LPI 103.5 Create monitor and kill processes
LPI 103.6 Modify process execution priorities

Curs 2021 - 2022

ASIX M01-ISO LPI 103-GNU_and_unix_commands

Administració de processos

Descripció

Ordres a treballar:

- ❏ ps, pstree, pgrep, pidof
- ❏ watch, time
- ❏ kill, killall, pkill
- ❏ SIGNALS 15, 9, 1, 2, 20, 18, 19
- ❏ &, jobs, bg, fg
- ❏ nohup

- ❏ nice, renice
- ❏ top, free, uptime, vmstat

Gestió de processos

Processes: ps, pstree, pgrep, watch

ps

Two notations for options BSD and GNU:

- ps
- ps ax
- ps aux
- ps -ef
- ps -l

Common options:

- a Allows the ps command to show all processes.
- u Shows processes by all users and ignores restrictions to only list the current user's processes.
- x Lists all processes and removes the restriction to only display the processes that are running in the current terminal.
- -e every process
- -f full details
- -l list long format

```
$ ps
    PID TTY          TIME CMD
   5230 pts/0    00:00:00 bash
   5236 pts/0    00:00:00 ps

$ ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S 100366   5230    5223  0  80   0 -  4377 -    pts/0    00:00:00 bash
4 R 100366   5244    5230  0  80   0 -  2405 -    pts/0    00:00:00 ps

$ ps a
    PID TTY      STAT   TIME COMMAND
   2952 tty2     Ss+    0:00 /sbin/agetty -o -p -- \u --noclear tty2 linux
   3413 tty3     Ssl+   0:00 /usr/libexec/gdm-x-session --run-script /usr/bin/gnome-session
   3415 tty3     Sl+    0:40 /usr/lib/xorg/Xorg vt3 -displayfd 3 -auth
/run/user/100366/gdm/Xauthority -nolisten tcp -backgroun
   3431 tty3     Sl+    0:00 /usr/libexec/gnome-session-binary --systemd
   5230 pts/0    Ss     0:00 bash
   5262 pts/1    Ss+    0:00 bash
   5270 pts/1    S      0:00 sleep 123456789
   5271 pts/1    S      0:00 sleep 22222222
   5278 pts/0    R+     0:00 ps a

$ ps -ef | head
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 09:55 ?        00:00:02 /sbin/init
root           2       0  0 09:55 ?        00:00:00 [kthreadd]
root           3       2  0 09:55 ?        00:00:00 [rcu_gp]
```

```
root           4       2  0 09:55 ?          00:00:00 [rcu_par_gp]
root           6       2  0 09:55 ?          00:00:00 [kworker/0:0H-events_highpri]
root           8       2  0 09:55 ?          00:00:00 [mm_percpu_wq]
root           9       2  0 09:55 ?          00:00:00 [rcu_tasks_rude_]
root           10      2  0 09:55 ?          00:00:00 [rcu_tasks_trace]
root           11      2  0 09:55 ?          00:00:00 [ksoftirqd/0]
```

Stat:

- D Uninterruptible Sleep
- R Running
- S Interruptible Sleep
- T Stopped
- Z Zombie

pgrep

- -l list name
- -i ignore case
- -u user

```
$ pgrep sleep
5270
5271

$ pgrep sleep -l
5270 sleep
5271 sleep

$ pgrep  -li BASH
5230 bash
5262 bash

$ pgrep systemd -l
1 systemd
253 systemd-journal
291 systemd-udevd
314 systemd-timesyn
571 systemd-logind
3294 systemd

$ pgrep systemd -l
1 systemd
253 systemd-journal
291 systemd-udevd
314 systemd-timesyn
571 systemd-logind
3294 systemd
```

