hog

By: Donovan Moini, Ian Lizarda, and Serena Zafiris

Intro to hog

A utility function named ***hog*** that returns which process is using the greatest percentage of the CPU

hog takes in a single argument *n*, where $0 < n \leq 20$, which returns the top *n* processes using the greatest percentage of the CPU.

If *n* is preceded by the option *-p*, ***hog*** will return all processes that use *n* or more percentage of the CPU

What is ps?

ps = process status

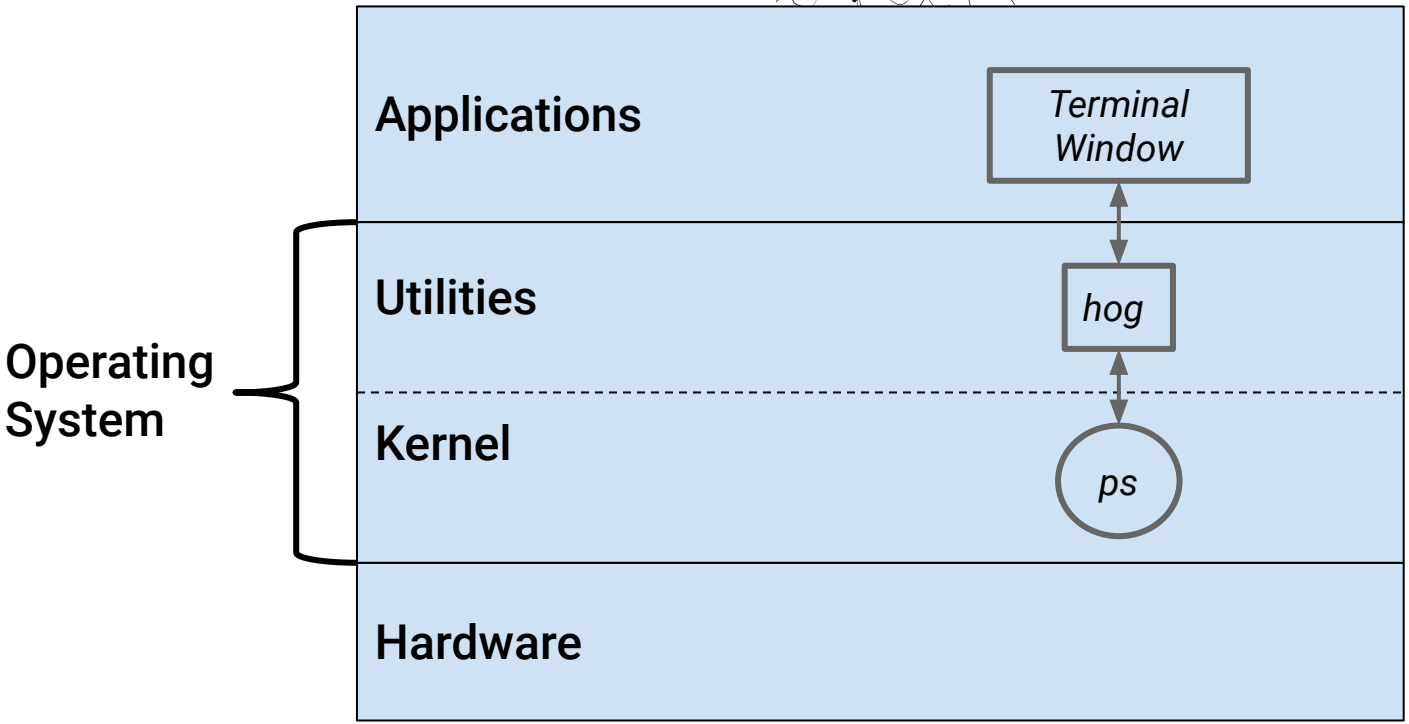
Used to provide information about the currently running processes including:

- Process identification number (PID)
- Percentage of CPU usage (%CPU)
- Percentage of memory usage (%MEM)

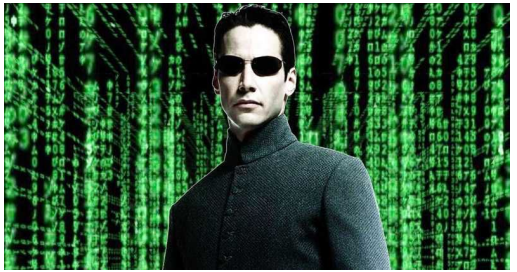
The -e option generates a list of information about every process currently running.

The -v options displays the output in virtual memory format.

How it works


Applications
Terminal Window
Utilities
hog
Kernel
ps
Hardware
Operating System

The Command Line



hog [-p] n

- p = optional argument that asks for a minimum usage requirement to show up in the list, i.e *-p 25* would require the program use at least 25% of the CPU
- n = argument to restrict the list itself. If preceded by *-p*, *n* refers to the programs using up *n* percentage or more of the CPU. If *n* is the sole argument, *n* refers to the top *n* programs using up the CPU.

