirtualBrowser_1" class="WirtualBrowserSpan" spellcheck="false" draggable="true">
10240<div id="WirtualBrowser_1_Location" class="WirtualBrowserLocationBar"></div>
10241<span id="WirtualBrowser_1_Command" class="WirtualBrowserCommandBar">Reload</span>
10242<iframe id="WirtualBrowser_1_Frame" class="WirtualBrowserFrame" style=""></iframe>
10243</div>
10244
10245<div id="WirtualBrowser_2" class="WirtualBrowserSpan" spellcheck="false" draggable="true">
10246<div id="WirtualBrowser_2_Location" class="WirtualBrowserLocationBar"></div>
10247<span id="WirtualBrowser_2_Command" class="WirtualBrowserCommandBar">Reload</span>
10248<iframe id="WirtualBrowser_2_Frame" class="WirtualBrowserFrame" style=""></iframe>
10249</div>
10250
10251<div id="WirtualBrowser_3" class="WirtualBrowserSpan" spellcheck="false" draggable="true">
10252<div id="WirtualBrowser_3_Location" class="WirtualBrowserLocationBar"></div>
10253<span id="WirtualBrowser_3_Command" class="WirtualBrowserCommandBar">Reload</span>
10254<iframe id="WirtualBrowser_3_Frame" class="WirtualBrowserFrame" style=""></iframe>
10255</div>
10256
10257<div id="WirtualDesktop_1_GridPlane" class="WirtualSpace"></div>
10258</div>
10259</div>
10260
10261<input class="HtmlCodeViewButton" type="button" value="ShowCode" onclick="vd_showHtmlCode()">
10262<span id="WirtualDesktopCodeview"></span>
10263<script id="WirtualDesktopScript">
10264function vd_showHtmlCode(){
10265 codespan = document.getElementById('WirtualDesktopCodeSpan');
10266 showHtmlCode(WirtualDesktopCodeview,codespan);
10267 WirtualDesktopCodeview.setAttribute('class','LiveHtmlCodeviewText');
10268}
10269DestroyEventSharingCodeview = function(){
10270 WirtualDesktopCodeview.innerHTML = "";
10271}
10272
10273function wdlog(log){
10274 if( GJ_Channel != null ){
10275 GJ_SendMessage('WD '+log);
10276 }
10277 console.log(log);
10278}
10279var topMostWin = 10000;
10280function onEnterWin(e){
10281 t = e.target;
10282 oindex = t.style.zIndex;
10283 //if( oindex == '' ) oindex = 0;
10284 //t.saved_zIndex = oindex;
10285 //t.style.zIndex = 10000;
10286 topMostWin += 1;
10287 t.style.zIndex = topMostWin;
10288 nindex = t.style.zIndex;
10289 wdlog('Enter '+t+' #' + t.id + '(' + oindex + '-' + nindex + ')');
10290 e.stopPropagation();
10291 e.preventDefault();
10292}

```

```

10293function onClickWin(e){ // can detect click on the thick border? t = e.target;
10294    oindex = t.style.zIndex;
10295    topMostWin += 1;
10296    t.style.zIndex = topMostWin;
10297    nindex = t.style.zIndex;
10298    wdlog('Click '+t+' #' +t.id+' ('+oindex+'->'+nindex+'');
10299    //e.stopPropagation();
10300    //e.preventDefault();
10301}
10302function onLeaveWin(e){
10303    t = e.target;
10304    //oindex = t.style.zIndex;
10305    //nindex = t.saved_zIndex;
10306    //t.style.zIndex = nindex;
10307    //wdlog('Leave '+e.target+' #' +e.target.id+' ('+oindex+'->'+nindex+'');
10308    e.stopPropagation();
10309    e.preventDefault();
10310}
10311
10312var WinDragstartX; // event
10313var WinDragstartY;
10314var WinDragstartTX; // target
10315var WinDragstartTY;
10316
10317function onWinDragstart(e){
10318    WinDragstartX = e.x;
10319    WinDragstartY = e.y;
10320
10321    t = e.target;
10322
10323    //WinDragstartTX = t.getBoundingClientRect().left.toFixed(0)
10324    //WinDragstartTY = t.getBoundingClientRect().top.toFixed(0)
10325    if( t.style.left == '' ){
10326        WinDragstartTX = x0 = 0;
10327        t.style.left = '0px';
10328    }else{
10329        //WinDragstartTX = x0 = Number(t.style.left);
10330        WinDragstartTX = x0 = parseInt(t.style.left);
10331    }
10332    if( t.style.top == '' ){
10333        WinDragstartTY = y0 = 0;
10334        t.style.top = '0px';
10335    }else{
10336        //WinDragstartTY = y0 = Number(t.style.top);
10337        WinDragstartTY = y0 = parseInt(t.style.top);
10338    }
10339    if( true ){ // to be undo
10340        t.wasAtX = WinDragstartTX;
10341        t.wasAtY = WinDragstartTY;
10342    }
10343    wdlog('DragSTA #' +t.id
10344        + ' event('+e.x+', '+e.y+')'
10345        + ' position=' + t.style.position
10346        + ' style left,top('+t.style.left+', '+t.style.top+')'
10347    );
10348    e.stopPropagation();
10349    //e.preventDefault();
10350    return true;
10351}
10352function onWinDragEvent(wh,e,set,dolog){
10353    t = e.target;
10354    dx = e.x - WinDragstartX;
10355    dy = e.y - WinDragstartY;
10356    nx = WinDragstartTX + dx;
10357    ny = WinDragstartTY + dy;
10358    log = 'Drag'+wh+' #' +t.id
10359        + ' event0('+WinDragstartX+', '+WinDragstartY+')'
10360        + ' event('+e.x+', '+e.y+')'
10361        + ' diff('+dx+', '+dy+')'
10362        + ' (' + nx + ', ' + ny + ')';
10363    + ' (' + t.style.left + ', ' + t.style.top + ')';
10364    + ' wasAt('+ t.wasAtX + ', ' + t.wasAtY + ')';
10365
10366    if( e.x != 0 || e.y != 0 ){
10367        if( set == true ){
10368            //t.style.x = nx + 'px'; // not effective
10369            //t.style.y = ny + 'px'; // not effective
10370            t.style.left = nx + 'px';
10371            t.style.top = ny + 'px';
10372            log += ' Set';
10373        }else{
10374            log += ' NotSet';
10375            if( !dolog ){
10376                log = '';
10377            }
10378        }
10379    }else{
10380        log += ' What?'; // the type is event start?
10381        if( !dolog ){
10382            log = '';
10383        }
10384    }
10385    if( and(dolog, log != '' )){
10386        wdlog(log);
10387    }
10388    if( true ){
10389        // should be propagated to parent in FireFox ?
10390        e.stopPropagation();
10391    }
10392    e.preventDefault();
10393    return false;
10394}
10395function onWinDrag(e){
10396    return onWinDragEvent('Ing',e,true,false);
10397}
10398function onWinDragend(e){
10399    return onWinDragEvent('End',e,false,true);
10400}
10401function onWinDragexit(e){
10402    return onWinDragEvent('Exit',e,false,true);
10403}
10404function onWinDragover(e){
10405    return onWinDragEvent('Over',e,false,true);
10406}
10407function onWinDragenter(e){
10408    return onWinDragEvent('Enter',e,false,true);
10409}
10410function onWinDragleave(e){
10411    return onWinDragEvent('Leave',e,false,true);
10412}
10413function onWinDragdrop(e){
10414    return onWinDragEvent('Drop',e,false,true);
10415}
10416function onFaviconChange(e){

```

