

2020年9月19日 投稿者: SATOXITS

GShell 0.3.6 – 暴走事案

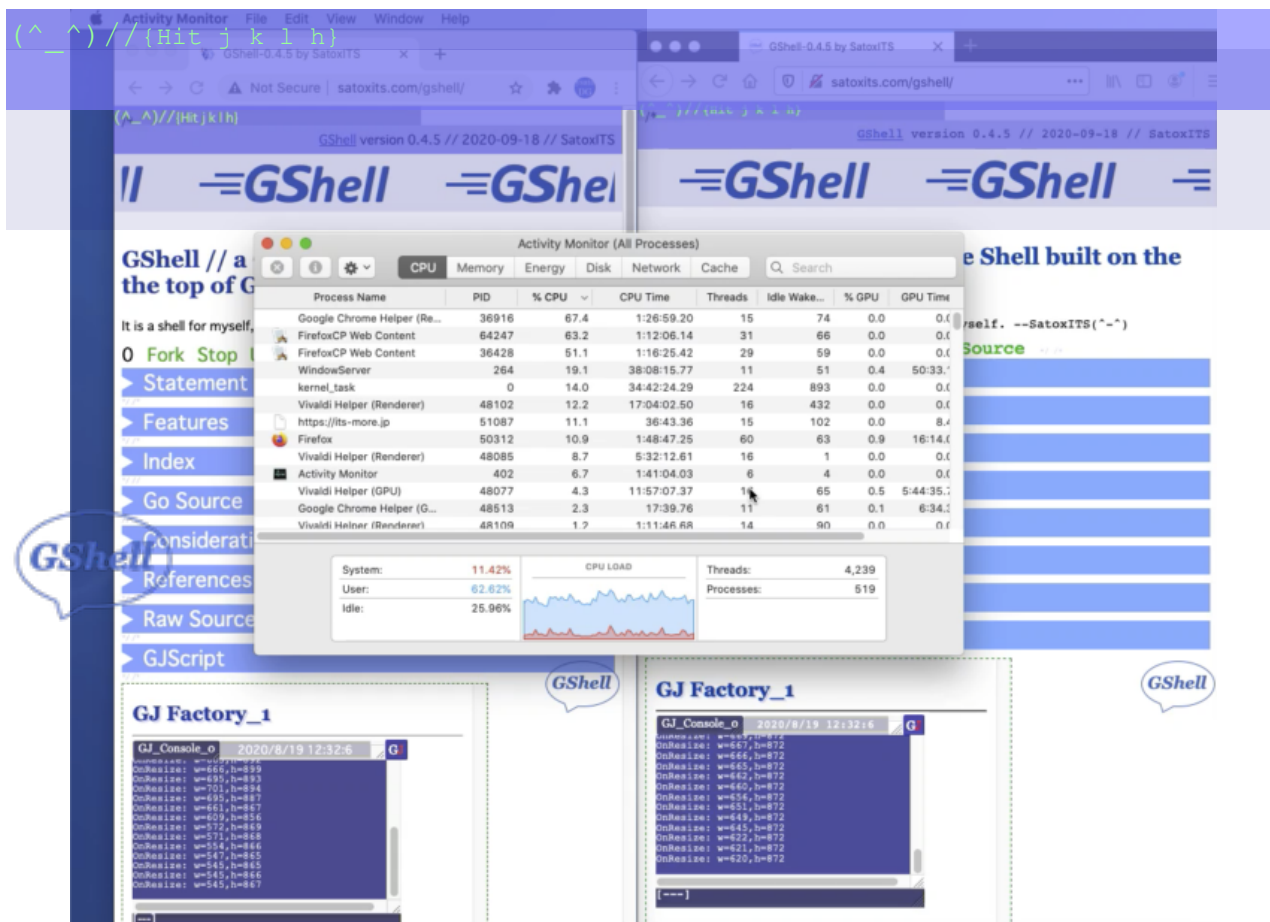
基盤：緊急事態発生。

社長：なんじゃね朝っぱらから吉田くん。

開発：いえ、もうお昼です。

謎のギクシャク君

基盤：GShellのバナーが変だと思ったら、暴走している模様です。CPU食いまくり。



基盤：めっちゃギクシャク動いております。

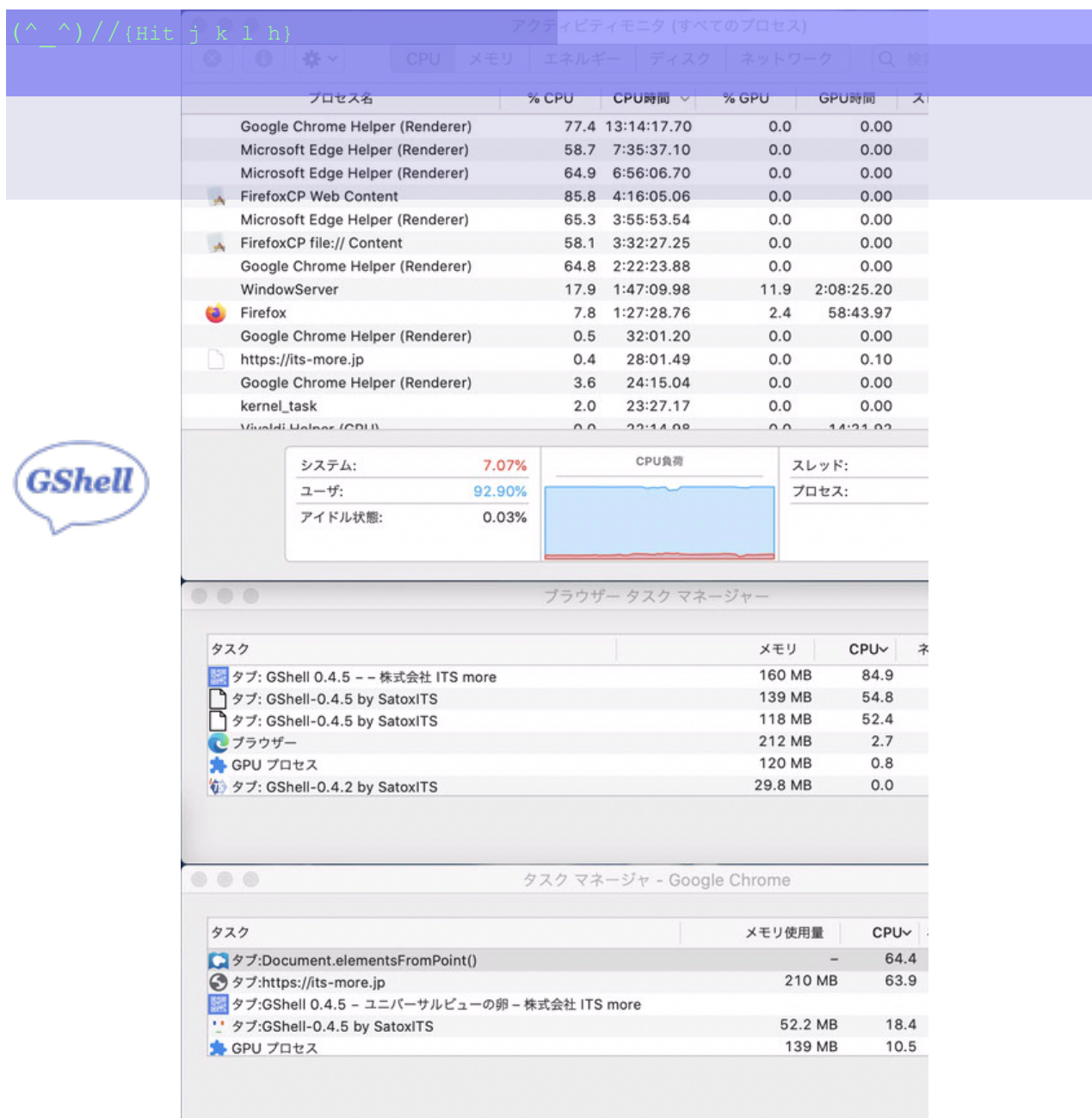
```
(^_^) // {Hit j k l h}
```



社長：新しい芸風ですかね。

開発：iMacの増設メモリのせい？

基盤：いえ、これはMacMiniでの状況。でも、iMacも同様です。



The screenshot displays the Windows Task Manager interface, showing system activity and browser task managers. A GShell logo is visible on the left side of the image.

