

## Practica 1

Programa: Es el conjunt d'instruccions escrites per a executar unes tasques determinades.

Proces: Es el conjunt de operacions que es realitzen durant l'execució d'un programa.

Executem top -i i esbrinem que el pid de firefox es 4851

```
top - 09:54:22 up 4 min, 1 user, load average: 0,22, 0,23, 0,10
Tareas: 342 total, 1 ejecutar, 341 hibernar, 0 detener, 0 zombie
%Cpu(s): 1,1 usuario, 0,9 sist, 0,0 adecuado, 97,8 inact, 0,0 en espera, 0,0 hardw in
MiB Mem : 15854,0 total, 11965,4 libre, 2104,5 usado, 1784,2 búfer/caché
MiB Intercambio: 2048,0 total, 2048,0 libre, 0,0 usado. 13370,6 dispon Mem
```

%CPU	PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%MEM	HORA+	ORDEN
12,0	1503	root	20	0	868736	176092	114208	S	1,1	0:16.16	Xorg
6,6	1858	pakels	20	0	321012	71692	53356	S	0,4	0:04.32	xfwm4
6,0	5532	pakels	20	0	239084	31968	26840	S	0,2	0:00.42	xfce4-screensho
2,0	1511	root	-51	0	0	0	0	S	0,0	0:02.22	irq/145-nvidia
1,0	4694	pakels	20	0	1573704	45264	35028	S	0,3	0:01.44	Arduino_Create_
1,0	4996	pakels	20	0	3303008	559768	96800	S	3,4	0:32.99	WebExtensions
0,3	11	root	20	0	0	0	0	I	0,0	0:00.30	rcu_sched
0,3	993	root	20	0	1936452	44756	24048	S	0,3	0:00.57	containerd
0,3	4607	pakels	20	0	412356	50272	40968	S	0,3	0:00.54	xfce4-terminal
0,3	4851	pakels	20	0	3307696	384380	204160	S	2,4	0:41.04	firefox
0,3	5119	pakels	20	0	2578852	181668	101028	S	1,1	0:10.92	Web Content
0,3	5226	pakels	20	0	2531696	177012	97812	S	1,1	0:02.80	Web Content
0,3	5279	pakels	20	0	2525728	172740	98192	S	1,1	0:02.89	Web Content
0,3	5429	pakels	20	0	7783724	306852	160352	S	1,9	0:02.41	soffice.bin

Acabem el proces amb la ordre:

```
kill -9 4851
```

Si executem nice, retorna 0.

Executem >nice -n 5 firefox i comprovem la seua prioritat:

```
Tareas: 330 total, 1 ejecutar, 329 hibernar, 0 detener, 0 zombie
%Cpu(s): 0,1 usuario, 0,2 sist, 0,6 adecuado, 99,1 inact, 0,0 en espera, 0,0 hardw in
MiB Mem : 15854,0 total, 11810,8 libre, 1722,1 usado, 2321,2 búfer/caché
MiB Intercambio: 2048,0 total, 2048,0 libre, 0,0 usado. 13744,3 dispon Mem
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
7067	pakels	25	5	3063324	334564	176612	S	8,3	2,1	0:03.30	firefox
7202	pakels	25	5	3093576	458336	96532	S	1,0	2,8	0:09.48	WebExtensions
1503	root	20	0	868264	175728	113844	S	0,7	1,1	0:34.50	Xorg
4694	pakels	20	0	1573960	45264	35028	S	0,7	0,3	0:05.15	Arduino_Create_
656	root	-51	0	0	0	0	S	0,3	0,0	0:00.19	irq/138-iwlwifi
857	systemd+	20	0	24024	13120	8992	S	0,3	0,1	0:00.24	systemd-resolve
900	message+	20	0	8852	5880	3868	S	0,3	0,0	0:00.55	dbus-daemon
1390	root	20	0	1652616	10784	9300	S	0,3	0,1	0:01.28	teamviewerd
5429	pakels	20	0	7783724	307156	160352	S	0,3	1,9	0:03.46	soffice.bin
7263	pakels	25	5	2395512	76552	65368	S	0,3	0,5	0:00.07	Web Content
7346	pakels	20	0	14892	4264	3432	R	0,3	0,0	0:00.03	top

Amb la ordre jobs, podem llistar els treball en según pla.

Iniciem dos procesos en segon pla:

```
pakels@Pakels-MSI:~$ sleep 60 &  
[1] 7494  
pakels@Pakels-MSI:~$ sleep 100 &  
[2] 7495
```

I a continuació els llistem:

```
pakels@Pakels-MSI:~$ jobs  
[1]-  Ejecutando          sleep 60 &  
[2]+  Ejecutando          sleep 100 &
```

La ordre nohup manté la execució de un comando encara que tanquem la terminal.

Per a poder llistar els fitxers del directori dins de 24 segons utilitzarem la següent ordre:

```
>sleep 24; ls home/usuari
```

Cron: Es el administrador de procesos en segon pla que se executen de forma regular.

Crontab: es el arxiu de text on configurem els comandos a executar en un temps determinat.

Per a poder programar la copia del directori del usuari tots els dies a les 23:00h, tindriem que editar el arxiu crontab i incloure la seguent linea:

```
00 23 * * * usuari cp/home/usuari /home/altre_directori
```