```

10417     wdllog('--Favicon #' + e.target.id + ' href=' + e.details.href);
10418 }
10419 var savedSuppressGJShell = false;
10420 function stopGShell(e){
10421     //wdllog('enter Gsh STOP');
10422     savedSuppressGJShell = SuppressGJShell;
10423     SuppressGJShell = true;
10424     e.stopPropagation();
10425     e.preventDefault();
10426 }
10427 function contGShell(e){
10428     //wdllog('leave Gsh STOP');
10429     SuppressGJShell = savedSuppressGJShell;
10430     e.stopPropagation();
10431     e.preventDefault();
10432 }
10433
10434 function WD_onKeyDown(e){
10435     keycode = e.code;
10436     console.log('Keydown #' + e.target.id + ' ' + keycode);
10437     if( keycode == 'KeyG' ){
10438         WD_setGrid1(WD_Grid1);
10439     }else
10440     if( keycode == 'KeyI' ){
10441         WD_doZoomIN();
10442     }else
10443     if( keycode == 'KeyO' ){
10444         WD_doZoomOUT();
10445     }else
10446     if( keycode == 'KeyR' ){
10447         WD_RestoreScope(null);
10448     }else
10449     if( keycode == 'KeyS' ){
10450         WD_SaveScope(null);
10451     }
10452     e.stopPropagation();
10453     e.preventDefault();
10454 }
10455 function WD_onKeyUp(e){
10456     e.stopPropagation();
10457     e.preventDefault();
10458 }
10459 function WD_EventSetup1(){
10460     WirtualDesktop_1.addEventListener('keydown', e => { WD_onKeyDown(e); });
10461     WirtualDesktop_1_Content.addEventListener('keydown', e => { WD_onKeyDown(e); });
10462     WD_Help_1.addEventListener('keydown', e => { WD_onKeyDown(e); });
10463     WD_Help_1.addEventListener('keyup', e => { WD_onKeyUp(e); });
10464 }
10465 function settleWin(s,l,cmd,f,u,w,h,x,y,c,scale){
10466     function WirtualBrowserCommand(e,s,l,cmd,f){
10467         command = cmd.innerHTML
10468         if( command == "Reload" ){
10469             href_id = e.target.href_id;
10470             d = document.getElementById(href_id);
10471             wdllog('href_tag#' + href_id + '\n elem#' + href_id + '\n href=' + d);
10472             url = d.innerHTML;
10473             wdllog('---- Load href_tag#' + href_id + '\n elem#' + href_id + '\n href=' + d
10474                 + '\n url=' + url);
10475             wdllog('---- Load target #' + f.id + ' with url=' + url;
10476                 f.src = url;
10477         }else{
10478             alert('unknown command "' + command + '" ' + e.target.id + ', ' + l.id + ', ' + f.id);
10479         }
10480     }
10481     function onKeyDown(e){
10482         if( e.code == 'Enter' ){
10483             e.stopPropagation();
10484             e.preventDefault();
10485         }
10486     }
10487     function onKeyUp(e){
10488         if( e.code == 'Enter' ){
10489             e.stopPropagation();
10490             e.preventDefault();
10491             // should reload immediately ?
10492         }
10493     }
10494
10495     if( false ){
10496         wdllog('start settle WirtualBrowser url=' + u + '\n'
10497             + 'id=' + s.id + '\n'
10498             + 'width=' + s.style.width + '\n'
10499             + 'height=' + s.style.height
10500         );
10501     }
10502     // very important for WordPress ??
10503     s.style.width = f.style.width = 501; // for WordPress ...??
10504     s.style.height = f.style.height = 271; // for WordPress ...??
10505     if( false ){
10506         wdllog('midway settle WirtualBrowser url=' + u + '\n'
10507             + 'id=' + s.id + '\n'
10508             + 'width=' + s.style.width + '\n'
10509             + 'height=' + s.style.height
10510         );
10511     }
10512     s.width = 502; // for WordPress ...??
10513     s.height = 272; // for WordPress ...??
10514     if( false ){
10515         wdllog('midway-2 settle WirtualBrowser url=' + u + '\n'
10516             + 'id=' + s.id + '\n'
10517             + 'span-width=' + s.width + '\n'
10518             + 'span-height=' + s.height
10519         );
10520     }
10521
10522     s.style.width = w + 'px';
10523     s.style.height = h + 'px';
10524     f.style.width = w + 'px';
10525     f.style.height = h + 'px';
10526     //f.style.setProperty('-webkit-transform','scale('+scale+')');
10527     f.style.setProperty('transform','scale('+scale+')');
10528
10529     //wdllog('--x1-- u=' + u + ' width s=' + s.style.width + ', f=' + f.style.width);
10530     //wdllog('--x2-- u=' + u + ' width s=' + s.style.width + ', f=' + f.style.width);
10531     s.setAttribute('draggable','true')
10532     f.setAttribute('draggable','false'); // why necessary?
10533     l.setAttribute('draggable','false'); // why necessary?
10534     cmd.setAttribute('draggable','false'); // why necessary?
10535     s.addEventListener('dragstart', e => { onWinDragstart(e); });
10536     s.addEventListener('drag', e => { onWinDrag(e); });
10537     s.addEventListener('exit', e => { onWinDragexit(e); });
10538     s.addEventListener('dragend', e => { onWinDragend(e); });
10539     s.addEventListener('dragexit', e => { onWinDragexit(e); });
10540     s.addEventListener('dragenter', e => { onWinDragenter(e); });

```