アクティビティモニタ (すべてのプロセス)

プロセス名	% CPU	CPU時間	% GPU	GPU時間	ス
Google Chrome Helper (Renderer)	77.4	13:14:17.70	0.0	0.00	
Microsoft Edge Helper (Renderer)	58.7	7:35:37.10	0.0	0.00	
Microsoft Edge Helper (Renderer)	64.9	6:56:06.70	0.0	0.00	
FirefoxCP Web Content	85.8	4:16:05.06	0.0	0.00	
Microsoft Edge Helper (Renderer)	65.3	3:55:53.54	0.0	0.00	
FirefoxCP file:// Content	58.1	3:32:27.25	0.0	0.00	
Google Chrome Helper (Renderer)	64.8	2:22:23.88	0.0	0.00	
WindowServer	17.9	1:47:09.98	11.9	2:08:25.20	
Firefox	7.8	1:27:28.76	2.4	58:43.97	
Google Chrome Helper (Renderer)	0.5	32:01.20	0.0	0.00	
https://its-more.jp	0.4	28:01.49	0.0	0.10	
Google Chrome Helper (Renderer)	3.6	24:15.04	0.0	0.00	
kernel_task	2.0	23:27.17	0.0	0.00	
Visual Studio Code	0.0	22:14.08	0.0	14:21.02	

システム: 7.07%
ユーザー: 92.90%
アイドル状態: 0.03%

CPU負荷

スレッド:
プロセス:

ブラウザ タスク マネージャー

タスク	メモリ	CPU	ネ
タブ: GShell 0.4.5 -- 株式会社 ITS more	160 MB	84.9	
タブ: GShell-0.4.5 by SatoxITS	139 MB	54.8	
タブ: GShell-0.4.5 by SatoxITS	118 MB	52.4	
ブラウザ	212 MB	2.7	
GPU プロセス	120 MB	0.8	
タブ: GShell-0.4.2 by SatoxITS	29.8 MB	0.0	

タスク マネージャ - Google Chrome

タスク	メモリ使用量	CPU
タブ: Document.elementsFromPoint()	-	64.4
タブ: https://its-more.jp	210 MB	63.9
タブ: GShell 0.4.5 - ユニバーサルビューの卵 - 株式会社 ITS more		
タブ: GShell-0.4.5 by SatoxITS	52.2 MB	18.4
GPU プロセス	139 MB	10.5

基盤：メモリでパンパンなのはよく見ましたが、CPUが100%はりつきというのは初めて見ました。今回は、メモリはスカスカです。

アクティビティモニタ (すべてのプロセス)

CPU メモリ エネルギー ディスク ネットワーク

プロセス名	メモリ	スレッド	ポート	PID
accountsd	48.4 MB	3	スレッド数	444 y
Acrobat Reader Synchronizer	8.7 MB	12	111	820 y
Acrobat Updater	13.8 MB	7	158	931 y
adid	13.1 MB	2	33	511 _
Adobe Acrobat Reader DC	8 KB	1	0	539 y
adprivacyd	12.6 MB	3	86	1083 y
AirPlayUIAgent	3.4 MB	3	147	812 y
AirPlayXPCHelper	5.0 MB	6	176	159 r
airportd	8.8 MB	7	169	268 r
akd	3.1 MB	2	78	470 y
AlertNotificationService	1.0 MB	2	24	682 y
AlertNotificationService	2.1 MB	2	45	746 y
amfid	20.7 MB	2	42	526 r
AMNetworkAgent	2.0 MB	2	38	610 y

メモリプレッシャー

物理メモリ: 48.00 GB
 使用済みメモリ: 38.11 GB
 キャッシュされたファイル: 5.56 GB
 スワップ使用領域: 0 バイト

アプリケーションメモリ: 32.65 GB
 確保されているメモリ: 5.46 GB
 圧縮: 0 バイト

GShell

基盤：ディスクもまったく静寂。

アクティビティモニタ (すべてのプロセス)

CPU メモリ エネルギー ディスク ネットワーク

プロセス名	書き込みバイト数	読み込みバイト数	PID
FirefoxCP Web Content	0 バイト	0 バイト	1619
Google Chrome Helper (Renderer)	0 バイト	0 バイト	6094
Microsoft Edge Helper (Renderer)	0 バイト	0 バイト	1322
Google Chrome Helper (Renderer)	0 バイト	0 バイト	1029
DiskImageMounter	0 バイト	0 バイト	10284
Microsoft Edge Helper (Renderer)	0 バイト	0 バイト	664
FirefoxCP file:// Content	0 バイト	0 バイト	2159
Microsoft Edge Helper (Renderer)	0 バイト	0 バイト	683
WindowServer	0 バイト	0 バイト	239
Google Chrome Helper (Renderer)	0 バイト	0 バイト	9574
Google Chrome Helper (GPU)	0 バイト	0 バイト	741
Google Chrome	0 バイト	0 バイト	531
Firefox	0 バイト	0 バイト	1611
kernel_task	0 バイト	0 バイト	0

読み込み回数: 840,151
 書き込み回数: 619,240
 毎秒読み込み回数: 3
 毎秒書き込み回数: 0

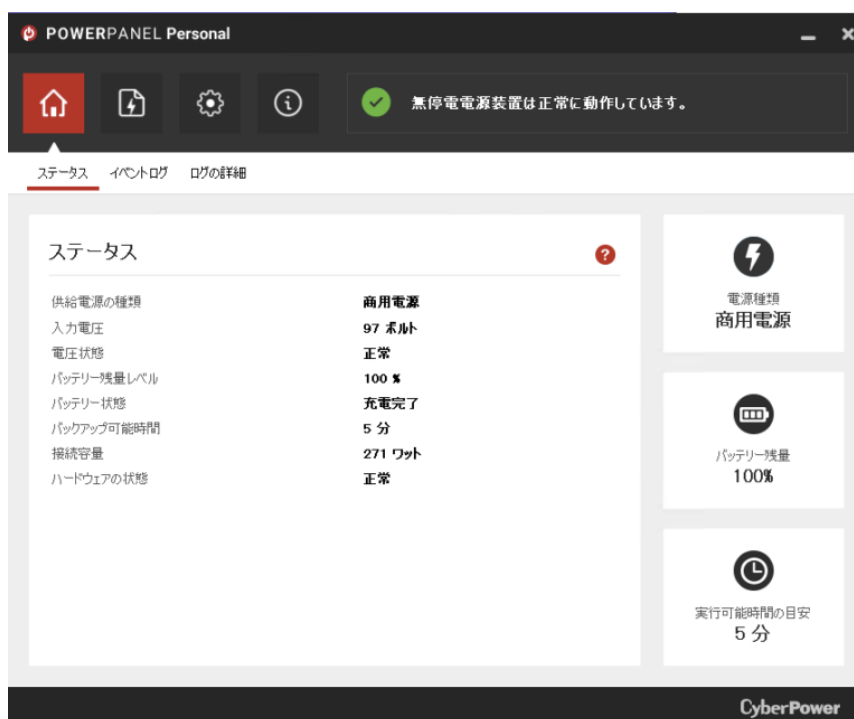
読み込まれたデータ: 13.75 GB
 書き込まれたデータ: 14.56 GB
 毎秒データ読み込み: 12.9 KB
 毎秒データ書き込み: 4.00 KB

基盤：ロードアベレージは50前後で推移。

```
(^_^) //Hit j k l h)
Processes: 532 total, 21 running, 1 stuck, 510 sleeping, 3577 threads      13:29:44
Load Avg: 35.88, 50.48, 44.50  CPU usage: 90.83% user, 8.21% sys, 0.94% idle
SharedLibs: 539M resident, 78M data, 61M linkedit.
MemRegions: 344917 total, 13G resident, 344M private, 3528M shared.
PhysMem: 42G used (5643M wired), 6066M unused.
VM: 3592G vsize, 2000M framework vsize, 0(0) swapins, 0(0) swapouts.
Networks: packets: 4539985/1107M in, 6814885/6638M out.
Disks: 845817/14G read, 623238/15G written.
```

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPR	PGRP	PPID	STATE
1619	plugin-conta	96.6	04:22:57	33/1	1	308	335M-	8192B	0B	1611	1611	running
664	Microsoft Ed	75.0	07:40:20	19/1	1	145	105M-	0B	0B	534	534	running
1029	Google Chrom	74.0	13:19:50	13/1	1	146	2223M+	0B	0B	531	531	running
6094	Google Chrom	72.0	02:28:01	19/1	2	184	204M+	8192B	0B	531	531	running
1322	Microsoft Ed	71.6	04:01:29	19/1	2	175	226M-	8192B	0B	534	534	running
2159	plugin-conta	61.5	03:37:17	33/1	1	230	251M+	0B	0B	1611	1611	running
683	Microsoft Ed	40.4	07:00:45	19/1	1	146	101M-	0B	0B	534	534	running
9574	Google Chrom	19.7	09:39:32	18	1	154	52M+	0B	0B	531	531	sleeping
239	WindowServer	17.8	01:48:45	10	4	4228+	155M	2728K	0B	239	1	sleeping
741	Google Chrom	10.6	23:01.23	10/1	1	315	139M	0B	0B	531	531	running
531	Google Chrom	9.5	09:33.90	34	7	708	308M+	184K-	0B	531	1	sleeping
451	Screensharin	8.7	20:04.35	7	4	124	77M	0B	0B	451	1	stuck
1611	firefox	7.7	88:05.97	61	1	2267	507M	236K	0B	1611	1	sleeping
10460	top	5.4	00:09.34	1/1	0	44	7932K	0B	0B	10460	583	running
0	kernel_task	4.8	23:45.88	229/6	0	0	33M	0B	0B	0	0	running
534	Microsoft Ed	3.1	07:44.20	81	6	774	213M	13M	0B	534	1	sleeping
292	sysmond	2.0	04:03.38	3/1	2/1	28	5208K	0B	0B	292	1	running