pstree

```
$ pstree  | head
systemd-+-ModemManager---2*[{ModemManager}]
        |-NetworkManager---2*[{NetworkManager}]
        |-accounts-daemon---2*[{accounts-daemon}]
        |-agetty
        |-automount---2*[{automount}]
        |-avahi-daemon---avahi-daemon
        |-colord---2*[{colord}]
        |-containerd---8*[{containerd}]
```

```
       |-cron
       |-dbus-daemon


$ pstree -lap  | head
systemd,1
  |-ModemManager,500
  |   |-{ModemManager},520
  |   `-{ModemManager},522
  |-NetworkManager,454 --no-daemon
  |   |-{NetworkManager},501
  |   `-{NetworkManager},503
  |-accounts-daemon,569
  |   |-{accounts-daemon},582
  |   `-{accounts-daemon},584
```

```
$ ps
       PID TTY        TIME CMD
   5230 pts/0  00:00:00 bash
   5784 pts/0  00:00:00 sleep
   5785 pts/0  00:00:00 sleep
   5844 pts/0  00:00:00 sleep
   6237 pts/0  00:00:00 sleep
   6426 pts/0  00:00:00 ps


$ pstree -spl 5230
systemd(1)───systemd(3294)───gnome-terminal-(5223)───bash(5230)─┬─pstree(6429)
                                                                ├─sleep(5784)
                                                                ├─sleep(5785)
                                                                ├─sleep(5844)
                                                                └─sleep(6237
```

```
      CODE   NORMAL   HEADER
       %C     pcpu     %CPU
       %G     group    GROUP
       %P     ppid     PPID
       %U     user     USER
       %a     args     COMMAND
       %c     comm     COMMAND
       %g     rgroup   RGROUP
       %n     nice     NI
       %p     pid      PID
       %r     pgid     PGID
       %t     etime    ELAPSED
       %u     ruser    RUSER
       %x     time     TIME
       %y     tty      TTY
       %z     vsz      VSZ


$ ps -o pid,ppid,user,%cpu,cmd
    PID    PPID USER     %CPU CMD
   4283    4184 ecanet    0.0 bash
   4385    4283 ecanet    0.0 sleep 666666
   5177    4283 ecanet    0.0 vim /tmp/carta
   6268    4283 ecanet    0.0 sleep 12345
   6680    4283 ecanet    0.0 sleep 22332233
   7167    4283 ecanet    0.0 sleep 22332233
   7187    4283 ecanet    0.0 sleep 22332233
   8318    4283 ecanet    0.0 ps -o pid,ppid,user,%cpu,cmd
```

Watch

- 2s default
- -n nº seconds

- ^c
- -d diference

```
# watch date
Every 2.0s: date
       d02: Thu Oct 21 10:29:44 2021

Thu 21 Oct 2021 10:29:44 AM CEST

# watch ps a
# des d'una altra consola llançar processos, per exemple sleep


# watch free


# watch du -sh /var/tmp/img
# des d'una altra consola generar un disc imatge
# dd if=/dev/zero of=disc.img bs=1k count=2M
```

time

```
$ cp /usr/bin/ls /tmp/

$ time gzip /usr/bin/ls
gzip: /usr/bin/ls.gz: Permission denied
real    0m0.002s
user    0m0.000s
sys     0m0.002s

$ time tree &> /dev/null
real    0m0.013s
user    0m0.001s
sys     0m0.004s

$ time tree &> /tmp/tree.txt
real    0m0.005s
user    0m0.002s
sys     0m0.003s
```

Signals: kill killall pkill

Signals:

- 1 SIGHUP HUP Hang up, usually ends a process
- 2 SIGINT INT Interrupt, usually ends a process
- 3 SIGQUIT QUIT Quit, usually ends a process
- 9 SIGKILL KILL Kill, forcefully ends a process
- 15 SIGTERM TERM Terminate, usually ends a process
- 18 SIGCONT CONT Continue, resumes a stopped process
- 19 SIGSTOP STOP Stop, forcefully stops a process
- 20 SIGTSTP TSTP Terminal Stop, usually stops a process

```
$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT      7) SIGBUS      8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT     17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
```

```
21) SIGTTIN    22) SIGTTOU    23) SIGURG    24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM    27) SIGPROF    28) SIGWINCH    29) SIGIO    30) SIGPWR
31) SIGSYS    34) SIGRTMIN    35) SIGRTMIN+1    36) SIGRTMIN+2    37) SIGRTMIN+3
38) SIGRTMIN+4    39) SIGRTMIN+5    40) SIGRTMIN+6    41) SIGRTMIN+7    42) SIGRTMIN+8
43) SIGRTMIN+9    44) SIGRTMIN+10    45) SIGRTMIN+11    46) SIGRTMIN+12    47)
SIGRTMIN+13
48) SIGRTMIN+14    49) SIGRTMIN+15    50) SIGRTMAX-14    51) SIGRTMAX-13    52)
SIGRTMAX-12
53) SIGRTMAX-11    54) SIGRTMAX-10    55) SIGRTMAX-9    56) SIGRTMAX-8    57) SIGRTMAX-7
58) SIGRTMAX-6    59) SIGRTMAX-5    60) SIGRTMAX-4    61) SIGRTMAX-3    62) SIGRTMAX-2
63) SIGRTMAX-1    64) SIGRTMAX
```