The Command Line Issue

ps -ev (Mac)

```
➜  ~ ps -ev
  PID STAT      TIME  SL  RE PAGEIN      VSZ    RSS   LIM     TSIZ  %CPU %MEM COMMAND
 6126 S      1:49.84   0   0      0  5600156 382128     -        0   0.4  4.6 /Applications/Google Chrome.app/Contents/
 6133 S      0:18.72   0   0      0  5367972 256212     -        0   0.1  3.1 /Applications/Google Chrome.app/Contents/
  376 S     20:37.11   0   0      0  5795516 243024     -        0   0.1  2.9 /Applications/Google Chrome.app/Contents/
 6030 S      5:10.92   0   0      0  5271664 232420     -        0   0.0  2.8 /Applications/Google Chrome.app/Contents/
 6119 S      1:02.15   0   0      0  5566932 230240     -        0   0.3  2.7 /Applications/Google Chrome.app/Contents/
```

ps -ev (Ubuntu)

```
dis@dis-VirtualBox:/usr/bin$ ps -ev
  PID TTY      STAT   TIME  MAJFL   TRS    DRS    RSS %MEM COMMAND
 2011 tty2     Ssl+   0:00      1    58 187837   6144  0.0 /usr/lib/gdm3/gdm-x-ses
 2013 tty2     Sl+    0:01     34  1828 479063 148608  1.8 /usr/lib/xorg/Xorg vt
```

The Command Line Issue Resolved

ps -ev (Mac)

```
➜  ~ ps -ev
  PID STAT      TIME  SL  RE PAGEIN      VSZ    RSS   LIM     TSIZ  %CPU %MEM COMMAND
 6126 S      1:49.84   0   0      0  5600156 382128     -        0   0.4  4.6 /Applications/Google Chrome.app/Contents/
 6133 S      0:18.72   0   0      0  5367972 256212     -        0   0.1  3.1 /Applications/Google Chrome.app/Contents/
  376 S     20:37.11   0   0      0  5795516 243024     -        0   0.1  2.9 /Applications/Google Chrome.app/Contents/
 6030 S      5:10.92   0   0      0  5271664 232420     -        0   0.0  2.8 /Applications/Google Chrome.app/Contents/
 6119 S      1:02.15   0   0      0  5566932 230240     -        0   0.3  2.7 /Applications/Google Chrome.app/Contents/
```

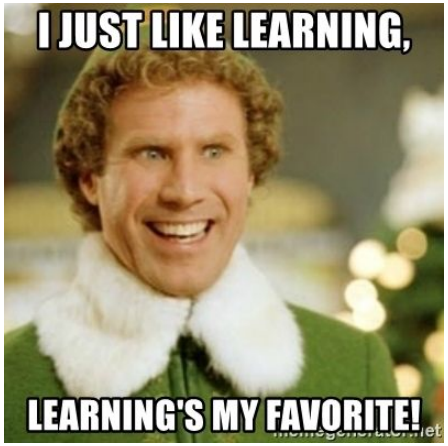
ps -eo pid,stat,time,sl,re,pagein,vsz,rss,lim,tsiz,%cpu,%mem,command (Ubuntu)

```
dis@dis-VirtualBox:/usr/bin$ ps -eo pid,stat,time,sl,re,pagein,vsz,rss,lim,tsiz,
%cpu,%mem,command
  PID STAT      TIME  SL  RE PAGEIN    VSZ   RSS   LIM TSIZ %CPU %MEM COMMAND
    1 Ss   00:00:06   -   -     90 127496  9432    xx    0  1.1  0.1 /sbin/init
    2 S    00:00:00   -   -      0      0     0    xx    0  0.0  0.0 [kthreadd]
```

Benefits

- User is able to quickly see what programs are using up the CPU
- Easier for the user to spot potential threatening programs that use up lots of the CPU

What we learned



- How commands interact with the operating system. Where things fall into place in the big picture
- Difference between system call and utility function
- VMWare is annoying (especially resizing the window)
- Always do sudo when rebuilding a kernel or else your rebuilding kernel will fail at 3am

Demonstration

Usage Instructions

```
dis@dis-VirtualBox:~$ hog

usage: hog [-p] n

     -p: instructs n to specify minimum required percentage of process
      n: instructs to return n top processes using up %CPU, in descend
ing order
   NOTE: if no -p, n will be floored
```

hog *n*

```
dis@dis-VirtualBox:~$ hog 10
PID                 %CPU                %MEM
1490                6.9                 4.7
1979                4.3                 4.1
1885                3.3                 4.5
2075                2.1                 5.3
1320                1.4                 3.8
1                   1.2                 0.1
1071                0.8                 0.4
2452                0.7                 1.0
2179                0.4                 1.4
630                 0.2                 0.3
```

hog *-p n*

```
dis@dis-VirtualBox:~$ hog -p 5
PID                 %CPU                %MEM
1490                9.6                 4.7
```

Edge cases covered

```
dis@dis-VirtualBox:~$ hog 100
n must be a nonnegative integer less than or equal to 20. You inputted
 100
```

```
dis@dis-VirtualBox:~$ hog -bj 5
Option unrecognized.

usage: hog [-p] n

     -p: instructs n to specify minimum required percentage of process
      n: instructs to return n top processes using up %CPU, in descend
ing order
   NOTE: if no -p, n will be floored
```