A história do htop

um ”making of”

Hisham Muhammad

@hisham_hm http://hisham.hm

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Sobre mim

autor original do htop, um projeto iniciado em 2004

http://hisham.hm/htop/

desenvolvedor principal do LuaRocks, o gerenciador de pacotes da linguagem Lua

http://luarocks.org/

co-fundador da distribuição GoboLinux

http://gobolinux.org/

Date: 28/03/2017
Talk: A história do htop
Presenter: @hisham_hm

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

O que é o htop

um gerenciador de tarefas interativo

o objetivo era ser um “top melhorado”

o que eu queria dizer com isso: scrolling!

(as versões do top melhoraram muito de lá pra cá!)

```
last pid: 86494;  load averages:  0.83,  0.65,  0.69   up 67+22:48:43  14:44:15
227 processes: 1 running, 224 sleeping, 2 zombie
CPU: 20.2% user,  0.0% nice,  6.5% system,  0.2% interrupt, 73.1% idle
Mem: 1657M Active, 1868M Inact, 273M Wired, 190M Cache, 112M Buf, 11M Free
Swap: 4500M Total, 249M Used, 4251M Free, 5% Inuse

  PID USERNAME  THR PRI NICE   SIZE    RES STATE  C   TIME   WCPU COMMAND
86460 www         1   4    0   150M 30204K accept 1   0:02 11.18% php-cgi
86458 www         1   4    0   150M 29912K accept 0   0:02  8.98% php-cgi
86463 pgsql       1   4    0   949M    99M sbwait 1   0:01  7.96% postgres
85885 www         1   4    0   150M 35204K accept 2   0:07  7.57% php-cgi
85274 www         1   4    0   149M 40868K sbwait 3   0:27  5.18% php-cgi
85267 www         1   4    0   151M 40044K sbwait 2   0:33  4.59% php-cgi
85884 www         1   4    0   150M 41584K accept 2   0:14  4.59% php-cgi
85887 pgsql       1   4    0   951M   128M sbwait 1   0:04  4.20% postgres
85886 pgsql       1   4    0   949M   161M sbwait 0   0:08  3.37% postgres
86459 pgsql       1   4    0   949M 75960K sbwait 2   0:01  3.37% postgres
85279 pgsql       1   4    0   950M   192M sbwait 2   0:14  2.39% postgres
85269 pgsql       1   4    0   950M   199M sbwait 1   0:19  2.20% postgres
85268 www         1   4    0   152M 44356K sbwait 2   0:32  1.17% php-cgi
85273 pgsql       1   4    0   950M   215M sbwait 0   0:19  1.17% postgres
97082 pgsql       1  44    0 26020K  6832K select 0  46:55  0.00% postgres
  892 root        1   4    0  3160K     8K -      2  13:33  0.00% nfsd
 1796 root        1  44    0 19780K 13660K select 3  12:43  0.00% Xvfb
```