Kill

- -1
- -HUP
- -SIGHUP

```
$ ps
       PID TTY        TIME CMD
   5230 pts/0  00:00:00 bash
   5784 pts/0  00:00:00 sleep
   5785 pts/0  00:00:00 sleep
   5844 pts/0  00:00:00 sleep
   6237 pts/0  00:00:00 sleep
   6484 pts/0  00:00:00 ps

$ jobs
[1]   Running                 sleep 111111111 &
[2]   Running                 sleep 222222222 &
[3]-  Running                 sleep 333333333 &
[4]+  Running                 sleep 12345678 &

$ kill %4

$ kill 5844
[4]+  Terminated              sleep 12345678

$ jobs
[1]   Running                 sleep 111111111 &
[2]-  Running                 sleep 222222222 &
[3]+  Terminated              sleep 333333333
```

```
$ ps
       PID TTY        TIME CMD
   5230 pts/0  00:00:00 bash
   5784 pts/0  00:00:00 sleep
   5785 pts/0  00:00:00 sleep
   6502 pts/0  00:00:00 ps

$ killall sleep
[1]-  Terminated              sleep 111111111
[2]+  Terminated              sleep 222222222
```

```
$ sleep 123456 &
[1] 6509
$ sleep 567890 &
[2] 6515

$ jobs
[1]-  Running                 sleep 123456 &
[2]+  Running                 sleep 567890 &

$ pgrep -l lee
6509 sleep
6515 sleep
```

```
$ pkill lee
[1]-  Terminated              sleep 123456
[2]+  Terminated              sleep 567890
```

```
$ ps a
        PID TTY        STAT   TIME COMMAND
    2952 tty2   Ss+    0:00 /sbin/agetty -o -p -- \u --noclear tty2 linux
    3413 tty3   Ssl+   0:00 /usr/libexec/gdm-x-session --run-script /usr/bin/gnome-session
    3415 tty3   Sl+    1:54 /usr/lib/xorg/Xorg vt3 -displayfd 3 -auth
/run/user/100366/gdm/Xauthority -nolisten tcp -backgroun
    3431 tty3   Sl+    0:00 /usr/libexec/gnome-session-binary --systemd
    5230 pts/0  Ss     0:00 bash
    5262 pts/1  Ss+    0:00 bash
    5349 pts/2  Ss+    0:00 bash
    6541 pts/0  R+     0:00 ps a

$ kill 5262

$ ps -l 5262
F S   UID      PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S 100366   5262    5223  0  80   0 -  4608 -    pts/1   0:00 bash
```

```
$ kill -TERM 5262
$ ps -l 5262
F S   UID      PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S 100366   5262    5223  0  80   0 -  4608 -    pts/1   0:00 bash

$ kill -9 5262

$ ps -l 5262
F S   UID      PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
```

Background: & jobs bg fg

- command; command
- command &
- +
- -
- %n job number
- ^z
- foreground: apropiative console
- background: desirable no stdout and no stderr

```
$ sleep 111111111 &
[1] 5784

$ sleep 222222222 &
[2] 5785

$ sleep 333333333  &
[3] 5844


$ jobs
[1]   Running                 sleep 111111111 &
[2]-  Running                 sleep 222222222 &
[3]+  Running                 sleep 333333333 &
```

```
$ fg
```

```
sleep 333333333
^Z
[3]+  Stopped                 sleep 333333333

$ jobs
[1]   Running                 sleep 111111111 &
[2]-  Running                 sleep 222222222 &
[3]+  Stopped                 sleep 333333333
```

```
$ sleep 12345678
^Z
[4]+  Stopped                 sleep 12345678

$ jobs
[1]   Running                 sleep 111111111 &
[2]   Running                 sleep 222222222 &
[3]-  Stopped                 sleep 333333333
[4]+  Stopped                 sleep 12345678

$ bg -
[3]- sleep 333333333 &

$ jobs
[1]   Running                 sleep 111111111 &
[2]   Running                 sleep 222222222 &
[3]-  Running                 sleep 333333333 &
[4]+  Stopped                 sleep 12345678

$ fg +
sleep 12345678
^Z
[4]+  Stopped                 sleep 12345678

$ bg %4
[4]+ sleep 12345678 &

$ jobs
[1]   Running                 sleep 111111111 &
[2]   Running                 sleep 222222222 &
[3]-  Running                 sleep 333333333 &
[4]+  Running                 sleep 12345678 &
```

```
$ tree / > /tmp/tree.txt 2> /dev/null &
[5] 6297


# this command should not generate errorsat the console
$ find / -size +1M -print > /tmp/size.txt &
find: ‘/lost+found’: Permission denied
find:
‘/home/groups/inf/inf/repositori/Credits/zDAI/DAI-C2/DAI-C2_Curs-0708/c2-groups_0708/UD2
/A2/wida12236/exer1formulari’: Permission denied

$ find / -size +1M -print > /tmp/size.txt 2> /dev/null &
```

