Contents

[**ps command** 2](#_Toc495652835)

[**Process Selection** 2](#_Toc495652836)

[**Output Format Control** 2](#_Toc495652837)

[**Standard format specifiers used with option -o** 3](#_Toc495652838)

[**Process state code** 4](#_Toc495652839)

[**Addition process code** 4](#_Toc495652840)

[ **RSS,VSZ & PSS calculation by an example** 4](#_Toc495652841)

[ **Terminal: tty & pts** 5](#_Toc495652842)

[ TTY: Teletypewriter 5](#_Toc495652843)

[ PTS: pseudo terminal slave 5](#_Toc495652844)

[ **Controlling terminal** 5](#_Toc495652845)

[ Output of ps command 5](#_Toc495652846)

[ Output of ps aux command 6](#_Toc495652847)

[ To find zombie process if any 6](#_Toc495652848)

[ To display process hierarchy in tree format 6](#_Toc495652849)

[ **Notes** 7](#_Toc495652850)

**ps command**:

* ps: process status
* It reports snapshot of currently running process sequentially.
* Process is an instance of an executable program whereas daemon is an instance of the service.
* Process mainly consists of PID, process state, security policies including ownership.
* Each process is assigned a unique ID and is called PID (Process ID).

|  |  |
| --- | --- |
| **Option** | **Description** |
| **Process Selection** | |
| -e OR -A | all processes |
| -a | all processes except both session leaders and processes not associated with a terminal |
| -d | all processes except session leaders |
| r | only running processes |
| -p OR --pid | Select processes by PID |
| --ppid | Select processes by PPID (Parent PID) |
| -U OR --User | select processes whose **Real** UID (**R**UID) is in userlist |
| -u OR --user | select processes whose **Effective** UID (**E**UID) is in userlist |
| -N OR --deselect | negate selection |
| **Output Format Control** | |
| -f | full format listing |
| -F | extra full format listing |
| -o | user defined format |
| -H | show process hierarchy (forest tree) |
|  |  |
| -M OR -Z | add column of security data |
| v | virtual memory format |
| --headers | report header lines, one per page of output |

|  |  |  |
| --- | --- | --- |
| **Standard format specifiers used with option -o** | | |
| **Code** | **Header** | **Description** |
| %cpu OR pcpu | %CPU | CPU utilization of the process |
| %mem OR pmem | %MEM | memory utilization of the process |
| args | COMMAND | command with all its arguments as a string |
| etime | ELAPSED | elapsed time since the process was started in the form [[DD-]hh:]mm:ss |
| etimes | ELAPSED | elapsed time since process was started in seconds |
| start\_time | START | starting time or date of the process |
| ni | NI | nice value |
| pid | PID | process ID |
| ppid | PPID | parent process ID |
| state | S | process state |
| rss OR rsz | RSS | resident set size non swapped physical memory that task has used (in KBs)  Doesn’t include memory that is swapped out |
| sz | SZ | size in physical pages of the core images of the process including text, data and stack space. |
| vsz | VSZ | virtual memory size of process in KB |
| user OR euser OR uname | USER | effective user |
| ruser | RUSER | real user |
| euid OR  uid | EUID | effective UID |
| egid OR gid | GID | effective GID |
| ruid | RUID | real UID |
| rgid | RGID | real GID |

Example: ps -eo user,pid,ppid,args

Example: ps -eo user,pid,ppid,rss,vsz,sz,args

Example: ps -eo user,pid,pcpu,pmem,args

Example: ps -eo user,pid,etime

Example: ps -p 3286 here, 3286 is PID

|  |  |
| --- | --- |
| **Process state code** | |
| **Code** | **Description** |
| R | running or runnable (on run queue) |
| S | interruptible sleep |
| D | uninterruptible sleep |
| T | stopped, either by a job control or because it is being traced |
| W | paging |
| X | dead |
| Z | zombie process |

|  |  |
| --- | --- |
| **Addition process code** | **Description** |
| < | high priority (not nice to other users) |
| N | low-priority (nice to other users) |
| L | pages locked in memory |
| s | session leader |
| l | multi-threaded |
| + | process is in the foreground process group |

|  |  |
| --- | --- |
| TTY | Controlling terminal |
| C | integer value of processor (CPU) utilization in percentage |
| WCHAN | kernel system call |
| TIME | Total accumulated CPU utilization time for a particular process 00:00:00 indicates no CPU time has been given for the particular process by the kernel till now.  E.g., cpu time by ps is also 00 because ps gets over in no time which could be milliseconds. |
| STIME | time when process started |