./htop

```
  0  [|||||                        8.4%]     Tasks: 94, 42 thr; 1 running
  1  [|||                          3.8%]     Load average: 0.47 0.26 0.22
  2  [|                            0.6%]     Uptime: 04:28:22
  3  [||                           1.3%]     Battery: n/a
  Mem[|||||||||||||||||||||||||||||||||||1114/3819MB]
  Swp[                                0/0MB]

IO Priority:               PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM% Command
None (based on nice)         1 root       40   0  1672   572   508 S  0.0  0.0 init [2]
Realtime 0 (High)        15814 hisham     40   0 16140  6284  3856 S  0.0  0.2 ├─ urxvt -cr green -fn *-lode-* -fb *
Realtime 1               15815 hisham     40   0  8804  6004  1692 S  0.0  0.2 │  └─ zsh
Realtime 2               15836 hisham     40   0 45512 30796 13920 S  0.0  0.8 │     └─ gimp
Realtime 3               16474 hisham     40   0 26020 12624  8792 S  0.6  0.3 │        ├─ /System/Index/lib/gimp/2.
Realtime 4               16047 hisham     40   0 20552  5344  3236 S  0.0  0.1 │        └─ /System/Index/lib/gimp/2.
Realtime 5               15345 hisham     40   0 62852 54784  3920 S  0.0  1.4 ├─ urxvt -cr green -fn *-lode-* -fb *
Realtime 6               15346 hisham     40   0  9132  6460  1868 S  0.0  0.2 │  └─ zsh
Realtime 7 (Low)         15357 hisham     40   0  1716   564   468 T  0.0  0.0 │     └─ cw: wrapping [find] {pid=153
Best-effort 0 (High)     15358 hisham     40   0 11768  9128   756 S  0.0  0.2 │        └─ find
Best-effort 1            15291 hisham     40   0 16332  6492  3920 S  0.0  0.2 ├─ urxvt -cr green -fn *-lode-* -fb *
Best-effort 2            15292 hisham     40   0  9000  6316  1856 S  0.0  0.2 │  └─ zsh
Best-effort 3            15340 hisham     40   0  3116  1852  1148 R  4.4  0.0 │     └─ ./htop
Best-effort 4            14628 hisham     40   0 16140  6304  3864 S  0.0  0.2 ├─ urxvt -cr green -fn *-lode-* -fb *
Best-effort 5            14629 hisham     40   0  9240  6496  1868 S  0.0  0.2 │  └─ zsh
Best-effort 6            14646 hisham     40   0  1712   548   468 S  0.0  0.0 │     └─ cw: wrapping [env] {pid=1464
Best-effort 7 (Low)      14647 hisham     40   0  5644  2620  1204 S  0.0  0.1 │        └─ /bin/bash /System/Links/E
Idle                     16439 hisham     40   0  8012  5340  2348 D  0.6  0.1 │           └─ python /System/Links/E
                         13475 hisham     40   0 16140  6380  3864 S  0.0  0.2 ├─ urxvt -cr green -fn *-lode-* -fb *
                         13476 hisham     40   0  8988  6320  1868 S  0.0  0.2 │  └─ zsh
                         13808 hisham     40   0  3976  1912  1532 S  0.0  0.0 │     └─ ssh -t loderunner,htop@shell
                         13384 hisham     40   0 16140  6288  3864 S  0.0  0.2 ├─ urxvt -cr green -fn *-lode-* -fb *
                         13385 hisham     40   0  9000  6296  1844 S  0.0  0.2 │  └─ zsh
                         10544 hisham     40   0 26212 16596  3920 S  0.0  0.4 ├─ urxvt -cr green -fn *-lode-* -fb *
EnterSet      EscCancel
```

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Motivações iniciais

Resolver um problema próprio

(“scratch your itch”)

Aprender!

Date: 28/03/2017
Talk: A história do htop
Presenter: @hisham_hm

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

“Refazer” um programa clássico

!?


Marshall
Marshall

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

“Refazer” um programa clássico

...não é um bicho de sete cabeças

Os programas que você usa todo dia não são mágicos

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Como obter os dados dos processos?

O kernel os apresenta em /proc

Um sistema de arquivos virtual

/proc/12345/mem

/proc/12345/stat

/proc/cpuinfo

...

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Como ler esses dados?

Ora, da maneira mais óbvia e simples possível

open()

read()

close()

Date: 28/03/2017
Talk: A história do htop
Presenter: @hisham_hm

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Mas como o top faz?

strace top 2> top.log


EXIT
45
RUSH

```
close(4)                                = 0
open("/proc/6/statm", O_RDONLY)         = 4
read(4, "0 0 0 0 0 0 0\n", 1023)        = 14
close(4)                                = 0
stat("/proc/7", {st_mode=S_IFDIR|0555, st_size=0, ...}) = 0
open("/proc/7/stat", O_RDONLY)          = 4
read(4, "7 (watchdog/0) S 2 0 0 0 -1 2216"..., 1023) = 161
close(4)                                = 0
open("/proc/7/statm", O_RDONLY)         = 4
read(4, "0 0 0 0 0 0 0\n", 1023)        = 14
close(4)                                = 0
stat("/proc/21", {st_mode=S_IFDIR|0555, st_size=0, ...}) = 0
open("/proc/21/stat", O_RDONLY)         = 4
read(4, "21 (cpuset) S 2 0 0 0 -1 2216722"..., 1023) = 157
close(4)                                = 0
open("/proc/21/statm", O_RDONLY)        = 4
read(4, "0 0 0 0 0 0 0\n", 1023)        = 14
close(4)                                = 0
stat("/proc/22", {st_mode=S_IFDIR|0555, st_size=0, ...}) = 0
open("/proc/22/stat", O_RDONLY)         = 4
read(4, "22 (khelper) S 2 0 0 0 -1 221672"..., 1023) = 158
close(4)                                = 0
open("/proc/22/statm", O_RDONLY)        = 4
read(4, "0 0 0 0 0 0 0\n", 1023)        = 14
close(4)                                = 0
stat("/proc/23", {st_mode=S_IFDIR|0555, st_size=0, ...}) = 0
open("/proc/23/stat", O_RDONLY)         = 4
read(4, "23 (kdevtmpfs) S 2 0 0 0 -1 2149"..., 1023) = 159
close(4)                                = 0
open("/proc/23/statm", O_RDONLY)        = 4
read(4, "0 0 0 0 0 0 0\n", 1023)        = 14
close(4)                                = 0
stat("/proc/24", {st_mode=S_IFDIR|0555, st_size=0, ...}) = 0
open("/proc/24/stat", O_RDONLY)         = 4
read(4, "24 (netns) S 2 0 0 0 -1 22167224"..., 1023) = 156
close(4)                                = 0
open("/proc/24/statm", O_RDONLY)        = 4
:
```