基盤：おかげで電力消費は100Wほど上積み状態。



経理：即座にリブートしましょう。

開発：あいや、こういうのはきっと、ブラウザでページをリロードすると直っちゃうんです。

(^) 社長：表示後すぐには発生しませんでしたから、再現にも時間がかかるでしょう。

基盤：ぱっと見、1秒毎にGShellのバナーが止まっている、というのはGJ Consoleの時刻表示のタイミングですねかね。

開発：だと思えます。

社長：ガベッジコレクションか何かですかね？

基盤：同じページを表示しているのに、Safariは重くならないというのは面白いですね。

開発：とりあえずリロードしてみましょう。たぶん解消すると思います。リロードをぷちっ。

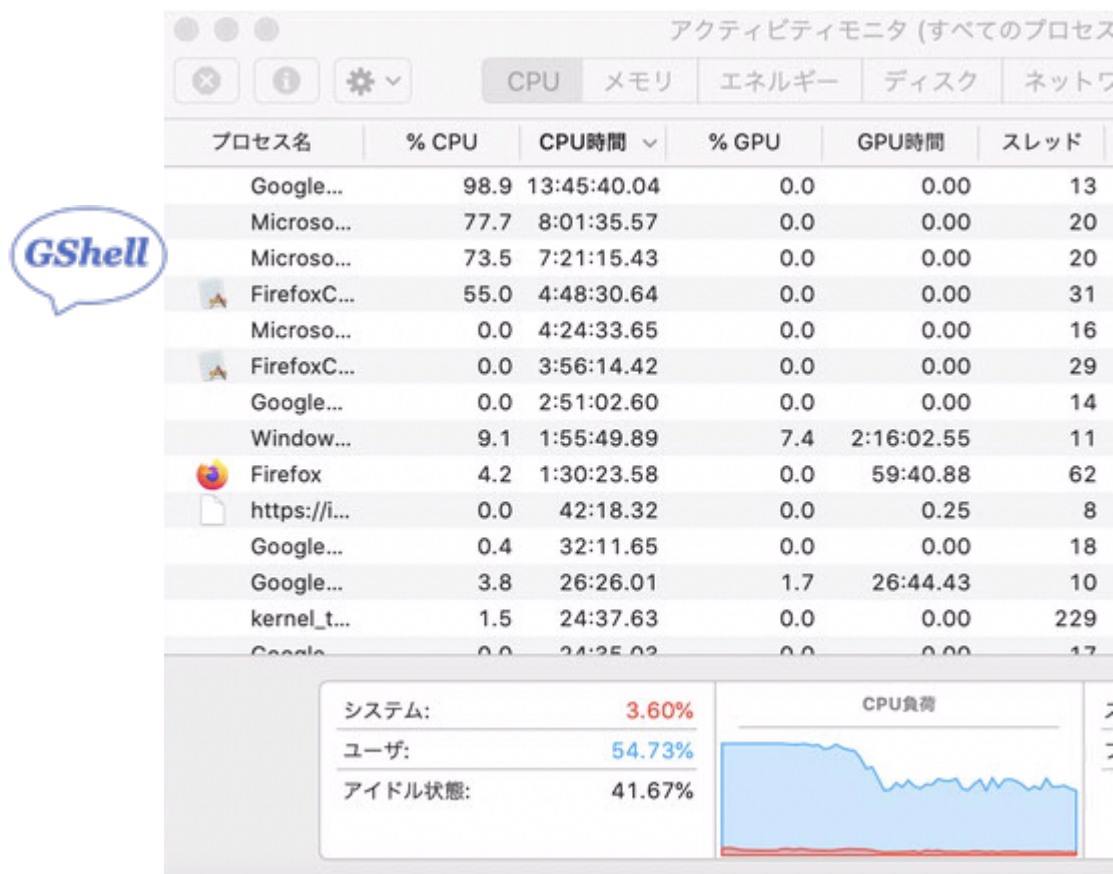


基盤：収まりましたね。

(^_開発)もうひとつもリロードをぷちっ。ああ収まりますね。

社長：アイコン化して画面表示を止めるとどうなるでしょう？

開発：ではiMacのほうで… ぷちぷちぷちぷちぷちっ。




```
(^_^) // Hit Ctrl-h 14:08:46
Processes: 521 total, 5 running, 516 sleeping, 3523 threads
Load Avg: 5.80, 10.67, 19.27 CPU usage: 65.98% user, 4.9% sys, 29.92% idle
SharedLibs: 548M resident, 78M data, 61M linkedit.
MemRegions: 344950 total, 13G resident, 339M private, 3614M shared.
PhysMem: 43G used (5686M wired), 5401M unused.
VM: 3540G vsize, 2000M framework vsize, 0(0) swapins, 0(0) swapouts.
Networks: packets: 4627934/1116M in, 6941881/6746M out.
Disks: 900416/15G read, 666054/17G written.
```

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPR	PGRP	PPID
1029	Google Chrom	99.9	13:49:07	13/1	1	146	2260M+	0B	0B	531	531
664	Microsoft Ed	99.2	08:04:27	20/1	2	146	120M+	0B	0B	534	534
1619	plugin-conta	95.5	04:50:41	31/1	1	306	321M-	8192B	0B	1611	1611
683	Microsoft Ed	91.1	07:24:00	20	2	147	150M+	0B	0B	534	534
531	Google Chrom	7.7	12:35.44	33	6	710	309M+	184K-	0B	531	1
239	WindowServer	5.2	01:56:09	11	5	4503+	159M	2664K	0B	239	1
10460	top	5.1	02:32.37	1/1	0	35	8004K	0B	0B	10460	583
1611	firefox	4.2	90:32.66	62	2	2303	506M	236K	0B	1611	1
741	Google Chrom	3.8	26:34.17	10	1	306	133M-	0B	0B	531	531
451	Screensharin	2.0	20:35.44	6	3	121-	77M-	0B	0B	451	1
0	kernel_task	1.7	24:40.27	229/6	0	0	33M	0B	0B	0	0
1308	Microsoft Ed	1.5	13:33.40	24	2	279	89M	0B	0B	534	534
532	Activity Mon	0.9	07:04.48	8	6	2741	145M	43M	0B	532	1
530	Terminal	0.4	01:35.02	8	1	325	150M	15M	0B	530	1
319	screensharin	0.4	05:51.48	7	1	86	47M	0B	0B	319	1
155	bluetoothd	0.4	00:43.16	5	3	260	8432K	0B	0B	155	1
930	Google Chrom	0.3	32:12.42	17	2	148	87M+	0B	0B	531	531
929	Google Chrom	0.3	03:33.32	20	2	456	111M	0B	0B	531	531
135	launchservic	0.2	01:49.61	5	4	563	12M	0B	0B	135	1
842	Opera Helper	0.2	02:46.95	17	2	841	148M	0B	0B	537	537

基盤：おもての処理が無くなったぶん、裏でアプリが食いたい放題にCPU食ってますね。

開発：メモリが肥大化はしてないですから、リークとかでは無いですね。そもそもスワップ祭りが起きてるわけでも無い。

謎のダブルクリック君

社長：ひとつ、2GBを超えてるChromeがありますが。

開発：ああ、これはうちの子じゃありません。

Processes: 521 total, 9 running, 512 sleeping, 3538 threads 14:19:29
 Load Avg: 5.84, 6.35, 12.07 CPU usage: 65.24% user, 3.94% sys, 30.80% idle
 SharedLibs: 548M resident, 78M data, 61M linkedit.
 MemRegions: 345269 total, 13G resident, 339M private, 3615M shared.
 PhysMem: 43G used (5684M wired), 5216M unused.
 VM: 3546G vsize, 2000M framework vsize, 0(0) swapins, 0(0) swapouts.
 Networks: packets: 4668513/1123M in, 7020161/6825M out.
 Disks: 903915/15G read, 674751/17G written.