* To get PID by process name,

Command: pidof <process name>

## **RSS,VSZ & PSS calculation by an example**

RSS: Resident set size (actual physical memory used by process)

VSZ: Virtual memory

PSS: Proportional set size

If process A has

* 500K binary, linked to 2500K of shared libraries, 200K of stack/heap allocation

Out of 200K stack, only 100K is actually in memory (rest is swapped out)

Out of 2500K shared libraries, only 1000K is loaded

Out of 500K binary, only 400K is loaded

* RSS=400K + 1000K + 100K =1500K
* VSZ=500K + 2500K + 200K= 3200K
* If two processes A & B are using shared memory,

PSS=400K + (1000K/2) + 100K =1000K

## **Terminal: tty & pts**

* There are two types of terminals:

1. terminal devices (tty)
2. pseudo terminal devices (pts)

### TTY: Teletypewriter

Regular terminal in which user type commands.

Direct connection to computer such as keyboard/mouse or serial port connection to device.

Ctrl+Alt+F1 to F7

### PTS: pseudo terminal slave

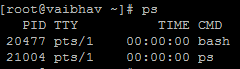
GUI windows

SSH, telnet connections.

Ctrl+Shift+t

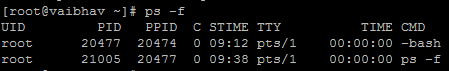
## **Controlling terminal**

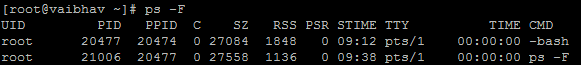
* Controlling terminal is the login session in which process/program is invoked.
* Each process is bound to its controlling terminal. When its terminal is closed, process associated it, also closed (killed).
* If you login to system to start the bash session, shell’s controlling terminal is your login session.
* Daemons which run in background, start at system boot and get terminated at system shutdown. Daemons are not bound to any terminal. So, in the ps output, TTY is listed as question mark (**?**) in terminal column.
* Output of ps command,



By default, ps command with no options selects all processes with the same effective UID as the current user and associated with the same terminal where ps was invoked.

* ps command with full (-f option)and extra full (-F option) information,





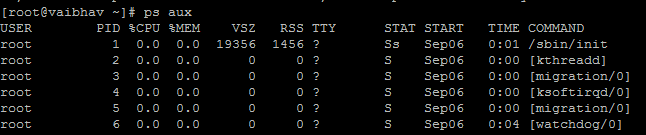
* Output of ps aux command,

Command: ps aux

a: show processes for all users

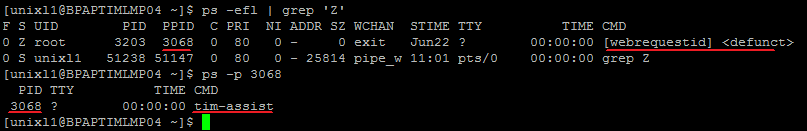
u: display process’s user/owner

x: show processes not attached to the terminal



* To find zombie process if any,

Command: ps -efl | grep ‘Z’



* To display process hierarchy in tree format,

Command: pstree <PID/user name>

Example: pstree Display all processes in hierarchical structure

Example: pstree 3203

Example: pstree root Display processes owned by root user in hierarchical structure

Example: pstree -a 3203 Display command line arguments associated with particular process

Example: pstree -H 3203 Highlight particular process

* **Notes**:
* UNIX95: alters behavior of ps command execution
* Processes in brackets [] are scheduled kernel thread.
* In output of ps aux, all processes are displayed sequentially by PID.
* In output of ps command, recent processes are at bottom end whereas in output of top command, recent processes are at top end.

|  |  |
| --- | --- |
| etime | elapsed time since process was started in the form of [[dd-]hh]:mm:ss |
| start | time the command started If less than 24 hours, in HH:MM:SS If more than 24 hours, in mmm dd format; mmm:month,dd:day |
| lstart | time command started |
|  |  |
|  |  |