```

10541 s.addEventListener('dragover', e => { onWinDragover(e); });
10542 s.addEventListener('dragleave', e => { onWinDragleave(e); });
10543 s.addEventListener('drop', e => { onWinDragdrop(e); });
10544
10545 s.addEventListener('mouseenter',e => { onEnterWin(e); });
10546 s.addEventListener('mouseleave',e => { onLeaveWin(e); });
10547
10548 if( false ){
10549     s.style.position = "absolute";
10550     s.style.x = x+'px';
10551     s.style.left = x+'px';
10552     s.style.y = y+'px';
10553     s.style.top = y+'px';
10554 }else{
10555     s.style.setProperty('position','absolute','important');
10556     s.style.setProperty('x',x+'px','important');
10557     s.style.setProperty('left',x+'px','important');
10558     s.style.setProperty('y',y+'px','important');
10559     s.style.setProperty('top',y+'px','important');
10560 }
10561
10562 favicon = './favicon.ico';
10563 uv1 = u.split(':///');
10564 if( 2 <= uv1.length ){
10565     uv2 = uv1[1].split('/');
10566     if( 2 <= uv2.length ){
10567         if( uv1[0] == 'file' ){
10568             //favicon = 'file://' + uv2.slice(0,uv2.length-1).join('/');
10569             // + '/favicon.ico';
10570             favcion = './favicon.ico';
10571         }else{
10572             favicon = uv1[0] + '://' + uv2[0] + '/favicon.ico';
10573         }
10574     }
10575 }
10576 //wlog("---- favicon-url="+favicon);
10577 href_id = l.id + "_href";
10578 l.innerHTML = ''
10579 + '<a id="'+href_id+'" class="VirtualBrowserLocation" href="'+u+'>'+u+'</a>';
10580 //l.addEventListener('click', e => { onClickWin(e); });
10581 l.addEventListener('mouseenter',e => { stopGShell(e); });
10582 l.addEventListener('mouseleave',e => { contGShell(e); });
10583 l.addEventListener('keydown', e => { onKeyDown(e); });
10584 l.addEventListener('keyup', e => { onKeyUp(e); });
10585
10586 cmd.href_id = href_id;
10587 wlog(' (0)cmd=#'+cmd.id);
10588 wlog(' (1)href_id=#'+href_id);
10589 wlog(' (2)href_id=#'+cmd.href_id);
10590 cmd.addEventListener('click', e => { VirtualBrowserCommand(e,s,l,cmd,f); });
10591
10592 f.style.borderColor = c;
10593 f.src = u;
10594 //f.addEventListener('mouseenter',e => { onEnterWin(e); });
10595 //f.addEventListener('mouseleave',e => { onLeaveWin(e); });
10596
10597 //s.addEventListener('click', e => { onClickWin(e); });
10598 //f.addEventListener('click', e => { wlog('click wbl'); });
10599 f.addEventListener('mozbrowsericonchange',onFaviconChange);
10600
10601 wlog('done settle VirtualBrowser url='+u +'\n'
10602 + 'id=' + s.id + ' '
10603 + 'width=' + s.style.width + ' '
10604 + 'height=' + s.style.height + ' '
10605 + 'cmd=' + cmd.id
10606 );
10607 }
10608
10609 function WD_EventSetup2(){
10610     dt = VirtualDesktop_1;
10611     dt.style.width = "800px";
10612     dt.style.height = "500px";
10613     dt.addEventListener('dragstart',e => { onWinDragstart(e); });
10614     dt.addEventListener('drag', e => { onWinDrag(e); });
10615     dt.addEventListener('exit', e => { onWinDragexit(e); });
10616 }
10617
10618 function GRonClick(){
10619     WD_SaveScope(null); // should be push
10620     t = event.target;
10621     x = t.getAttribute('data-leftx');
10622     y = t.getAttribute('data-topy');
10623     zoom = WD_Zoom_1_XY.value;
10624     x *= zoom;
10625     y *= zoom;
10626     WD_doScrollXY(event,x,y);
10627     //alert('scroll #'+t.id+' x='+tx+', y='+y);
10628 }
10629 function WD_setGrid(e){
10630     t = e.target;
10631     WD_setGrid1(t);
10632 }
10633 function WD_setGrid(t){
10634     //ds = VirtualDesktop_1_Content; // should be VirtualSpace_1
10635     ds = VirtualDesktop_1_GridPlane;
10636     if( t.value == 'GridOn' ){
10637         for( col = 0; col < 16; col++ ){
10638             for( row = 0; row < 16; row++ ){
10639                 g1 = document.createElement('span');
10640                 g1.setAttribute('class','VirtualGrid');
10641                 leftx = col * 800;
10642                 topy = row * 500;
10643                 gid = col + '.' + row
10644                 label = '<'+'span '
10645                     + 'id="'+gid+'"' + 'class="WD_GridScroll" '
10646                     + 'contenteditable="false" onclick="GRonClick()" '
10647                     + 'data-leftx=' + leftx + ' ' + 'data-topy=' + topy + ' '
10648                     + '>'
10649                 + gid + '<'+'span>';
10650                 console.log('grid '+label);
10651                 g1.innerHTML = label;
10652                 g1.position = 'relative';
10653                 g1.leftx = leftx;
10654                 g1.topy = topy;
10655                 g1.style.left = g1.leftx + 'px';
10656                 g1.style.top = g1.topy + 'px';
10657                 if( col % 2 == row % 2 ){
10658                     g1.style.backgroundColor = 'rgba(255,255,255,0.3)';
10659                 }
10660                 ds.appendChild(g1);
10661             }
10662         }
10663         t.value = 'GridOff';
10664     }else{

```

```

10665     ds.innerHTML = '';
10666     t.value = 'GridOn';
10667 }
10668 }
10669 }
10670 var sav_scrollLeft;
10671 var sav_scrollTop;
10672 var sav_nscale;
10673 function WD_SaveScope(e){
10674     sav_scrollLeft = WD_Left_1.value;
10675     sav_scrollTop = WD_Top_1.value;
10676     sav_nscale = WD_Zoom_1_XY.value;
10677     //console.log('saved zoom='+sav_oscale+','+sav_nscale);
10678 }
10679 function WD_RestoreScope(e){
10680     WD_Zoom_1_XY.value = sav_nscale;
10681     WD_doZoom();
10682 }
10683     WD_Left_1.value = sav_scrollLeft;
10684     WD_Top_1.value = sav_scrollTop;
10685     WD_doScroll(null);
10686 }
10687 function ignoreEvent(e){
10688     e.stopPropagation();
10689     //e.preventDefault();
10690 }
10691 function zoomMag(){
10692     return WD_Zoom_1_MAG.value;
10693 }
10694 function WD_EventSetup3(){
10695     WD_Grid_1.addEventListener('click', e => { WD_setGrid(e); });
10696     WD_Zoom_1_Save.addEventListener('click', e => { WD_SaveScope(e); });
10697     WD_Zoom_1_Restore.addEventListener('click', e => { WD_RestoreScope(e); });
10698     WD_Width_1.value = dt.style.width;
10699     WD_Width_1.addEventListener('keydown', ignoreEvent);
10700     WD_Width_1.addEventListener('keyup', ignoreEvent);
10701     WD_Height_1.value = dt.style.height;
10702     WD_Height_1.addEventListener('keydown', ignoreEvent);
10703     WD_Height_1.addEventListener('keyup', ignoreEvent);
10704     WD_Zoom_1_MAG.addEventListener('keydown', ignoreEvent);
10705     WD_Zoom_1_MAG.addEventListener('keyup', ignoreEvent);
10706 }
10707
10708 function escale1(e,oscale,nscale){
10709     e.style.setProperty('transform','scale('+nscale+')');
10710     rscale = oscale / nscale;
10711     w = parseInt(e.style.width);
10712     h = parseInt(e.style.height);
10713     w = w * rscale; //(oscale/nscale);
10714     h = h * rscale; //(oscale/nscale);
10715     e.style.width = w + 'px';
10716     e.style.height = h + 'px';
10717 }
10718 function scaleWD(ds,oscale,nscale){
10719     if( true ){
10720         escale1(WirtualBrowser_1,oscale,nscale);
10721         escale1(WirtualBrowser_1_Location,oscale,nscale);
10722         escale1(WirtualBrowser_1_Command,oscale,nscale);
10723         escale1(WirtualBrowser_1_Frame,oscale,nscale);
10724 }
10725     escale1(WirtualBrowser_2,oscale,nscale);
10726     escale1(WirtualBrowser_2_Location,oscale,nscale);
10727     escale1(WirtualBrowser_2_Command,oscale,nscale);
10728     escale1(WirtualBrowser_2_Frame,oscale,nscale);
10729 }
10730     escale1(WirtualBrowser_3,oscale,nscale);
10731     escale1(WirtualBrowser_3_Location,oscale,nscale);
10732     escale1(WirtualBrowser_3_Command,oscale,nscale);
10733     escale1(WirtualBrowser_3_Frame,oscale,nscale);
10734 }
10735 }
10736 function WD_doZoom(){
10737     ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10738     oscale = WD_Zoom_1_XY.ovalue; // hidden value for current zoom
10739     nscale = WD_Zoom_1_XY.value;
10740     ds.style.zoom = nscale;
10741     WD_Zoom_1_XY.value = ds.style.zoom;
10742     WD_Zoom_1_XY.ovalue = ds.style.zoom;
10743     scaleWD(ds,oscale,nscale);
10744 }
10745 function WD_EventSetup4(){
10746     WD_Zoom_1.addEventListener('click',WD_doZoom);
10747     WD_Zoom_1_XY.addEventListener('keydown',ignoreEvent);
10748     WD_Zoom_1_XY.addEventListener('keyup',ignoreEvent);
10749 }
10750
10751 function WD_doZoomOUT(){
10752     ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10753     oscale = WD_Zoom_1_XY.value;
10754     if( oscale == 0 || oscale == '' ){
10755         oscale = 1;
10756     }
10757     nscale = oscale / zoomMag();
10758     ds.style.zoom = nscale;
10759     WD_Zoom_1_XY.value = ds.style.zoom;
10760     WD_Zoom_1_XY.ovalue = ds.style.zoom;
10761     scaleWD(ds,oscale,nscale);
10762 }
10763 function WD_doZoomIN(){
10764     ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10765     oscale = WD_Zoom_1_XY.value;
10766     if( oscale == 0 || oscale == '' ){
10767         oscale = 1;
10768     }
10769     nscale = oscale * zoomMag();
10770     if( 4 < nscale ){
10771         alert('maybe too large, zoom='+nscale);
10772         return;
10773     }
10774     ds.style.zoom = nscale;
10775     WD_Zoom_1_XY.value = ds.style.zoom;
10776     WD_Zoom_1_XY.ovalue = ds.style.zoom;
10777     scaleWD(ds,oscale,nscale);
10778 }
10779 function WD_EventSetup5(){
10780     WD_Zoom_1_OUT.addEventListener('click',WD_doZoomOUT);
10781     WD_Zoom_1_IN.addEventListener('click',WD_doZoomIN);
10782 }
10783
10784 function WD_doResize(e){
10785     dt = WirtualDesktop_1;
10786     dt.style.width = WD_Width_1.value;
10787     dt.style.height = WD_Height_1.value;
10788     WD_Width_1.value = dt.style.width;

```