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Aí “só era questão de programar”

Ler os dados do /proc

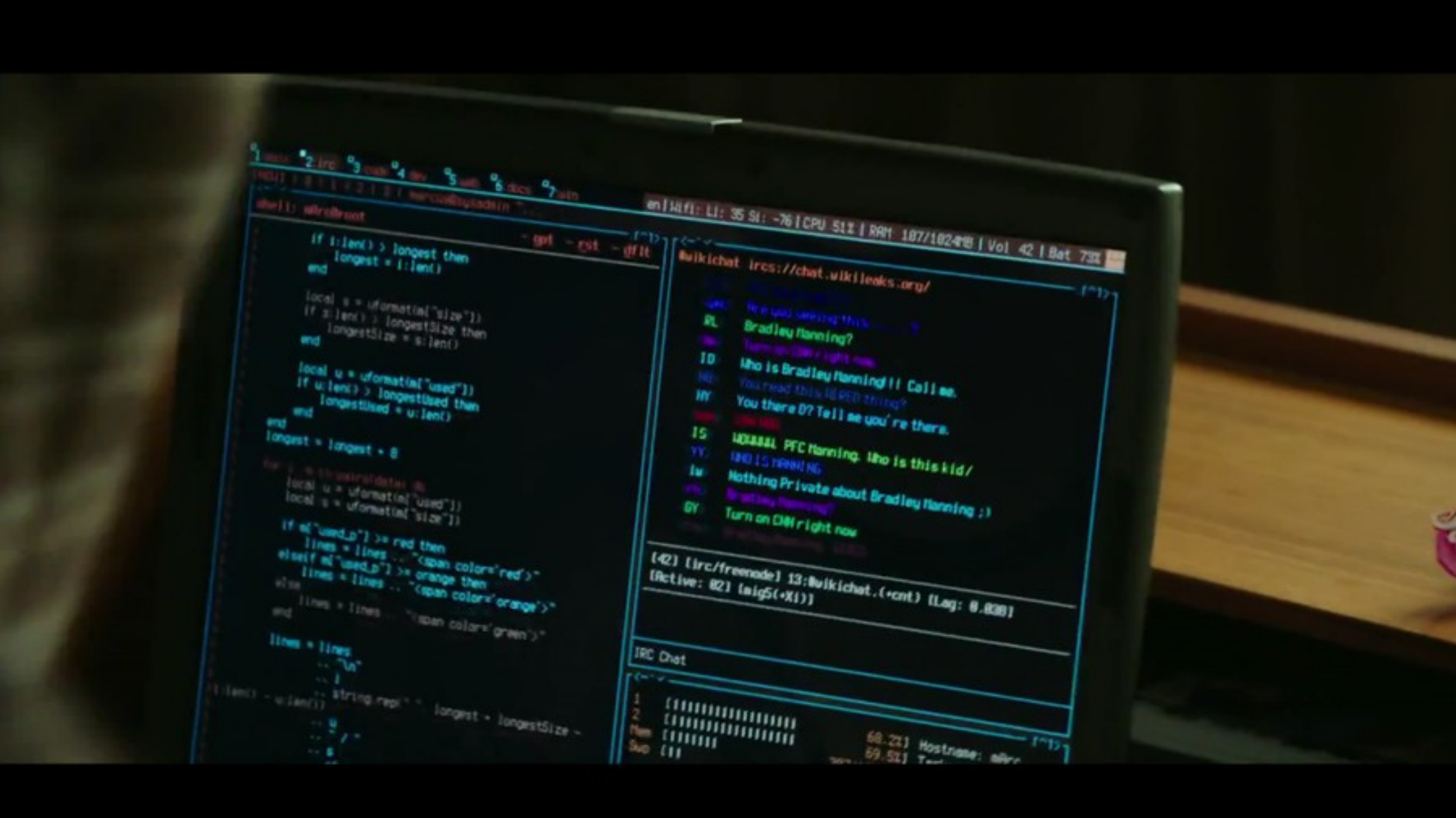
“Desenhar” a interface

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Um detalhe sobre a interface