PID	COMMAND	%CPU	TIME	#TH	#WQ	#PORT	MEM	PURG	CMPR	PGRP	PPID
1029	Google Chrom	99.3	13:59:44	13/1	1	146	2274M+	0B	0B	531	531
683	Microsoft Ed	98.3	07:32:30	20/1	2	147	112M-	0B	0B	534	534
664	Microsoft Ed	87.2	08:13:20	20/1	2	146	72M-	0B	0B	534	534
1619	plugin-conta	80.0	04:57:10	31/1	1	306	350M+	8192B	0B	1611	1611
239	WindowServer	14.0	01:57:12	10	4	4132-	158M	2664K	0B	239	1
10460	top	5.0	03:09.48	1/1	0	36	8004K	0B	0B	10460	583

タスク マネージャ - Google Chrome

タスク	メモリ使用量	CPU	ネットワーク	プロセス...
タブ:Document.elementsFromPoint()	-	97.4	0	1029
タブ:https://its-more.jp	190 MB	44.5	0	6094
タブ:GShell 0.4.5 - ユニバーサルビューの卵 - 株式会社 ITS more				
ブラウザ	293 MB	6.4	0	531
GPU プロセス	132 MB	3.9	0	741

プロセスを終了

contents/Resources/Documents/developer.mozilla.org/en-US/docs/Web/API/document/elementsFromPoint.html

開発：この子もリロードしてみましよう。ぷちっ。あれ？リロードできないですね。じゃ閉じます。

社長：おお、とたんに平和な風景になりましたね。日常的なプロセス風景。

基盤：iMacのファンが唸らなくなりましたね。というか、電力消費が平常に戻りました。



POWERPANEL Personal

ステータス イベントログ ログの詳細

ステータス

供給電源の種類	商用電源
入力電圧	98 ボルト
電圧状態	正常
バッテリー残量レベル	100 %
バッテリー状態	充電完了
バックアップ可能時間	8 分
接続容量	158 ワット
ハードウェアの状態	正常



The image shows a terminal window with system statistics and a task manager window. The terminal output includes:

```

Processes: 520 total, 5 running, 515 sleeping, 3528 threads
Load Avg: 3.82, 4.22, 7.54 CPU usage: 36.86% user, 3.0% sys, 60.12% idle
SharedLibs: 555M resident, 78M data, 61M linkedit.
MemRegions: 339322 total, 11G resident, 337M private, 3618M shared.
PhysMem: 41G used (5677M wired), 7570M unused.
VM: 3526G vsize, 2000M framework vsize, 0(0) swapins, 0(0) swapouts.
Networks: packets: 4703902/1140M in, 7066764/6856M out.
Disks: 911922/15G read, 691415/17G written.
    
```

The task manager window shows the following tasks:

タスク	メモリ使用量	CPU	ネットワーク	プロセス...
ブラウザ	285 MB	2.2	0	531
サブフレーム: https://amazon.co.jp/	55.8 MB	0.2	0	747
ユーティリティ: Network Service	33.8 MB	0.0	0	743
タブ:elementsFromPoint - Google Search	42.3 MB	0.0	0	6121
サブフレーム: YouTube	51.4 MB	0.0	0	928

The browser task manager window shows the following tasks:

タスク	メモリ	CPU	ネットワーク	プロセス ID
タブ: GShell-0.4.5 by SatoxITS	158 MB	82.4	0	664
タブ: GShell-0.4.5 by SatoxITS	144 MB	81.2	0	683
ブラウザ	233 MB	2.8	0	534

The activity monitor window shows the following processes:

プロセス名	% CPU	CPU時間	% GPU	GPU時間	スレッド	アイ...	PID
Microsoft Edge Helper...	84.5	8:23:37.42	0.0	0.00	20	3	66
Microsoft Edge Helper...	82.0	7:42:21.71	0.0	0.00	20	4	68
FirefoxCP Web Content	56.5	5:04:16.86	0.0	0.00	31	6	161
Microsoft Edge Helper...	0.0	4:24:49.95	0.0	0.00	16	1	132
FirefoxCP file:// Content	11.0	3:57:04.46	0.0	0.00	29	5	215
Google Chrome Helper...	0.0	2:51:19.06	0.0	0.00	14	1	609
WindowServer	2.8	1:57:50.31	1.7	2:19:04.24	11	17	23
Firefox	4.4	1:31:33.14	0.0	59:40.88	62	60	161
https://its-more.jp	0.0	42:24.07	0.0	0.25	8	0	56
Google Chrome Helper...	0.0	27:11.26	0.0	27:20.10	10	1	74
kernel_task	0.8	24:58.26	0.0	0.00	229	612	
Google Chrome Helper...	0.0	24:51.17	0.0	0.00	18	1	92
Vivaldi Helper (GPU)	0.0	22:15.20	0.0	14:21.92	15	0	67
ScreenSharingAgent	0.5	20:52.02	2.2	2:01:44.52	7	12	45

The activity monitor also shows system statistics:

システム:	2.77%
ユーザ:	41.65%
アイドル状態:	55.58%

Additional statistics:

スレッド:	3,528
プロセス:	520

社長：たたんで置いた GJ Console を開くとどうなりますかね。

開発：ぷちぷちぷちぷちぷちっ。おー、再びiMacがうなりはじめました。



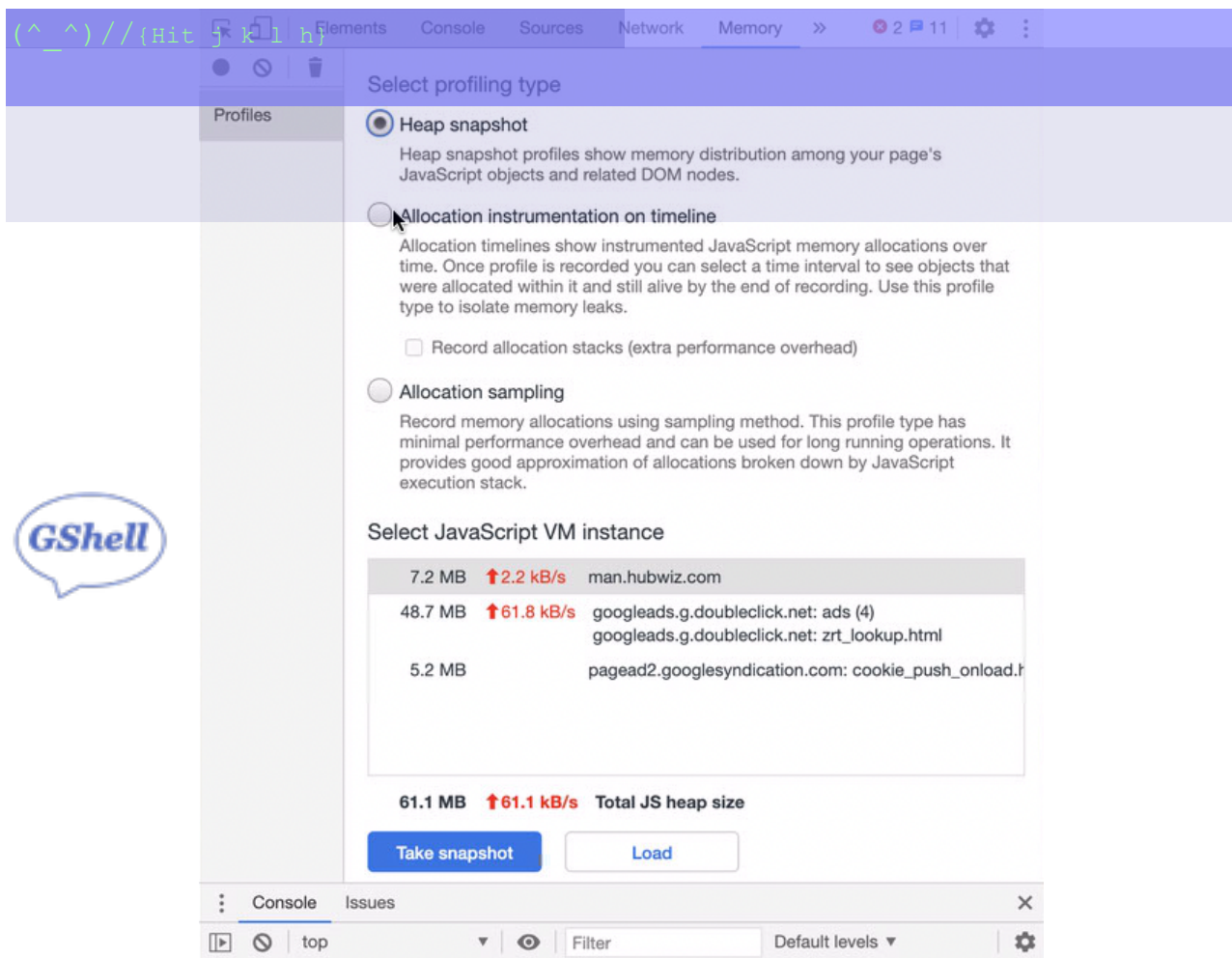
The screenshot shows the GShell POWERPANEL Personal interface. At the top, there is a navigation bar with icons for Home, Power, Settings, Information, and a '無停電' (No Power Off) button. Below this is a menu with 'ステータス' (Status), 'イベントログ' (Event Log), and 'ログの詳細' (Log Details). The 'ステータス' section is active and displays the following information:

項目	値
供給電源の種類	商用電源
入力電圧	98 ボルト
電圧状態	正常
バッテリー残量レベル	100 %
バッテリー状態	充電完了
バックアップ可能時間	5 分
接続容量	267 ワット
ハードウェアの状態	正常

The GShell logo is visible on the left side of the interface.