```

10789   WD_Height_1.value = dt.style.height;
10790}
10791WD_Resize_1.addEventListener('click',e => { WD_doResize(e); });
10792
10793
10794function WD_doRSResize(e){
10795   //alert('Resize Space');
10796   ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10797   ds.style.width = WS_1_Width.value;
10798   ds.style.height = WS_1_Height.value;
10799   WS_1_Width.value = ds.style.width;
10800   WS_1_Height.value = ds.style.height;
10801}
10802function WD_EventSetup6(){
10803   ds = WirtualDesktop_1_Content; // should be WirtualSpace_1
10804   ds.style.width = '5120px';
10805   ds.style.height = '2880px';
10806   WS_1_Width.value = ds.style.width;
10807   WS_1_Height.value = ds.style.height;
10808   WS_1_Width.addEventListener('keydown',ignoreEvent);
10809   WS_1_Width.addEventListener('keyup',ignoreEvent);
10810   WS_1_Height.addEventListener('keydown',ignoreEvent);
10811   WS_1_Height.addEventListener('keyup',ignoreEvent);
10812   WS_Resize_1.addEventListener('click',e => { WD_doRSResize(e); });
10813}
10814
10815function WD_doScrollXY(e,sleft,stop){
10816   dt = WirtualDesktop_1;
10817   dt.scrollLeft = sleft;
10818   dt.scrollTop = stop;
10819   WD_Left_1.value = dt.scrollLeft;
10820   WD_Top_1.value = dt.scrollTop;
10821   console.log('--Scroll #' + dt.id + ' (' + sleft + ', ' + stop + ')');
10822}
10823function WD_doScroll(e){
10824   //dt = WirtualDesktop_1_Content;
10825   dt = WirtualDesktop_1;
10826   sleft = parseInt(WD_Left_1.value);
10827   stop = parseInt(WD_Top_1.value);
10828   dt.scrollLeft = sleft;
10829   dt.scrollTop = stop;
10830   WD_Left_1.value = dt.scrollLeft;
10831   WD_Top_1.value = dt.scrollTop;
10832   console.log('--Scroll #' + dt.id + ' (' + sleft + ', ' + stop + ')');
10833}
10834function showScrollPosition(){
10835   if( false )
10836     console.log(
10837       'wstop=' + WirtualDesktop_1.style.top + ', ' +
10838       'wsx=' + WirtualDesktop_1.style.y + ', ' +
10839       'wss=' + WirtualDesktop_1.scrollTop + ', ' +
10840       'wdtop=' + WirtualDesktop_1_Content.style.top + ', ' +
10841       'wdx=' + WirtualDesktop_1_Content.style.y + ', ' +
10842       'wds=' + WirtualDesktop_1_Content.scrollTop + ', '
10843     );
10844   WD_Left_1.value = WirtualDesktop_1.scrollLeft;
10845   WD_Top_1.value = WirtualDesktop_1.scrollTop;
10846}
10847function WD_EventSetup7(){
10848   WD_Scroll_1.addEventListener('click',e => { WD_doScroll(e); });
10849   WD_Left_1.addEventListener('keydown',ignoreEvent);
10850   WD_Left_1.addEventListener('keyup',ignoreEvent);
10851   WD_Top_1.addEventListener('keydown',ignoreEvent);
10852   WD_Top_1.addEventListener('keyup',ignoreEvent);
10853}
10854function WD_EventSetup8(){
10855   WirtualDesktop_1.addEventListener('scroll',showScrollPosition);
10856   WirtualDesktop_1_Content.addEventListener('scroll',showScrollPosition);
10857}
10858
10859if( false ){
10860   w = 1000 + 'px';
10861   dt.style.width = w;
10862   dt.style.height = "300px";
10863   dt.style.resize = 'both';
10864   dt.style.borderWidth = 50 + 'px';
10865   dt.style.borderRadius = 25 + 'px';
10866   console.log('--2----- #' + dt.id + ' style=' + dt.style);
10867   console.log('----- #' + dt.id + ' width=' + dt.style.width);
10868   console.log('----- #' + dt.id + ' left=' + dt.style.left);
10869   console.log('----- #' + dt.id + ' border=' + dt.style.border);
10870}
10871function onDTRresize(e){
10872   dt = e.target;
10873   h = parseInt(dt.style.height);
10874   dt.style.borderWidth = (h * 0.075) + 'px';
10875   console.log('----- borderWidgh=' + dt.style.borderWidth);
10876}
10877
10878WirtualDesktopDetails.addEventListener('toggle',WirtualDesktop_init);
10879function WirtualDesktop_init(){
10880   if( !WirtualDesktopDetails.open ){
10881     return;
10882   }
10883   //GJ_Join();
10884   WirtualDesktop_1.addEventListener('resize', e => { onDTRresize(e); });
10885   //console.log('----- #' + dt.id
10886   // + ' borderWidth=' + dt.style.getProperty('border-width'));
10887
10888   settleWin(
10889     WirtualBrowser_1,
10890     WirtualBrowser_1_Location,
10891     WirtualBrowser_1_Command,
10892     WirtualBrowser_1_Frame,
10893     document.URL,
10894     500,280,50,20,'#262',1.0);
10895   settleWin(
10896     WirtualBrowser_2,
10897     WirtualBrowser_2_Location,
10898     WirtualBrowser_2_Command,
10899     WirtualBrowser_2_Frame,
10900     'https://its-more.jp/ja_jp/',
10901     500,280,150,100,'#448',1.0);
10902   settleWin(
10903     WirtualBrowser_3,
10904     WirtualBrowser_3_Location,
10905     WirtualBrowser_3_Command,
10906     WirtualBrowser_3_Frame,
10907     '// ../gshell/gsh.go.html',
10908     '// http://gshell.org/gshell/gsh.go.html',
10909     'https://golang.org',
10910     500,280,250,180,'#444',1.0);
10911     //1000,720,0,0,'#444',0.125);
10912     //1200,720,-100,-50,'#444',0.4);

```