- foreground is console appropriative

```
# edit carta and press ^z

$ vim /tmp/carta
[2]+  Stopped                 vim /tmp/carta


$ jobs
[1]-  Running                 nohup sleep 666666 &
```

```
[2]+  Stopped                 vim /tmp/carta

# can not restart in background, vim is console appropriative
$ bg
[2]+ vim /tmp/carta &

[2]+  Stopped                 vim /tmp/carta
```

nohup

- When a user logs off the system, all processes that are owned by that user are automatically sent the Hang Up SIGHUP signal. Typically, this signal causes those processes to end.
- In some cases, a user may want to execute a command that won't automatically exit when it is sent a HUP signal. To have a process ignore a Hang Up signal, start the process with the nohup command.

in a text console / then close the console

```
$ nohup sleep 666666 &
[1] 4385
$ nohup: ignoring input and appending output to 'nohup.out'
```

```
$ ps ax
 4385 ?        S      0:00 sleep 666666
```

Priority (nice): nice renice

- default nice priority 0
- [-20 0 19] -20=max 19=min
- User only from 0 to 19. Root from -20 to 20. Only root negative (more) priority.

```
$ sleep 12345 &
[3] 6268

$ ps -l 6268
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY        TIME CMD
0 S  1001    6268    4283  0  80   0 - 53824 -      pts/1      0:00 sleep 12345
```

```
$ renice -5 6268
renice: failed to set priority for 6268 (process ID): Permission denied

$ renice 5 6268
6268 (process ID) old priority 0, new priority 5

$ ps -l 6268
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY        TIME CMD
0 S  1001    6268    4283  0  85   5 - 53824 -      pts/1      0:00 sleep 12345

$ renice 20 6268
```

```
6268 (process ID) old priority 5, new priority 19
$ ps -l 6268
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY        TIME CMD
0 S  1001    6268    4283  0  99  19 - 53824 -      pts/1      0:00 sleep 12345
```

```
$ nice -15 sleep 22332233 &
[4] 6680

$ ps -l 6680
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY        TIME CMD
0 S  1001    6680    4283  0  95  15 - 53824 -      pts/1      0:00 sleep 22332233
```

```
$ nice -n 3 sleep 22332233 &
[6] 7187

$ renice 0 7187
renice: failed to set priority for 7187 (process ID): Permission denied

$ renice 5 7187
7187 (process ID) old priority 3, new priority 5
```

General information: top, free, uptime

```
$ uptime
 16:27:18 up 34 min,  1 user,  load average: 0.70, 0.53, 0.46
```

```
$ free
               total        used        free      shared  buff/cache   available
Mem:         7648128     2360260     1577772      672780     3710096     4305240
Swap:        7811068           0     7811068

$ free -h
               total        used        free      shared  buff/cache   available
Mem:           7.3Gi       2.3Gi       1.5Gi       630Mi       3.5Gi       4.1Gi
Swap:          7.4Gi          0B       7.4Gi
```

```
top - 16:27:58 up 34 min,  1 user,  load average: 0.49, 0.49, 0.45
Tasks: 280 total,   1 running, 278 sleeping,   1 stopped,   0 zombie
%Cpu(s):  2.0 us,  0.8 sy,  0.0 ni, 96.5 id,  0.0 wa,  0.4 hi,  0.3 si,  0.0 st
MiB Mem :   7468.9 total,   1592.1 free,   2279.2 used,   3597.6 buff/cache
MiB Swap:   7628.0 total,   7628.0 free,      0.0 used.   4256.8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2071 ecanet    20   0 4656988 186508 103600 S   6.6   2.4   1:00.19 gnome-shell
   1893 ecanet    20   0 1379308  80008  45940 S   3.3   1.0   0:55.23 Xorg
   3829 ecanet    20   0   36.6g 272548 107516 S   1.0   3.6   3:21.16 chrome
   2838 ecanet    20   0   32.5g 112204  85140 S   0.7   1.5   0:24.45 chrome
     14 root      20   0       0      0      0 I   0.3   0.0   0:01.27 rcu_sched
    722 root       0 -20       0      0      0 I   0.3   0.0   0:02.38 kworker/u9:2-i915_flip
   2776 ecanet    20   0   32.7g 255940 168972 S   0.3   3.3   1:10.81 chrome
   5106 root      20   0       0      0      0 I   0.3   0.0   0:01.37 kworker/0:3-events
      1 root      20   0  173316  16228  10460 S   0.0   0.2   0:01.51 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
```

