

Pràctica 8: Programes i processos (Primera part)

En aquesta pràctica veuràs com realitzar una gestió acurada dels processos del sistema, tant a Linux com a Windows. Aprenderàs una col·lecció de comandes que t'ajudarà realitzar aquesta gestió mitjançant la terminal.

Processos a Windows

Com ja sabeu, per accedir a l'administrador de tasques de Windows cal executar la comanda **taskmgr** al menú **Inicio>Ejecutar**.

1. L'administrador de tasques té diversos punts d'entrada. Executa totes les maneres possibles per a accedir a l'administrador de tasques.

Ctrl+Alt+Supr>Administrador de Tareas
Tecla Windows+R i escrivim taskmgr
Ctrl+Shift+Esc
Escriure taskmgr.exe al CMD

2. Executeu un programa com, per exemple, la calculadora o el bloc de notes i indiqueu el nom de l'executable, quin consum de CPU i quin ús de memòria té?

calc
0% de consum de CPU
2,3 MB d'ús de memòria

3. Com s'elimina un procés explorer (explorador d'arxius) al terminal? Elimineu el procés i vegeu què passa. Com podem solucionar aquesta situació?

taskkill /f /im explorer.exe
explorer.exe

4. Quin és el procés **IDLE** a Windows? Quin és el seu comportament? Com podeu mostrar, mitjançant **Get-Process** el procés IDLE? Intenteu eliminar-lo, mitjançant la comanda **Stop-Process** què passa?

System Idle Process
Significa el percentatge restant de processador que està lliure.

```
PS C:\Users\apere> get-process idle
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
0	0	60	8		0	0	Idle

No es pot matar l'Idle

5. Com podem mostrar, mitjançant la comanda Get-Process, l'usuari que ha executat cada procés i el pid de cadascun dels processos?

Get-Process Calculator -IncludeUserName

6. Indiqueu el significat de les columnes quan llancem la comanda **Get-Process**:

Veiem tots els processos, les seves IDs, el seu ús de la CPU i més.

7. Com podem mostrar, mitjançant la comanda **Get-Process**, la versió del procés que estem executant?

Get-Process Calculator

8. Com podem mostrar, mitjançant la comanda **Get-Process**, tots els processos que comencen el seu nom amb la cadena 'win' i 'ex'?

```
PS C:\Users\apere> get-process -name ex*, win*
```

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
2552	89	79652	133816	11,41	9904	2	explorer
164	11	1504	7340		664	0	wininit
275	12	2436	10056		8216	2	winlogon

9. Mitjançant la consola PowerShell, llenceu el programa 'Notepad'. A continuació, atureu-lo, també des de Powershell mitjançant el PID del programa:

```
PS C:\Users\apere> start notepad
PS C:\Users\apere> get-process
```

448	17	13732	23136	84,08	1332	2	notepad
-----	----	-------	-------	-------	------	---	---------

```
PS C:\Users\apere> taskkill /f /im 1332
Correcto: se terminó el proceso con PID 1332.
```

Processos a Unix/Linux

En aquesta part de la pràctica us remetem a les explicacions que pertanyen al material teòric d'aquest nucli d'activitat, per a que resolgueu les activitats següents:

10. Indiqueu què fa cadascun dels modificadors de la comanda ps següents: **-l, -f, -u, -p, -ef, -e, -C, -o, -aux, -eo**.

- l: mostra informació que inclou el UID i el valor nice.
- f: format "forest" de famílies en forma d'arbre.
- u: llista de informació dels processos.
- p: enumera els detalls dels processos.
- ef: agregar columnes a la sortida.
- e: mostra ambient després del nom de la comanda i "+".
- C: permet cercar un process utilitzant el nom de la comanda que va iniciar el procés.
- o: format definit per l'usuari.
- aux: per veure cada procés en el sistema.
- eo: per tenir informació de seguretat.

11. Escriviu una comanda que llisti tots els processos de l'usuari root.