```

10913 function WD_ClockUpdate(e){
10914     WirtualDesktop_1_Clock.innerHTML = DateShort();
10915     WirtualDesktop_1_Calender.innerHTML = DateHourMin();
10916 }
10917 window.setInterval(WD_ClockUpdate,500);
10918
10919 WD_EventSetup1();
10920 WD_EventSetup2();
10921 WD_EventSetup3();
10922 WD_EventSetup4();
10923 WD_EventSetup5();
10924 WD_EventSetup6();
10925 WD_EventSetup7();
10926 WD_EventSetup8();
10927}
10928//WirtualDesktop_init();
10929
10930Destroy_WirtualDesktop = function(){
10931     WirtualDesktop_1.style = "";
10932
10933     WirtualBrowser_1.removeAttribute('style');
10934     WirtualBrowser_1_Location.innerHTML = '';
10935     WirtualBrowser_1_Frame.removeAttribute('src');
10936     WirtualBrowser_1_Frame.removeAttribute('style');
10937     WirtualBrowser_1_Frame.style="";
10938
10939     WirtualBrowser_2.removeAttribute('style');
10940     WirtualBrowser_2_Location.innerHTML = '';
10941     WirtualBrowser_2_Frame.removeAttribute('src');
10942     WirtualBrowser_2_Frame.style="";
10943
10944     WirtualBrowser_3.removeAttribute('style');
10945     WirtualBrowser_3_Location.innerHTML = '';
10946     WirtualBrowser_3_Frame.removeAttribute('src');
10947     WirtualBrowser_3_Frame.style="";
10948
10949     GJFactory_1.style = "";
10950     iframe0.style = "";
10951     WirtualDesktop_1.style = "";
10952}
10953
10954</script>
10955<!-- WirtualDesktop -->
10956<details>
10957* //</span>
10958
10959//<!-- ===== Work { ===== -->
10960<span id="SBSidebar_WorkCodeSpan">
10961/*
10962<details><summary>SBSidebar</summary>
10963<!-- ----- SBSidebar // 2020-0928 SatoxITS { -->
10964<h2>SBSidebar</h2>
10965<input id="SBSidebar_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
10966<input id="SBSidebar_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
10967<input id="SBSidebar_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
10968<span id="SBSidebar_WorkCodeView"></span>
10969<script id="SBSidebar_WorkScript">
10970function SBSidebar_openWorkCodeView(){
10971     function SBSidebar_showWorkCode(){
10972         showHtmlCode(SBSidebar_WorkCodeView,SBSidebar_WorkCodeSpan);
10973     }
10974     SBSidebar_WorkCodeViewOpen.addEventListener('click',SBSidebar_showWorkCode);
10975}
10976SBSidebar_openWorkCodeView(); // should be invoked by an event
10977
10978console.log('-- Sbslider // 2020-1006-01 SatoxITS --');
10979function SetSidebar(){
10980     sidebar = document.getElementById('secondary');
10981//     console.log('primary='+primary+ ' + secondary='+sidebar+ ' + main='+main+ ' ');
10982     wrap = sidebar.parentNode;
10983     console.log('-- Sbslider parent is '+wrap+', #' +wrap.id+' .' +wrap.class);
10984//wrap = wrap.parentNode;
10985//console.log('-- Sbslider parent is '+wrap+', #' +wrap.id+' .' +wrap.class);
10986//wrap = wrap.parentNode;
10987//console.log('-- Sbslider parent is '+wrap+', #' +wrap.id+' .' +wrap.class);
10988//nsb = sidebar.cloneNode();
10989     nsb = sidebar;
10990     nsb.style.width = '100%';
10991     slider = document.createElement('div');
10992     slider.id = 'Sbslider';
10993     slider.appendChild(nsb);
10994     slider.setAttribute('class','Sbslider');
10995     nsb.style.position = 'relative';
10996     slider.style.position = 'fixed';
10997     slider.style.display = 'block'; //inline;
10998     slider.style.zIndex = 100000;
10999     // nsb.style.zIndex = 200000;
11000     nsb.style.position = 'absolute';
11001     nsb.style.minWidth = '80px';
11002     nsb.style.left = '0px';
11003     nsb.style.top = '0px';
11004
11005     w = window.innerWidth;
11006     console.log('SliderWidth '+w+: ' ,(w/3)+px');
11007     if( w < 640 ){
11008         slider.style.setProperty('width',(w/3) + 'px','important');
11009     }
11010     main.style.marginLeft = (parseInt(slider.style.width)/2 - 20) + 'px';
11011
11012     slider.style.resize = "both";
11013     slider.draggable = "true";
11014     wrap.appendChild(slider);
11015     console.log('-- added Sbslider');
11016     //nsb.addEventListener('scroll',SbScrolled);
11017
11018     buttons = document.createElement('div');
11019     buttons.id = 'NaviButtons';
11020     buttons.setAttribute('class','NaviButtons');
11021     buttons.align = "center";
11022     buttons.innerHTML += '<'+><a href="#TopOfPost" draggable="true">TOP<'+>/a><'+>/p>';
11023     buttons.innerHTML += '<'+><a href="#EndOfPost" draggable="true">END<'+>/p>';
11024     page.appendChild(buttons);
11025     buttons.style.position = 'fixed';
11026     buttons.style.zIndex = 30000;
11027     buttons.style.width = '180%';
11028     buttons.style.top = '320px';
11029     buttons.style.left = parseInt(w) * 1.0 + 'px';
11030     console.log('-- Sbslider installed (^-^)/ SatoxITS');
11031}
11032//window.addEventListener('load',SetSidebar); // after load
11033DestroyNaviButtons = function(){
11034     nb = document.getElementById('NaviButtons');
11035     if( nb != null ){
11036         nb.parentNode.removeChild(NaviButtons);

```

```
11037 }
11038 }
11039 </script>
11040
11041 // 2020-1006 its-more.jp-blog-60000-style.css
11042 <!-- {
11043 <style>
11044 #NaviButtons {
11045     position:fixed;
11046     display:block;
11047     xwidth:100%;
11048     xtop:100px;
11049     xxleft:10px;
11050     z-index:30000;
11051     font-size:10pt;
11052     color:#2ff !important;
11053     text-align:center;
11054     background-color:rgba(230,230,230,0.01);
11055 }
11056 #NaviButtons a {
11057     color:#2a2 !important;
11058     font-size:20px;
11059     text-align:center;
11060     xtext-shadow:2px 2px #8ff;
11061     resize:both;
11062     padding:6px;
11063     margin:10px;
11064     border:1px solid #288 !important;
11065     border-radius:3px;
11066     background-color:rgba(160,160,160,0.05);
11067 }
11068 #SbSlider {
11069     overflow:auto;
11070     resize:both !important;
11071     xxoverflow-y:hidden !important;
11072     height:100px !important;
11073     display:inline !important;
11074     position:fixed !important;
11075     left:0px;
11076     top:0px;
11077     xxwidth:180px;
11078     width:24%;
11079     min-width:80px;
11080     height:100% !important;
11081     background-color:rgba(100,100,200,0.1);
11082 }
11083 #secondary {
11084     position:fixed;
11085     left:0px;
11086     top:0px;
11087     xxxz-index:60000;
11088     scroll-behavior: overflow !important;
11089     xxxxxxxxxxxxxxxxxxxxxxxxxxxwidth:18% !important;
11090     padding-left:4pt;
11091     color:#fff;
11092     font-size:0.5em;
11093     background-color:rgba(64,160,64,0.6) !important;
11094     white-space:nowrap;
11095 }
11096 #secondary a {
11097     color:#fff !important;
11098     text-decoration:disable !important;
11099 }
11100 #primary {
11101     position:relative;
11102     width:75% !important;
11103     left:25% !important;
11104 }
11105 #main {
11106     position:relative;
11107     width:75% !important;
11108     left:25% !important;
11109 }
11110 #site-navigation {
11111     position:relative;
11112     left:120px;
11113 }
11114 #adswsc_countertext {
11115     color:#4169e1;
11116     font-size:16pt !important;
11117     xxfont-size:10% !important;
11118     font-weight:bold;
11119 }
11120 #nowTime {
11121     color:#a0ffa0;
11122     font-size:16pt !important;
11123     xxfont-size:10% !important;
11124     font-weight:bold;
11125     text-shadow:1px 1px #fff;
11126 }
11127 .navigation-top {
11128     color:#22a !important;
11129     border:0px;
11130     background-color:rgba(220,220,220,0.1);
11131 }
11132
11133 .visible-scrollbar, .invisible-scrollbar, .mostly-customized-scrollbar {
11134     display: block;
11135     xxwidth: 1em;
11136     xxoverflow: auto;
11137     xxheight: 1em;
11138 }
11139 .invisible-scrollbar::-webkit-scrollbar {
11140     xdisplay: none;
11141     width:1px !important;
11142     height:1px !important;
11143 }
11144 .mostly-customized-scrollbar::-webkit-scrollbar {
11145     width: 2px;
11146     height: 2px;
11147     xxbackground-color: #aaa; xxx:or add it to the track;
11148 }
11149 .mostly-customized-scrollbar::-webkit-scrollbar-thumb {
11150     background: #000;
11151 }
11152 </style>
11153 } -->
11154
11155
11156 </details>
11157 <!-- SBSidebar_WorkCodeSpan } -->
11158 *//</span>
11159 //<!-- ===== Work } ===== -->
11160
```