A arte imita a vida

A vida imita a arte


#wikichat irc://chat.wikileaks.org/
RL Bradley Manning?
ID Who is Bradley Manning!! Call me.
MY You there 0? Tell me you're there.
IS PFC Manning: Who is this kid/
IW Nothing Private about Bradley Manning ;)
GY Turn on CNN right now
[42] [irc/freenode] 13:#wikichat.(+cnt) [Lag: 0.038]
[Active: 02] [mig5(+Xi)]
IRC Chat


"salt":"u8CYIHcQ","md5":"414
abf2e3b53e7","sha1":"ad491ed75cf2
87f7e58094","sha256":"179f46e813e
9e8f874cb84f4c9b196a3c86ce03a4104
"1103343099","dob":"46047625","pho
8","cell":"(920)-571-4616","SSN":
cture":{"large":"http://api.random
women/2.jpg","medium":"http://api
traits/med/women/2.jpg","thumbnai
domuser.me/portraits/thumb/women/
"0.4.1"},"seed":"5f477579f6e3ce79
CPU[|||
Mem[||||||275
Swp[
Tasks: 60, 24 thr; 1 ru
Load average: 0.64
Uptime: 42 days, 14:08:
PID USER PRI NI VIRT RES SHR S CPU%
17339 mysql 20 0 451M 127M 3440 S 0.5
167 20 0 5220 2136 732 S 0.5
17386 20 0 31320 4912 1788 S 0.0
17833 20 0 74180 9128 3220 S 0.0
29565 20 0 74428 9180 3240 S 0.0
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7
52 37 0 2555M 104M 0 S 0.0 1
ime=21.339 ms
F1Help F2Setup F3Search F4Invert F5Tree F6SortBy F7
1 [|
2 [|||
3 [|
4 [|
Mem[|||52
Swp[
Tasks: 98, 189 thr; 1 r
Load average: 0.43
Uptime: 09:52:26
PID USER PRI NI VIRT RES SHR S CPU%
3386 mc 20 0 10.8G 2237M 26144 S 24.5 1
3427 20 0 10.8G 2237M 26144 S 22.5 1
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7

Efeito China provoca queda em bolsas de valores pelo mundo

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Um programa que “deu certo”

Sempre é difícil descobrir o porquê

Alguns palpites sobre fatores que influenciaram

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Ubiquidade da plataforma

o terminal!

talvez a única plataforma hoje tão presente
(para desenvolvedores)
quanto a web

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

A linguagem

fazer em C

...mas não pelos motivos “óbvios”

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Facilidade de compilar em qualquer lugar

Poucas dependências:

ncurses

cores e coordenadas no terminal

todo Linux tem (dependência do Bash)

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Facilidade de compilar em qualquer lugar

Poucas dependências:

ncurses

cores e coordenadas no terminal

todo Linux tem (dependência do Bash)

e só

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Facilidade de compilar em qualquer lugar

Build com AutoTools:

./configure

make

sudo make install

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Facilidade de empacotar

Fácil de compilar

=

Fácil de empacotar em uma distro

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

E assim começou o htop

```
  Press enter to kill
CPU[|||                                  5.8%]
Mem[||||||||                          757116KB]
Swp[                                       0KB]

  PID USER      PR  NI  VIRT   RES   SHR S CPU% COMM
 5936 hisham    23   3 107236 7148  3628 S  0.0 /usr/bin/pulseaudio --start
 5994 root      23   3  6224  1180   856 S  0.0 wpa_supplicant -B -c/etc/wpa_sup
 6059 root      23   3  2464   400   208 S  0.0 dhcpcd wlan0
 6088 hisham    23   3 26092  2804  1820 S  0.0 /Programs/LibreOffice/4.2.3.3/li
 6105 hisham    23   3 261576884800 59032 S  0.0 /Programs/LibreOffice/4.2.3.3/li
 6122 hisham    23   3  6016  1516  1156 S  0.0 /bin/sh /usr/bin/firefox
 6126 hisham    23   3 81205646535237432 S  0.4 /Programs/Firefox/28.0/bin/../fi
 6143 hisham    20   0 25640  2496  2112 S  0.0 /usr/lib/at-spi2-core/at-spi-bus
15278 root      20   0     0     0     0 S  0.0 (kworker/u8:0)
15502 root      20   0     0     0     0 S  0.0 (kworker/2:2)
16607 hisham    23   3 14008847164 12820 S  0.0 terminology
16612 hisham    23   3  7852  3580  2000 S  0.0 /usr/bin/zsh
17981 hisham    23   3 13986449280 12792 S  0.0 terminology
17986 hisham    23   3  7440  3104  1952 S  0.0 /usr/bin/zsh
20667 hisham    23   3 13980445648 12788 S  0.0 terminology
20796 hisham    23   3  7488  3156  1952 S  0.0 /usr/bin/zsh
23164 root      20   0     0     0     0 S  0.0 (kworker/2:0)
```