- Pressing the K key will allow a user to kill or send a signal to a process. After pressing the K key, the top command will prompt for a PID and then for a signal to send to that process.
- Pressing the R key will allow a user to renice a process by prompting for the PID and then the new niceness value.
- Press the Q key to quit the top command.
- M Sort by memory usage.
- N Sort by process ID number.
- T Sort by running time.
- P Sort by percentage of CPU usage.

Exercicis d'exemple

1. Github LPIC-1 103.5-Exercices.md
2. Github LPIC-1 103.6-Exercices.md
3. LPI Exercices 103.5 Create, monitor and kill processes

Exercices

1. List all the processes using *ps ax*, and using *ps -ef*.
2. Show all the processes hirarchy tree including PID and command.
3. Using pgrep shoow al the root user's processes.
4. Using the pidof command show the systemd PID.
5. Using pstree show all the 'genealogy' of the current shell (from systemd to the shell)
6. Llist all the PID processes staring by the name "sys".
7. Show the pid, user, %cpu and command of the processes.

[ ]

8. Use watch to monitor the processes list
9. Use whatch tho monitor the date command every 3 seconds showing the diferences
10. Compute the time to execute a full tree list of the system.

[ ]

11. List al the kill signals
12. Execute one sleep comamnd and kill it.
13. Enter in a subshell and kill it.
14. Start 3 sleep commands and kill all of them by name.

[ ]

15. Start 3 sleep commands in background. List the process in background.
16. Pass the second sleep process from background to foreground.
17. Now stop (no kill) the sleep process that is in foregroung.
18. Lists the jobs and observe the state of the previous sleep. Restart its execution in backgrond.
19. Which signal is ^c?

[ ]

Exercices of nice and renice commands are in 103.6_Exercicces.md
`postponed`

20.

[ ]

21. Wiitch tool monitors the process?
22. Show the memory information in human readable information.
23. For how long is the system running?

Alternate exercises:

1. Mostrar tots els processos del sistema.
2. Mostrar tot l'arbre de processos incloent el pid i la ordre.
3. Prova les ordres: ps, ps a, ps x, ps ax, ps -fu pere, ps -Fu pere.
4. Llistar els processos fent un llistat llarg on mostri el PID i el PPID.
5. Entrar en un subshell i fer un llistat llarg dels processos.
6. Identificar el PID del procés pare del shell actual.
7. Identifica el PID del procés systemd usant l'ordre pidof.
8. Identifica el pid del servei d'impressió cupsd amb l'ordre pidof.
9. Usant l'ordre pgrep llista els processos de l'usuari root.
10. Usant l'ordre pgrep localitza el procés systemd.
11. Utilitzant l'ordre fuser per saber quins processos utilitzen el directori /tmp. I quins utilitzen l'arrel del sistema?
12. Llista tots els senyals de l'ordre kill.
13. Genera un procés sleep 10000 i mata'l amb kill.
14. Mata el bash actual.
15. Llista tots els processos sleep i mata'ls de cop tots usant una sola ordre tipus kill. Per crear varis processos sleep fes: sleep 123456789 & almenys tres cops.

[ ]

16. Executa tres ordres sleep en segon pla i llista els treballs.

17. Inicia l'edició d'un fitxer amb vi i deixa'l suspès d'execució en segon pla. Mostrar els treballs.
18. Mata el segon dels treballs (un sleep).
19. Passa a primer pla el primer dels treballs (un sleep), i mata'l amb ctrl+c.
20. Passa a primer pla el treball més recent. Quin és. Acabar.
21. Llistar tota l'estructura de directoris partint de l'arrel (amb tree). Que no es generin missatges d'error i enviar la sortida al fitxer tree.txt. Un cop iniciat aturar el procés (no matar). Llistar els treballs.
22. Reanudar l'execució del tree anterior en segon pla.

[ ]

23. Executar l'ordre que monitoritza els processos. Llistar-los per prioritat.
24. Executar l'ordre vmstat. Descriu almenys tres dels elements dels que informa.
25. Executar l'ordre free i descriure la informació que mostra.
26. Digues quanta estona fa que el sistema està engegat ininterrompudament.