開発：で、GShell のページはそのままで GJ Console だけ止めたら、バナーがスムーズに動くようになるので、GJ Console が問題であることは確実です。

基盤：Chrome のインスペクタでJavaScriptのメモリ使用状況が見れますね。たとえばさっきの、2GBに肥大してたページですが。



Memory Profiler: Select profiling type

Heap snapshot
Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.

Allocation instrumentation on timeline
Allocation timelines show instrumented JavaScript memory allocations over time. Once profile is recorded you can select a time interval to see objects that were allocated within it and still alive by the end of recording. Use this profile type to isolate memory leaks.

Record allocation stacks (extra performance overhead)

Allocation sampling
Record memory allocations using sampling method. This profile type has minimal performance overhead and can be used for long running operations. It provides good approximation of allocations broken down by JavaScript execution stack.

Select JavaScript VM instance

7.2 MB	↑2.2 kB/s	man.hubwiz.com
48.7 MB	↑61.8 kB/s	googleads.g.doubleclick.net: ads (4) googleads.g.doubleclick.net: zrt_lookup.html
5.2 MB		pagead2.googlesyndication.com: cookie_push_onload.f

61.1 MB ↑61.1 kB/s **Total JS heap size**

Take snapshot Load

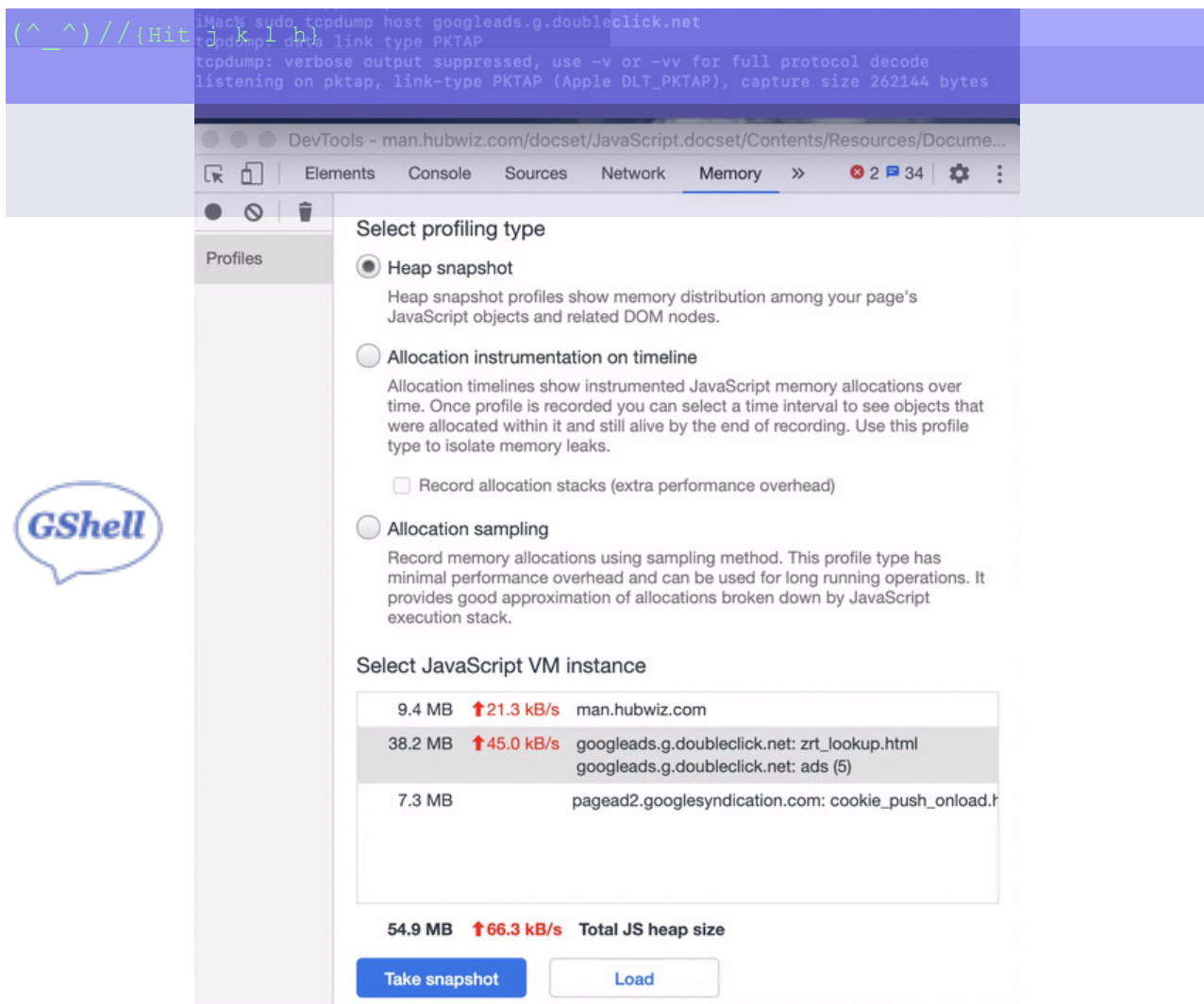
Console Issues

top Filter Default levels

基盤：だいたい60kB/sで、単調に肥大しています。

開発：0.06MB x 60 x 60 x 24 … 一日で5GBを超える計算ですね。

基盤：もう一度リロードして様子を見ます…



The image shows a terminal window at the top with the following text:

```
iMac% sudo tcpdump host googleads.g.doubleclick.net
tcpdump: data link type PKTAP
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on pktap, link-type PKTAP (Apple DLT_PKTAP), capture size 262144 bytes
```

Below the terminal is a Chrome DevTools window showing the Memory profiler. The "Select profiling type" section has three options:

- Heap snapshot**
Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.
- Allocation instrumentation on timeline**
Allocation timelines show instrumented JavaScript memory allocations over time. Once profile is recorded you can select a time interval to see objects that were allocated within it and still alive by the end of recording. Use this profile type to isolate memory leaks.
 Record allocation stacks (extra performance overhead)
- Allocation sampling**
Record memory allocations using sampling method. This profile type has minimal performance overhead and can be used for long running operations. It provides good approximation of allocations broken down by JavaScript execution stack.

The "Select JavaScript VM instance" section shows a table of instances:

9.4 MB	↑21.3 kB/s	man.hubwiz.com
38.2 MB	↑45.0 kB/s	googleads.g.doubleclick.net: zrt_lookup.html googleads.g.doubleclick.net: ads (5)
7.3 MB		pagead2.googleadsyndication.com: cookie_push_onload.f

Below the table, the total JS heap size is shown as 54.9 MB with an upward arrow and 66.3 kB/s. At the bottom, there are two buttons: "Take snapshot" and "Load".



基盤：最初はおとなしいんですが、徐々に来ます。Performanceとやらで10秒間観察…

The image shows a terminal window at the top with the following text:

```
(^_^)//(Hit Ctrl+H)
iMac% sudo tcpdump host googleads.g.doubleclick.net
tcpdump: data link type PKTAP
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on pktap, link-type PKTAP (Apple DLT_PKTAP), capture size 262144 bytes
```

Below the terminal is a Chrome DevTools Performance tab for the page `man.hubwiz.com`. The timeline shows a series of repeating, sawtooth-shaped blocks representing a busy timer. The 'Frames' section below the timeline shows a list of activities:

- JS Heap[5.9 MB - 7.5 MB]
- Documents[4 - 5]
- Nodes[2 488 - 2 497]
- Listeners[188 - 1]

The 'Event Log' section is expanded to show 'Timer Fired' events. The table below is a reproduction of the data shown in the event log:

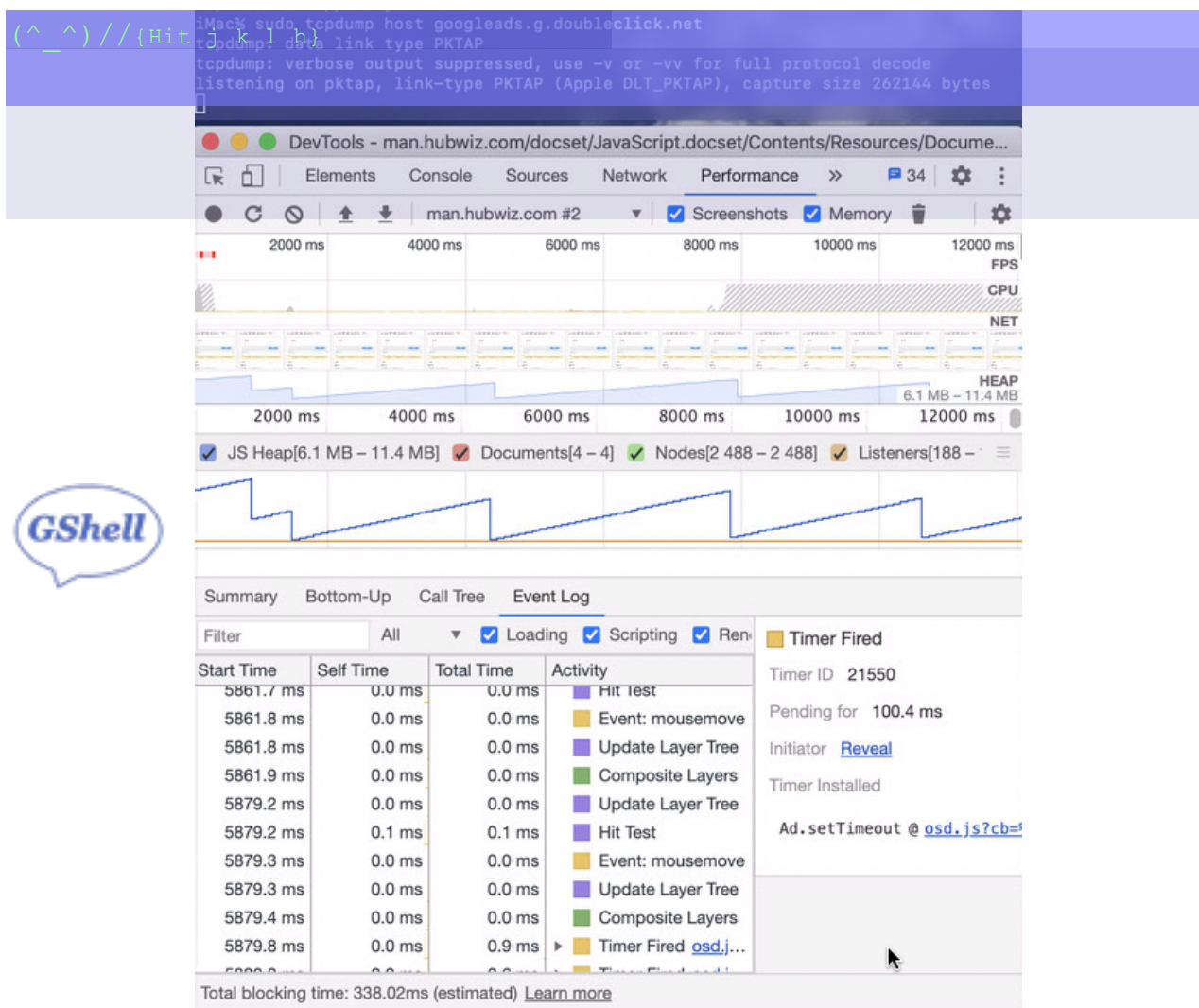
Start Time	Self Time	Total Time	Activity
36.4 ms	0.1 ms	28.5 ms	▶ Timer Fired
64.9 ms	0.0 ms	0.9 ms	▶ Timer Fired
65.9 ms	0.0 ms	0.6 ms	▶ Timer Fired
66.5 ms	0.0 ms	0.1 ms	▶ Timer Fired
169.7 ms	0.0 ms	0.1 ms	▶ Timer Fired h...

Below the table, it says: Total blocking time: 295.87ms (estimated) [Learn more](#)

A blue speech bubble with the text 'GShell' is overlaid on the left side of the DevTools screenshot.

開発：タイマーで処理し続けているということですね。

基盤：マウスイベントを出してみます。



The image shows a terminal window at the top with the following commands and output:

```
iMac% sudo tcpdump host googleads.g.doubleclick.net
tcpdump: data link type PKTAP
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on pktap, link-type PKTAP (Apple DLT_PKTAP), capture size 262144 bytes
```

Below the terminal is a screenshot of the Chrome DevTools Performance tab. The timeline shows a 'Timer Fired' event at 5879.8 ms. The event log table is as follows:

Start Time	Self Time	Total Time	Activity
5861.7 ms	0.0 ms	0.0 ms	Hit Test
5861.8 ms	0.0 ms	0.0 ms	Event: mousemove
5861.8 ms	0.0 ms	0.0 ms	Update Layer Tree
5861.9 ms	0.0 ms	0.0 ms	Composite Layers
5879.2 ms	0.0 ms	0.0 ms	Update Layer Tree
5879.2 ms	0.1 ms	0.1 ms	Hit Test
5879.3 ms	0.0 ms	0.0 ms	Event: mousemove
5879.3 ms	0.0 ms	0.0 ms	Update Layer Tree
5879.4 ms	0.0 ms	0.0 ms	Composite Layers
5879.8 ms	0.0 ms	0.9 ms	Timer Fired osd.j...

The event details for 'Timer Fired' show:

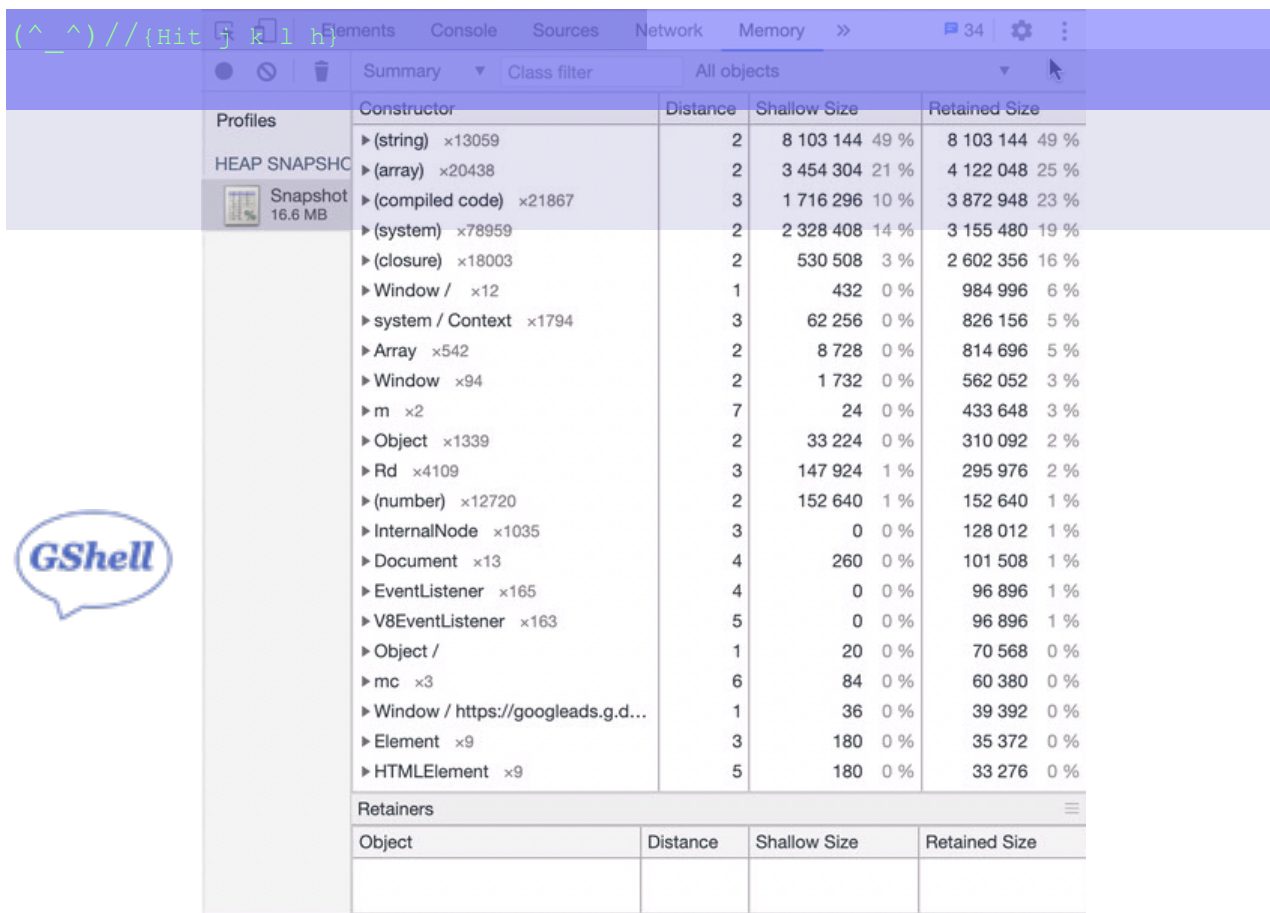
- Timer ID: 21550
- Pending for: 100.4 ms
- Initiator: [Reveal](#)
- Timer Installed
- Ad.setTimeout @ [osd.js?cb=](#)

Total blocking time: 338.02ms (estimated) [Learn more](#)

A 'GShell' logo is visible on the left side of the screenshot.