```

11161
11162//<!-- ===== Work { ===== -->
11163//<span id="Affiliate_WorkCodeSpan">
11164/*
11165<div id="AffViewDock">
11166<div id="AffView" class="AffView" draggable="true" style="">
11167<div id="AffiSet" class="AffPlate">
11168 <iframe id="aff_0" class="AffItem"></iframe>
11169 <iframe id="aff_1" class="AffItem"></iframe>
11170 <iframe id="aff_2" class="AffItem"></iframe>
11171 <iframe id="aff_3" class="AffItem"></iframe>
11172 <iframe id="aff_4" class="AffItem"></iframe>
11173 <iframe id="aff_5" class="AffItem"></iframe>
11174</div>
11175</div></div>
11176<details><summary>Affiliate</summary>
11177<!-- ----- Affiliate // 2020-1010 SatoxITS { -->
11178<h2>Supportive Affiliate</h2>
11179<style>
11180.AffView {
11181  z-index:1000;
11182  overflow-x:scroll;
11183  overflow-y:scroll;
11184  position:fixed;
11185  max-width:2560px;
11186  max-height:100%;
11187  width:270px;
11188  left:75%;
11189  height:95%;
11190  resize:both;
11191  xleft:-10%;
11192  margin-top:40px;
11193  xleft:0;
11194  xxalign:right;
11195  display:block;
11196  border:4px inset rgba(255,255,255,0.1);
11197  background-color:rgba(255,255,255.0.1);
11198}
11199.AffView:hover {
1200  z-index:10000000;
1201  width:300px;
1202  overflow:scroll;
1203  border:4px inset #ffc;
1204  background-color:rgba(80,80,255.0.5);
1205  background-color:#ffc;
1206}
1207.AffPlate:hover {
1208  border-left:4px dashed #888;
1209}
1210.AffPlate{
1211  overflow-x:visible;
1212  border-left:4px dashed rgba(255,255,255,0.1);
1213  max-width:2560px;
1214  max-height:2880px;
1215  margin-top:10px;
1216  margin-bottom:10px;
1217  margin-left:4px;
1218  width:300px;
1219  xheight:1440px;
1220}
1221.AffItem {
1222  overflow-x:visible;
1223  xoverflow-y:scroll;
1224  max-width:2560px;
1225  max-height:1440px;
1226  z-index:1000000;
1227  display:block;
1228  xposition:fixed;
1229  xposition:absolute;
1230  position:relative;
1231  //left:300px;
1232  xresize:both;
1233  padding:0px;
1234  width:600px;
1235  height:400px;
1236  max-height:800px !important;
1237  margin-top:0%;
1238  margin-left:0%;
1239  margin-right:0% !important;
1240  border:16px inset #ccc;
1241  transform:scale(0.5);
1242  background-color:rgba(255,255,255,0.8);
1243  xxalign:right;
1244}
1245.AffItem:hover {
1246  border:16px inset #bbf;
1247}
1248</style>
1249<input id="Affiliate_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
1250<input id="Affiliate_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
1251<input id="Affiliate_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
1252<span id="Affiliate_WorkCodeView"></span>
1253<script id="Affiliate_WorkScript">
1254function Affiliate_openWorkCodeView(){
1255  function Affiliate_showWorkCode(){
1256    showHtmlCode(Affiliate_WorkCodeView,Affiliate_WorkCodeSpan);
1257  }
1258  Affiliate_WorkCodeViewOpen.addEventListener('click',Affiliate_showWorkCode);
1259}
1260Affiliate_openWorkCodeView(); // should be invoked by an event
1261
1262<<iframe id="aff_8" xxsrc="https://stackoverflow.com/tags" class="AffItem"></iframe>
1263<<iframe id="aff_9" xxsrc="https://developer.mozilla.org/en-US/docs/Web" class="AffItem"></iframe>
1264function affsetup(){
1265  parent = document.documentElement;
1266  parent.appendChild(AffView);
1267  AffView.style.top = '0px';
1268  AffView.style.left = (window.innerWidth - 280) + 'px';
1269
1270  var off = 100;
1271  zoom = 0.5;
1272  ozoom = 0.3;
1273  leftx = window.innerWidth - 300;
1274  left = leftx + 'px';
1275left = -130 + 'px';
1276  console.log('aff-init window.innerWidth='+window.innerWidth);
1277  w = 1000;
1278  h = 560;
1279
1280  aff_0.src="/gshell/gsh.go.html";
1281  aff_1.src="https://golang.org";
1282  aff_2.src="https://drafts.csswg.org/";
1283  aff_3.src="https://html.spec.whatwg.org/dev/";
1284  aff_4.src="https://wikipedia.org";

```