```
alumne@alumne-VirtualBox:~$ ps -U root -u root
  PID TTY          TIME CMD
    1 ?           00:00:05 systemd
    2 ?           00:00:00 kthreadd
    3 ?           00:00:00 rcu_gp
    4 ?           00:00:00 rcu_par_gp
    6 ?           00:00:00 kworker/0:0H-kblockd
    7 ?           00:00:00 kworker/0:1-events
    8 ?           00:00:00 kworker/u2:0-events_power_efficient
    9 ?           00:00:00 mm_percpu_wq
   10 ?           00:00:00 ksoftirqd/0
   11 ?           00:00:00 rcu_sched
   12 ?           00:00:00 migration/0
   13 ?           00:00:00 idle_inject/0
   14 ?           00:00:00 cpuhp/0
   15 ?           00:00:00 kdevtmpfs
   16 ?           00:00:00 netns
   17 ?           00:00:00 rcu_tasks_kthre
```

12. Entreu al manual de la màquina (per exemple `man ps`) i premeu `Ctrl+z`. Què ha passat?
Feu el necessari per a continuar la consulta del manual i indiqueu-ho.

`Ctrl+z` per suspendre processos.
Amb `fg` tornam al manual.

13. Inicieu el procés **man find** i suspeneu-lo amb **Ctrl+Z**.

Executeu **xclock** en background

Utilitzeu `jobs` per a llistar els processos en segon pla i els processos aturats.

Utilitzeu la comanda `fg` per a portar el `man find` a primer pla. Sortiu normalment amb `q`

Utilitzeu `fg` per a portar `xclock` al foreground i acabeu-lo amb `Ctrl+c`.

Executeu **xclock** novament, però aquesta vegada en primer pla. (no podreu utilitzar la shell). Suspeneu-lo amb **Ctrl+Z** i descriviu què passa..

No es pot suspendre el `xclock`

14. Obriu un terminal i executeu la comanda `gedit`. Des d'un altre terminal executeu el `pstree` i vegeu qui és el procés pare de `gedit`. Mateu el pare de `gedit`. Què passa? Torneu a fer el mateix, però en aquest cas executeu `gedit` en segon pla. Mateu el pare. Què passa ara?

El pare de `gedit` es `bash`.

Es tanca el bloc de notes i la terminal amb la que l'hem executat.

Es tanca la terminal però no el programa.

15. Com podríem matar un procés a partir del seu número de tasca/job?

Cercam el seu número utilitzant `ps -ef`

S coneixem el nom del procés, podem fer `ps -ef | grep "nomdelproces"`

16. Apliqueu els paràmetres necessaris, de forma interactiva, a la comanda `top` per a que mostri únicament els processos de l'usuari `root`, els ordeni descendentment per ocupació de memòria i actualitzi les dades cada 0,1 segons.

```
alumine@alumne-VirtualBox:~$ top -o %MEM -d 0.1 -u root
```

top - 16:48:18 up 37 min, 1 user, load average: 0,35, 0,14, 0,16
Tareas: 166 total, 1 ejecutar, 165 hibernar, 0 detener, 0 zombie
%Cpu(s): 23,1 usuario, 7,7 sist, 0,0 adecuado, 69,2 inact, 0,0 en espera,
MiB Mem : 1987,8 total, 579,9 libre, 653,4 usado, 754,4 búfer/cach
MiB Intercambio: 2048,0 total, 2048,0 libre, 0,0 usado. 1173,4 dis

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
1477	root	20	0	630116	31968	16024	S	0,0	1,6	0:04.10	snapt
607	root	20	0	120688	22724	14612	S	0,0	1,1	0:00.17	unat+
220	root	19	-1	68300	22252	20636	S	0,0	1,1	0:00.91	syst+
482	root	20	0	489332	21912	18892	S	0,0	1,1	0:11.50	Netw+
495	root	20	0	41924	20060	11828	S	0,0	1,0	0:00.29	netw+
516	root	20	0	395512	14232	11860	S	0,0	0,7	0:00.22	udis+
506	root	20	0	248160	13132	9820	S	0,0	0,6	0:00.47	polk+
1	root	20	0	103368	12856	8392	S	0,0	0,6	0:05.76	syst+
575	root	20	0	180424	12692	11048	S	0,0	0,6	0:00.07	cups+
572	root	20	0	313728	10700	9040	S	0,0	0,5	0:00.14	Mode+
615	root	20	0	317124	10008	8524	S	0,0	0,5	0:00.07	gdm-+
470	root	20	0	244804	9728	8676	S	0,0	0,5	0:00.17	acco+

17. Quina prioritat assigna l'ordre nice quan no indiquem aquest valor de prioritat? Creeu un procés amb la màxima prioritat possible i un amb la mínima possible utilitzant aquesta comanda. Com podem consultar la prioritat d'aquest procés?

Quan no indicam el valor de prioritat sempre tindrem prioritat 10.

18. Creeu un procés amb una prioritat molt baixa i modifiqueu-li-la utilitzant la comanda **renice**. Com faríeu per a modificar la prioritat de tots els vostres processos. Quina opció de l'ordre top us permet dur a terme aquesta mateixa acció.

```
nice --adjustment=19 vim
renice n -20 -p 5500
```

19. Quina és la funció de les comandes **pidof**, **pgrep** i **pkill**.

pidof: per trobar l'identificador de procés d'un programa que s'està executant.
pgrep: recull una expressió regular de la línia de comandes i ens mostra el/els ID dels processos que coincideixen amb aquesta expressió regular.
pkill: per enviar una senyal a processos.