開発：いったい何が溜まっていくんでしょう？

基盤：メモリのスナップショットを撮ってみます…



Constructor	Distance	Shallow Size	Retained Size
▶(string) ×13059	2	8 103 144 49 %	8 103 144 49 %
▶(array) ×20438	2	3 454 304 21 %	4 122 048 25 %
▶(compiled code) ×21867	3	1 716 296 10 %	3 872 948 23 %
▶(system) ×78959	2	2 328 408 14 %	3 155 480 19 %
▶(closure) ×18003	2	530 508 3 %	2 602 356 16 %
▶Window / ×12	1	432 0 %	984 996 6 %
▶system / Context ×1794	3	62 256 0 %	826 156 5 %
▶Array ×542	2	8 728 0 %	814 696 5 %
▶Window ×94	2	1 732 0 %	562 052 3 %
▶m ×2	7	24 0 %	433 648 3 %
▶Object ×1339	2	33 224 0 %	310 092 2 %
▶Rd ×4109	3	147 924 1 %	295 976 2 %
▶(number) ×12720	2	152 640 1 %	152 640 1 %
▶InternalNode ×1035	3	0 0 %	128 012 1 %
▶Document ×13	4	260 0 %	101 508 1 %
▶EventListener ×165	4	0 0 %	96 896 1 %
▶V8EventListener ×163	5	0 0 %	96 896 1 %
▶Object /	1	20 0 %	70 568 0 %
▶mc ×3	6	84 0 %	60 380 0 %
▶Window / https://googleads.g.d...	1	36 0 %	39 392 0 %
▶Element ×9	3	180 0 %	35 372 0 %
▶HTMLElement ×9	5	180 0 %	33 276 0 %

Retainers			
Object	Distance	Shallow Size	Retained Size

The image shows a browser window with the URL `satoxits.com/gshell/`. The page content includes the GShell logo, the text "GShell // a General purpose Shell built on the top of Golang", and a navigation menu with items like "Statement", "Features", "Index", and "Go Source". A blue speech bubble with the word "GShell" is overlaid on the page. The browser's DevTools Memory profiler is open, showing the "Select profiling type" dialog. The "Heap snapshot" option is selected. The "Total JS heap size" is displayed as 4.5 MB with an increase of 1.0 kB/s. The "Take snapshot" button is highlighted.

開発：え？何か太るような事してましたっけ？

基盤：スナップショットを比較…

GShell // a General purpose Shell built on the top of Golang

GShell version 0.4.5 // 2020-09-18 // SatoxITS

Constructor	D..	Shallow Size	Retained Size
▶ (compiled code) ×11186	3	746 320 23 %	1 248 228 38
▶ (system) ×34936	2	860 400 26 %	1 049 616 32
▶ (array) ×16568	2	757 884 23 %	896 088 27
▶ (string) ×11814	2	772 296 24 %	772 456 24
▶ (closure) ×4094	2	117 248 4 %	304 276 9
▶ Window /	1	36 0 %	190 828 6
▶ Object / ×2	1	40 0 %	141 144 4
▶ Window ×12	2	348 0 %	106 288 3
▶ Window / http://satoxits....	1	36 0 %	77 688 2

GShell // a General purpose Shell built on the top of Golang

GShell version 0.4.5 // 2020-09-18 // SatoxITS

Constructor	D..	Shallow Size	Retained Size
▶ (compiled code) ×13443	3	890 768 25 %	1 515 728 42
▶ (system) ×37159	2	884 652 25 %	1 072 840 30
▶ (array) ×21053	2	855 952 24 %	995 804 28
▶ (string) ×14108	2	837 684 23 %	837 844 23
▶ (closure) ×4094	2	117 248 3 %	300 068 8
▶ Window /	1	36 0 %	188 192 5
▶ Object / ×2	1	40 0 %	141 144 4
▶ Window ×12	2	348 0 %	106 328 3
▶ Window / http://satoxits...	1	36 0 %	76 740 2

社長：毎回コードを生成してるような雰囲気ですね。

(^_開発/ユーザー)生成し直した跡は、昔のはもういないんで。それは処理系が察してくれて、参照が無くなったら自動的にゴミ回収してくれるんだと思ってたんですが…

基盤：マウスイベントでも太りますね。ただ、徐々に減っていく。ゴミ回収が遅延して行われているような。

開発：なんですかね。わかりやすいところで string から…

The screenshot shows a web browser window with the URL `satoxits.com/gshell/`. The page content includes the GShell logo and a description: "GShell // a General purpose Shell built on the top of Golang". Below this, it says "It is a shell for myself, by myself, of myself. -- SatoxITS(^-^)" and lists actions: "0 Fork", "Start", "Unfold", "Digest", and "Source".

The Chrome DevTools Memory tab is open, showing a summary of memory usage. The table below represents the data shown in the table:

Constructor	Dis...	Shallow Si
(compiled code) ×13443	3	890 768 ;
(system) ×37159	2	884 652 ;
(array) ×21053	2	855 952 ;
(string) ×14108	2	837 684 ;
"/*<html> <span id="gsh" data-title="GShell" data-auth	2	421 716 ;
"data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAEAAAB,	2	8 340 ;
"data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAFQAAAA,	2	5 728 ;
"data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAKwAAAB,	2	4 776 ;
"1600486432.136 http://satoxits.com/gshell/ 1600469521	2	600 ;
"data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAG8AAAB,	2	520 ;
"Torque assert 'Convert<uintptr>(endIndex) <= Convert<	-	196 ;
"http://satoxits.comB3F20BEF7767209996ADD7165DF20BD2ch	3	148 ;
"<span id="gsh" data-title="GShell" data-author="sato@	2	28 ;
"1600494988.045(100,575) 1 TEXTAREA#GJ_Console_0_Text"	-	64 ;
"1600494988.019(102,593) 1 TEXTAREA#GJ_Console_0_Mode"	-	64 ;
"1600494981.832(135,511) 1 TEXTAREA#GJ_Console_0_Text"	-	64 ;
"1600487448.865(33,833) 1 TEXTAREA#GJ_Console_0_Text" @	-	64 ;
"1600495012.365(24,571) 1 TEXTAREA#GJ_Console_0_Text" @	-	64 ;
"1600495012.357(27,590) 1 TEXTAREA#GJ_Console_0_Mode" @	-	64 ;
"Distinct or unexpected values in CheckSameObject " @46	-	64 ;
"http://satoxits.comB3F20BEF7767209996ADD7165DF20BD2" @	3	64 ;
" // 2020-0909 added, permanet local storage // https,	5	20 ;
" var PS1 = "% " function GJC_Keydown(keyevent){ key =	5	20 ;
"Range type assertion failed! (value/min/max) " @46181	-	60 ;
"1600494878.136(794,110) 1 HEADER#GshBanner" @21077	-	56 ;

開発：なるほど、コンソールログと、ソースを閲覧した時のテキストと、バナーにdata URLのpngを設定した時の文字列… しかしこれが重くなる原因とは思えないです。イベントリスナーを消さないで追加してるからですかね… ああでも、表示の更新は1秒ごとですが、タイマーイベントは200msごとにしてたような記憶が。textarea を広げてみる…



社長：コンソールログは別の窓にしたほうが良いような。センタリング表示はしないで。あと、メモ帳の窓がほしいですね。ローカルストレージに実体を持てる。

開発：楽しい話の前に、この問題はクリアにして置きたいです。

自前リソース使用モニタ

開発：普通に、参照がなくなったらガベッジコレクションするってMDNには書いてますね。一方、手動で解放する手段はまだ無い模様。

開発：Chromium系にはメモリの使用量を知る手段がある模様。まずこれを使います。

```
(^_^) // {Hit j k l h}
```

漏洩箇所発見

開発：というか、漏れてた場所に思い当たったので直しました。ここです。

```
update = '{GJ_showTime1('+wid+')}';',  
FProductInterval = window.setInterval(update,200)
```