```

11285 aff_5.src="https://www.bing.com/translator";
11286
11287 //parent.appendChild(aff_0);
11288 aff_0.style.width = zoom*w+'px';
11289 aff_0.style.height = zoom*h+'px';
11290 aff_0.style.left = left;
11291 //aff_0.style.top = off+'px'; off += ozoom*h;
11292 aff_0.draggable = 'true';
11293
11294 //parent.appendChild(aff_1);
11295 aff_1.style.width = zoom*w+'px';
11296 aff_1.style.height = zoom*h+'px';
11297 aff_1.style.left = left;
11298 //aff_1.style.top = off+'px'; off += ozoom*h;
11299 aff_1.style.top = '-150px';
11300 aff_1.draggable = 'true';
11301
11302 //parent.appendChild(aff_2);
11303 aff_2.style.width = zoom*w+'px';
11304 aff_2.style.height = zoom*h+'px';
11305 aff_2.style.left = left;
11306 //aff_2.style.top = off+'px'; off += ozoom*h;
11307 aff_2.style.top = '-300px';
11308 aff_2.draggable = 'true';
11309
11310 //parent.appendChild(aff_3);
11311 aff_3.style.transform = 'scale(0.25)';
11312 aff_3.style.width = 2*zoom*w+'px';
11313 aff_3.style.height = 2*zoom*h+'px';
11314 aff_3.style.border = '32px inset #ccc';
11315 //aff_3.style.left = -390 + 'px'; //left*2;
11316 //aff_3.style.left = (leftx - 265) + 'px';
11317 aff_3.style.left = -395 + 'px';
11318 //aff_3.style.top = (-155+off)+'px'; off += ozoom*h;
11319 aff_3.style.top = '-600px';
11320 aff_3.draggable = 'true';
11321
11322 //parent.appendChild(aff_4);
11323 aff_4.style.width = zoom*w+'px';
11324 aff_4.style.height = zoom*h+'px';
11325 aff_4.style.left = left;
11326 //aff_4.style.top = off+'px'; off += ozoom*h;
11327 aff_4.style.top = '-900px';
11328 aff_4.draggable = 'true';
11329
11330 //parent.appendChild(aff_5);
11331 aff_5.style.transform = 'scale(0.300)';
11332 aff_5.style.width = zoom*(w*1.67)+'px';
11333 aff_5.style.height = zoom*(h*1.67)+'px';
11334 aff_5.style.border = '25px inset #ccc';
11335 aff_5.style.left = -308+'px';
11336 //aff_5.style.left = (-175+leftx)+'px';
11337 //aff_5.style.top = (-95+off)+'px'; off += ozoom*h;
11338 //aff_5.style.left = '0px';
11339 //aff_5.style.top = '0px';
11340 aff_5.style.top = '-1150px';
11341 //aff_5.style.align = 'right';
11342 aff_5.draggable = 'true';
11343
11344function affresize(){
11345 AffView.style.left = (window.innerWidth - 280) + 'px';
11346 leftx = window.innerWidth - 400;
11347 left = leftx + 'px';
11348 console.log('aff-resize window.innerWidth='+window.innerWidth);
11349}
11350window.addEventListener('resize',affresize);
11351//document.addEventListener('resize',affresize);
11352//gsh.addEventListener('resize',affresize);
11353
11354function ResetAffView(){
11355 AffViewDock.appendChild(AffView);
11356 AffView.removeAttribute('style');
11357 aff_0.removeAttribute('src');
11358 aff_0.removeAttribute('style'); aff_0.removeAttribute('draggable');
11359 aff_1.removeAttribute('src');
11360 aff_1.removeAttribute('style'); aff_1.removeAttribute('draggable');
11361 aff_2.removeAttribute('src');
11362 aff_2.removeAttribute('style'); aff_2.removeAttribute('draggable');
11363 aff_3.removeAttribute('src');
11364 aff_3.removeAttribute('style'); aff_3.removeAttribute('draggable');
11365 aff_4.removeAttribute('src');
11366 aff_4.removeAttribute('style'); aff_4.removeAttribute('draggable');
11367 aff_5.removeAttribute('src');
11368 aff_5.removeAttribute('style'); aff_5.removeAttribute('draggable');
11369}
11370</script>
11371</details>
11372<!-- Affiliate_WorkCodeSpan } -->
11373*/ </span>
11374<!-- ===== Work } ===== -->
11375
11376
11377
11378<!-- ===== Work { ===== -->
11379</span id="Shading_WorkCodeSpan">
11380/*
11381<details open=""><summary>Shading Canvas</summary>
11382<!-- ----- Shading Canvas // 2020-1011 SatoxITS { -->
11383<h2>Shading Canvas</h2>
11384<note>
11385<b>Commands</b><br>
11386Placement Mode<br>
11387a ... apply (into absolute position)<br>
11388j ... bring down (ArrowDown)<br>
11389k ... bring up (ArrowUp)<br>
11390h ... bring left (ArrowLeft)<br>
11391l ... bring right (ArrowRight)<br>
11392o ... z-index = 0<br>
11393+ ... z-index += 1<br>
11394- ... z-index -= 1<br>
11395r ... return to here (relative position)<br>
11396c ... clear the log text<br>
11397Note: the HTML text must be contenteditable to catch Key Event.<br>
11398</note>
11399
11400<div id="Shading_1" class="ShadingPlate" draggable="true" contenteditable="true">
11401<div id="Shading_1_Html" class="ShadingHtml" draggable="true"></div>
11402<div id="Shading_1_Log" class="ShadingLog" draggable="true"></div>
11403
11404<canvas id="Shading_1_Canvas" class="ShadingCanvas" width="300px" height="300px" draggable="true">My Canvas.</canvas></div>
11405<style>
11406.ShadingPlate {
11407 z-index:0;
11408 position:static;

```

```

11409 overflow:scroll;
11410 display:block;
11411 width:100%;
11412 height:400px;
11413 font-size:9pt;
11414 font-family:Courier New;
11415 border:1px dashed #000;
11416 color:#444;
11417}
11418.ShadingLog {
11419 z-index:0;
11420 position:relative;
11421 display:block;
11422 top:0px;
11423 left:0px;
11424 overflow:scroll;
11425 width:100%;
11426 font-size:9pt;
11427 font-family:Courier New;
11428 color:#666;
11429 overflow:scroll;
11430 background:rgba(200,255,200,0.4);
11431 height:400px;
11432}
11433.ShadingHtml {
11434 z-index:2;
11435 position:relative;
11436 display:block;
11437 top:0px;
11438 left:0px;
11439 overflow:scroll;
11440 width:100%;
11441 font-size:12pt;
11442 font-family:Courier New;
11443 color:#666;
11444 overflow:scroll;
11445 background:rgba(200,255,200,0.4);
11446 height:400px;
11447}
11448.ShadingCanvas {
11449 z-index:3;
11450 position:relative;
11451 xdisplay:block;
11452 top:0px;
11453 left:100px;
11454 resize:both;
11455 border:1px solid #000;
11456}
11457</style>
11458<input id="Shading_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
11459<input id="Shading_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
11460<input id="Shading_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
11461<span id="Shading_WorkCodeView"></span>
11462<script id="Shading_WorkScript">
11463function Shading_openWorkCodeView(){
11464 function Shading_showWorkCode(){
11465 showHtmlCode(Shading_WorkCodeView,Shading_WorkCodeSpan);
11466 }
11467 Shading_WorkCodeViewOpen.addEventListener('click',Shading_showWorkCode);
11468}
11469const BR = '<'+>';
11470Shading_openWorkCodeView(); // should be invoked by an event
11471 function sh_onClick(e){
11472 Shading_1_Log.innerHTML += 'Click '+e.target.nodeName+'#'+e.target.id
11473 + ' offset('+e.offsetX+', '+e.offsetY+')'
11474 + ' client('+e.clientX+', '+e.clientY+')'
11475 + ' page('+e.pageX+', '+e.pageY+')'
11476 + ' screen('+e.screenX+', '+e.screenY+')'
11477 +BR;
11478 e.stopPropagation();
11479 e.preventDefault();
11480 }
11481 function sh_onKeyUp(e){
11482 if( Shading_1.style.zIndex == '' ){
11483 Shading_1.style.zIndex = 0;
11484 }
11485 zi = parseInt(Shading_1.style.zIndex);
11486
11487 if( e.key.length == 1 ){
11488 Shading_1_Html.innerHTML += e.key;
11489 }
11490
11491 if( e.key == '0' ){ zi = 0; }else
11492 if( e.key == '+' ){ zi += 1; }else
11493 if( e.key == '-' ){ zi -= 1; }else
11494 if( e.key == 'c' ){
11495 Shading_1_Log.innerHTML = '';
11496 }else
11497 if( e.key == 'r' ){
11498 Shading_1.style.position = "relative";
11499 Shading_1.style.top = '0px';
11500 Shading_1.style.left = '0px';
11501 zi = 0;
11502 }else
11503 if( e.key == 'j' || e.code == 'ArrowDown' ){
11504 topx = parseInt(Shading_1.style.top) + 50;
11505 Shading_1.style.top = topx + 'px';
11506 }else
11507 if( e.key == 'k' || e.code == 'ArrowUp' ){
11508 topx = parseInt(Shading_1.style.top) - 50;
11509 Shading_1.style.top = topx + 'px';
11510 }else
11511 if( e.key == 'l' || e.code == 'ArrowRight' ){
11512 lefty = parseInt(Shading_1.style.left) + 50;
11513 Shading_1.style.left = lefty + 'px';
11514 }else
11515 if( e.key == 'h' || e.code == 'ArrowLeft' ){
11516 lefty = parseInt(Shading_1.style.left) - 50;
11517 Shading_1.style.left = lefty + 'px';
11518 }else
11519 if( e.key == 'a' ){
11520 Shading_1.style.position = "absolute";
11521 Shading_1.style.top = '0px';
11522 Shading_1.style.left = '0px';
11523 }else{
11524 }
11525 Shading_1.style.zIndex = zi;
11526 Shading_1_Log.innerHTML += 'Keypup.'+e.target.nodeName+'#'+e.target.id
11527 + 'Up('+e.key+'/' +e.code+')'
11528 + 'z-index:'+zi+'/' +Shading_1.style.zIndex
11529 + 'top:'+Shading_1.style.top
11530 +BR;
11531 e.stopPropagation();
11532 e.preventDefault();

```