htop-0.13.tar.gz

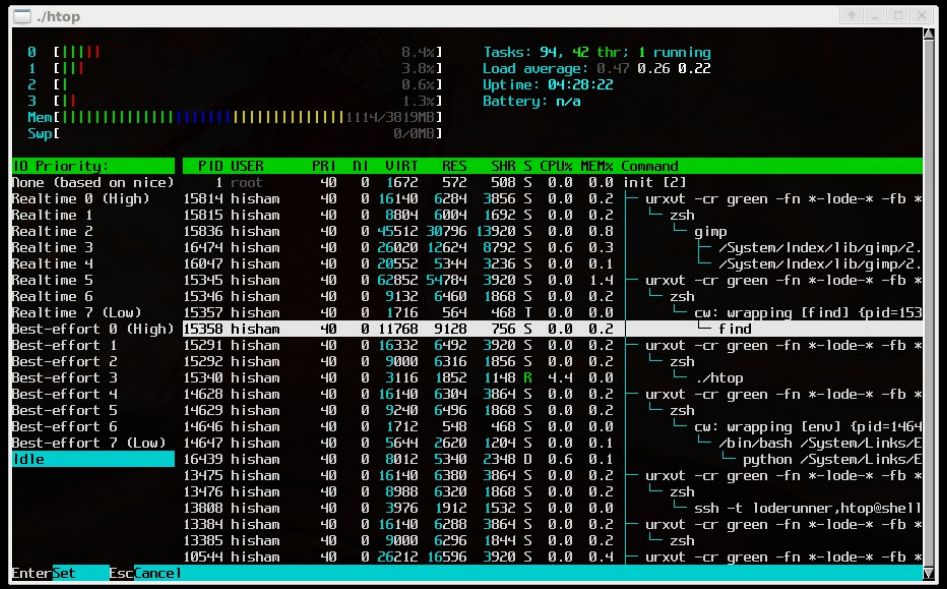
22kB

wc -l *.c

1684

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Para comparação



htop-2.0.2.tar.gz

476kB

wc -l *.c */*.c

13811

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

A grande mudança

Tornar o htop portável

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

htop off-Linux

alguns usuários já rodavam no FreeBSD
usando linprocfs (uma camada de emulação)

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Backstory

« A wild Mac OS X fork appears! »

um port ad-hoc do htop 0.8.2

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Passa o tempo

bug reports começam a aparecer

bugs antigos resolvidos há anos!

começa a ficar chato

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Crowdfunding

Portar o htop para Mac

1/3 da meta atingida

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Tornar o htop portável

Torná-lo mais fácil de ser portado

Abrir o caminho para novos ports

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Como quebrar a dependência de SO da maneira fácil

tente compilar em outro SO!

montar uma VM

primeira tentativa: FreeBSD

desisti rápido

segunda tentativa: PC-BSD

mais fácil para um usuário Linux!

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Abordagens para aplicações multi-plataforma

bases de código separadas

iOS vs. Android

um spaghetti de #ifdefs

às vezes é o jeito

APIs bem definidas entre a parte portável e não-portável

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Isolar a parte Linux

linux/

unsupported/

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Isolar a parte Linux

linux/

unsupported/

freebsd/

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Uma abordagem “OO”

Process.c

Process.h

ProcessList.c

ProcessList.h

...

linux/

LinuxProcess.c

LinuxProcessList.c

...

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Subi para o GitHub e...

unsupported/

linux/

freebsd/

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Subi para o Github e...

unsupported/

linux/

freebsd/ - merged PRs!

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Subi para o Github e...

unsupported/

linux/

freebsd/

openbsd/

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Subi para o Github e...

unsupported/

linux/

freebsd/

openbsd/

darwin/

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Impacto positivo

novas pessoas contribuindo

sempre algo bom!

maior diversidade de contribuições

Galera do OpenBSD e a preocupação com segurança!

alguém quer portar para o NetBSD?

...Cygwin!?

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Conclusões

Siga os padrões da sua comunidade

(código, empacotamento)

Portabilidade vale a pena mesmo

“Stratch your itch!”

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command

Obrigado!

http://hisham.hm/htop

http://github.com/hishamhm/htop

@hisham_hm