開発：これが200ミリ秒毎にコンパイルされてたわけです。コンパイルされたコードを解放する手段があるのかは不明。

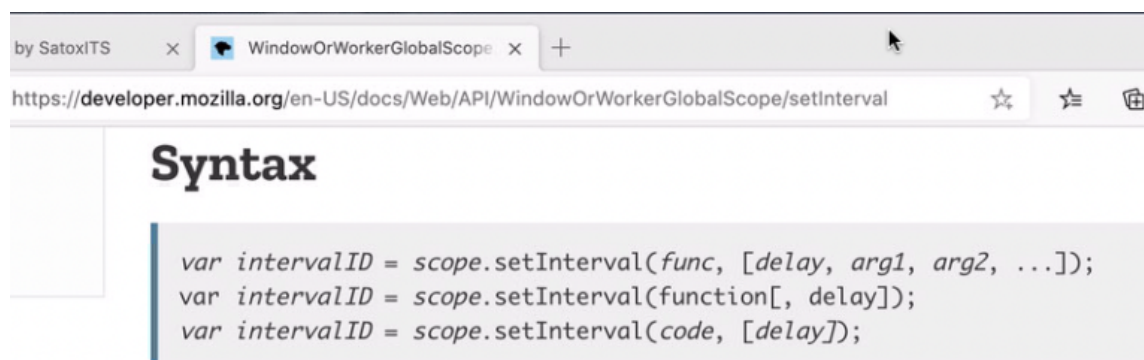
社長：なぜこのような作りに？

開発：まあこんなこともできるかなと、面白かったからですが。要するに、このイベントに専用のハンドラーコードを作りたかった。

開発：まず刹那的な解決。というか、もとはこういう実装でした。

```
update = '{GJ_showTime1('+wid+')}';',  
// 2020-09-19 this causes memory leaks  
//FProductInterval = window.setInterval(update,200)  
FProductInterval = window.setInterval(GJWin_StatUpdate,200)
```

開発：これでリークがなくなりますが、複数のターゲットを作る時によろしくないです。なので、状態をDOMの中のターゲット自体に持たせるということも考えられます。と、思ったのですが、思えばタイマーを設定する時に関数名の他にも引数が渡せることを思い出しました。



```
var intervalID = scope.setInterval(func, [delay, arg1, arg2, ...]);  
var intervalID = scope.setInterval(function[, delay]);  
var intervalID = scope.setInterval(code, [delay]);
```


(^ 社長: まあ、あたりまえっちゃあたり前ですね。

開発: あの時は、引数の順番がなんか変なので使うのに抵抗感があったような気がします。で、引数を使ってこうしました。

```
update = '{GJ_showTime1('+wid+')}';',
// 2020-09-19 this causes memory leaks
//FProductInterval = window.setInterval(update,200)
//FProductInterval = window.setInterval(GJWin_StatUpdate,200)
FProductInterval = window.setInterval(GJ_showTime1,200,wid);
```

開発: これで、オブジェクトコードでメモリが膨らむ問題は解決です。重くなる問題と関係するかはまだ不明。でも、膨らんだコードをコンパイルするのに時間がかかっていたという可能性は大だと思います。それと、Safariで問題が発生しなかったのは、Safariでは使われなくなったコードを認識して捨てることができるからではないだろうか、とか。

* * *

開発: その後、全く太らなくなったので、この暴走事案は解決したと思います。

タスク	メモリ	CPU	ネット...	プロ...
タブ: element.innerHTML - Web API	53.5...	0.0	0	1298
タブ: GShell-0.1.1 by SatoxITS	28.5...	0.0	0	15846
タブ: GShell-0.1.2 by SatoxITS	36.8...	7.1	0	15883
タブ: GShell-0.1.5 by SatoxITS	38.1...	8.9	0	15943
タブ: GShell-0.1.6 by SatoxITS	36.2...	8.9	0	15870
タブ: GShell-0.3.2 by SatoxITS	37.5...	7.9	0	15821
タブ: GShell-0.4.0 by SatoxITS	36.4...	7.9	0	15794
タブ: GShell-0.4.2 by SatoxITS	58.5...	10.0	0	15768
タブ: GShell-0.4.4 by SatoxITS	54.0...	9.7	0	15921
タブ: GShell-0.4.5 by SatoxITS	42.3...	9.1	0	15954
タブ: GShell-0.4.6 by SatoxITS	152 MB	17.9	0	14322
タブ: GShell-0.4.6 by SatoxITS	67.4...	13.1	0	15430

社長: これ、スナップショットの時刻は表示出来ないんですかね。

経理: 解決までに失われた総電力、約0.5kWh。

(^_^) 開発：部屋を暖める熱と消えましたね。

基盤：げ、なんか涼しいと思ったら室温26.5度になってます。急ぎエアコンをOFF。

経理：設定は28度なのに。

開発：温度センサーは室内機についてのでは。吸気の温度とか。上空は暑いでしょう。



* * *

基盤：新情報です。過去の版をいくつかを並べて動かしてみています。

タスク	メモリ	CPU	ネット...	プロ...
タブ: element.innerHTML - Web API	53.5...	0.0	0	1298
タブ: GShell-0.1.1 by SatoxITS	28.5...	0.0	0	15846
タブ: GShell-0.1.2 by SatoxITS	36.8...	7.1	0	15883
タブ: GShell-0.1.5 by SatoxITS	38.1...	8.9	0	15943
タブ: GShell-0.1.6 by SatoxITS	36.2...	8.9	0	15870
タブ: GShell-0.3.2 by SatoxITS	37.5...	7.9	0	15821
タブ: GShell-0.4.0 by SatoxITS	36.4...	7.9	0	15794
タブ: GShell-0.4.2 by SatoxITS	58.5...	10.0	0	15768
タブ: GShell-0.4.4 by SatoxITS	54.0...	9.7	0	15921
タブ: GShell-0.4.5 by SatoxITS	42.3...	9.1	0	15954
タブ: GShell-0.4.6 by SatoxITS	152 MB	17.9	0	14322
タブ: GShell-0.4.6 by SatoxITS	67.4...	13.1	0	15430

基盤：バナーを動かすようになったのが 0.1.2からで、CPUを7%消費してま
す。タイマーで GJ Consoleを動かすようになったのが 0.4.2 から。

JavaScriptの仮想マシンが使用するメモリは論理的には2MB程度のように
すが、仮想マシンとしてははるかに大量のメモリを消費しています。ただ、増加し
続けるかというところでもなく、遅延したゴミ収集が行われている様子が見え
ます。フラグメンテーションも溜まっていくのかなと思います。CPU時間はた
しかに急激には増えていませんが、GJ Consoleにインタラクティブに変更を加
えたpid14322では、より多くのCPUを消費しています。

(^_^) 開発 | ホーム | ひとこと

社長：でも今日はもう疲れましたから、しばらくコンピュータにまかせて、飲みに行きましょう。

— 2020-0919 SatoxITS



gsh-0.4.6.go

ダウンロード

/* */ /*

GShell version 0.4.6 // 2020-09-19 // SatoxITS

≡GShell **≡GShell** **≡(**

GShell // a General purpose Shell built on the top of Golang

It is a shell for myself, by myself, of myself. -SatoxITS(^-^)

0 Fork Stop Unfold Digest Source */ /*

▶ Statement

/ /

▶ Features

/ /

▶ Index

/ /

▶ Go Source

//

▶ Considerations

(^_^)://{Hit j k l h}

▶ References

~//~

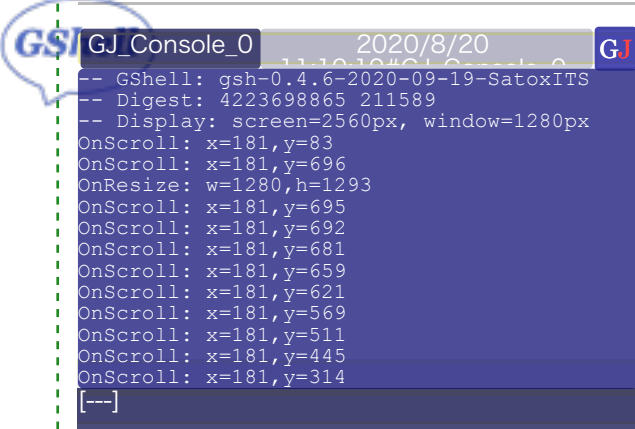
▶ Raw Source

~//~

▶ GJScript

~//~

GJ Factory_1



The screenshot shows a GJ Console window with the following text:

```
GJ_Console_0 2020/8/20
-- GShell: gsh-0.4.6-2020-09-19-SatoxITS
-- Digest: 4223698865 211589
-- Display: screen=2560px, window=1280px
OnScroll: x=181,y=83
OnScroll: x=181,y=696
OnResize: w=1280,h=1293
OnScroll: x=181,y=695
OnScroll: x=181,y=692
OnScroll: x=181,y=681
OnScroll: x=181,y=659
OnScroll: x=181,y=621
OnScroll: x=181,y=569
OnScroll: x=181,y=511
OnScroll: x=181,y=445
OnScroll: x=181,y=314
[---]
```

```
GJShell Console // gsh-0.4.6-2020-09-19-SatoxITS
%
```

~//~