```

11533 }
11534 function sh_onKeydown(e){
11535     Shading_1_Log.innerHTML += 'Keydown'+e.target.nodeName+'#'+e.target.id
11536     + 'Down('+e.key+' '+e.code+')'+BR;
11537     e.stopPropagation();
11538     e.preventDefault();
11539 }
11540 function Shading_Setup(){
11541     Shading_1_Log.innerHTML += '<'+h4>Click here<'+/h4>;
11542     Shading_1.addEventListener('keydown',sh_onKeydown);
11543     //window.addEventListener('keydown',sh_onKeydown);
11544     Shading_1.addEventListener('keyup',sh_onKeyUp);
11545     Shading_1.addEventListener('click',sh_onClick);
11546     window.addEventListener('keyup',sh_onKeyUp);
11547     window.addEventListener('click',sh_onClick);
11548
11549     Shading_1_Log.style.top = "-400px";
11550     Shading_1_Log.style.left = "200px";
11551
11552     Shading_1.appendChild(Shading_1_Canvas);
11553     Shading_1_Canvas.style.width = "300px";
11554     Shading_1_Canvas.style.height = "300px";
11555     Shading_1_Canvas.style.position = "relative";
11556     Shading_1_Canvas.style.top = "-750px";
11557     Shading_1_Canvas.style.left = "100px";
11558
11559     const ctx = Shading_1_Canvas.getContext('2d');
11560     ctx.fillStyle = 'rgba(160,0,0,0.9)';
11561     ctx.fillRect(50,50,40,40);
11562     ctx.fillStyle = 'rgba(0,160,0,0.9)';
11563     ctx.fillRect(60,60,40,40);
11564     ctx.fillStyle = 'rgba(0,0,160,0.9)';
11565     ctx.fillRect(70,70,40,40);
11566 }
11567 Shading_Setup();
11568 function Reset_ShadingCanvas(){
11569     Shading_1_Log.removeAttribute('style');
11570     Shading_1_Log.innerHTML = '';
11571     Shading_1_Canvas.removeAttribute('style');
11572 }
11573
11574 </script>
11575 </details>
11576 <!-- Shading_WorkCodeSpan } -->
11577 *//</span>
11578 <!-- ===== Work } ===== -->
11579
11580
11581
11582 <!-- ===== Work { ===== -->
11583 <span id="Template_WorkCodeSpan">
11584 /*
11585 <details><summary>Work Template</summary>
11586 <!-- ===== Template of Work// 2020-0928 SatoxITS { -->
11587 <h2>Template of Work</h2>
11588 <input id="Template_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
11589 <input id="Template_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
11590 <input id="Template_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
11591 <span id="Template_WorkCodeView"></span>
11592 <script id="Template_WorkScript">
11593 function Template_openWorkCodeView(){
11594     function Template_showWorkCode(){
11595         showHtmlCode(Template_WorkCodeView,Template_WorkCodeSpan);
11596     }
11597     Template_WorkCodeViewOpen.addEventListener('click',Template_showWorkCode);
11598 }
11599 Template_openWorkCodeView(); // should be invoked by an event
11600 </script>
11601 </details>
11602 <!-- Template_WorkCodeSpan } -->
11603 *//</span>
11604 <!-- ===== Work } ===== -->
11605
11606
11607
11608 <!-- ===== Work { ===== -->
11609 <span id="OriginalSource_WorkCodeSpan">
11610 /*
11611 <details open=""><summary>Original Source</summary>
11612 <!-- ===== OriginalSource // 2020-1009 SatoxITS { -->
11613 <h2>Original Source of GShell</h2>
11614 <input id="OriginalSource_WorkCodeViewOpen" class="HtmlCodeViewButton" type="button" value="ShowCode">
11615 <input id="OriginalSource_WorkOpenSnapshot" class="HtmlCodeViewButton" type="button" value="Snapshot">
11616 <input id="OriginalSource_WorkCodeSignature" class="HtmlCodeViewButton" type="button" value="Signature">
11617 <span id="OriginalSource_TextElement"></span>
11618 <span id="OriginalSource_WorkCodeView"></span>
11619 <script id="OriginalSource_WorkScript">
11620 function OriginalSource_openWorkCodeView(){
11621     function OriginalSource_showWorkCode(){
11622         //showHtmlCode(OriginalSource_WorkCodeView,OriginalSource_WorkCodeSpan);
11623         //OriginalSource_TextElement = OriginalSourceNode;
11624         //console.log('src3=\n'+OriginalSourceNode.outerHTML);
11625         showHtmlCode(OriginalSource_WorkCodeView,OriginalSourceNode);
11626     }
11627     OriginalSource_WorkCodeViewOpen.addEventListener('click',OriginalSource_showWorkCode);
11628 }
11629 //OriginalSourceNode = document.documentElement.cloneNode();
11630 //OriginalSourceNode = gsh.cloneNode(true); //=====
11631 //console.log('src0=\n'+document.documentElement.outerHTML);
11632 //console.log('src1=\n'+gsh.outerHTML);
11633 //console.log('src2=\n'+OriginalSourceNode.innerHTML);
11634 OriginalSource_openWorkCodeView(); // should be invoked by an event
11635 //showNodesAsHtmlSource(OriginalSource_WorkCodeView,OriginalSourceNode);
11636 function SaveOriginalNode(){
11637     if( false ){
11638         m0 = performance.memory;
11639         mu0 = m0.usedJSHeapSize;
11640         console.log('-- heap bef clone: '
11641             +m0.usedJSHeapSize+' '+m0.totalHeapSize+' '+m0.jsHeapSizeLimit);
11642     }
11643     OriginalSourceNode = gsh.cloneNode(true);
11644     if( false ){
11645         m1 = performance.memory;
11646         mu1 = m1.usedJSHeapSize;
11647         mu = mu1 - mu0;
11648         //alert('-- clone: used heap '+mu0+' -> '+mu1+' = '+mu+' bytes');
11649         console.log('-- heap aft clone: '
11650             +m1.usedJSHeapSize+' '+m1.totalHeapSize+' '+m1.jsHeapSizeLimit);
11651         //OriginalSourceNode = document.documentElement.cloneNode(true);
11652     }
11653 }
11654
11655 function Gsh_setupPage(){
11656     GshSetImages();

```

```
11657 //Indexer_afterLoaded();
11658 //Gshell_initKeyCommands();
11659 //GJConsole_initConsole();
11660 GJConsole_initFactory();
11661 GJLink_init();
11662 InterFrameComm_init();
11663 Gshell_initTopBar();
11664 //VirtualDesktop_init();
11665 Banner_init();
11666 affsetup();
11667 window.setInterval(ShowResourceUsage,1000);
11668//document.addEventListener('keydown',jgshCommand); // should be applied later?
11669}
11670function OnLoad(){
11671 SaveOriginalNode();
11672 Gsh_setupPage();
11673}
11674document.addEventListener('load',Gsh_setupPage);
11675</script>
11676</details>
11677<!-- OriginalSource_WorkCodeSpan } -->
11678*/ //</span>
11679/<!-- ===== Work } ===== -->
11680
11681
11682
11683//</div>
11684//<br><script>OnLoad();</script></span></html>
11685
```