

# Tema - Gestió de processos

- ***1 Processos, conceptes***
- ***2 Tipus de processos***
- ***3 Planificació de processos***
- ***4 Programació de l'execució de processos***
- ***5 Monitorització de processos***
- ***6 Senyals***
- ***7 Directori /proc***

## ***Objectius***

L'estudiant serà capaç de :

- Enumerar i descriure els atributs dels processos.
- Descriure les característiques dels diferents tipus de processos.
- Conèixer les maneres d'enviar a executar un procés.
- Enumerar les eines per monitoritzar els processos i conèixer la informació que donen.

# 1 Processos, conceptes

- Estructures de dades d'un procés (kernel):
  - Propietari
  - Estat
  - prioritat d'execució (Nice)
  - informació dels recursos utilitzats
  - procés retorna: 0 si finalitza correctament  
1..255 si finalitza incorrectament
- Recursos dels processos modificables:
  - Temps CPU acumulat
  - Mida arxiu creat
  - Mida segment de dades i segment de pila
  - Mida arxiu core
  - Memòria usada
    - ✓ cshell i tcsh: `limit, limit -h`
    - ✓ bash i ksh: `ulimit, ulimit -ha`

## 1.1 Relació entre processos i comandes

- Una comanda simple crea un sol procés:
  - Ex: `ls`, `cat fitxer`
- Una comanda composta
  - ✓ que conté pipes, genera tants processos com connexions fa la pipe
    - Ex: `who | grep david`
    - `ls -l | sort >fitxer`
  - ✓ que conté varies comandes agrupades, genera tants processos com ; conté
    - Ex: `( gvim sc1 ; chmod +x sc1 )`

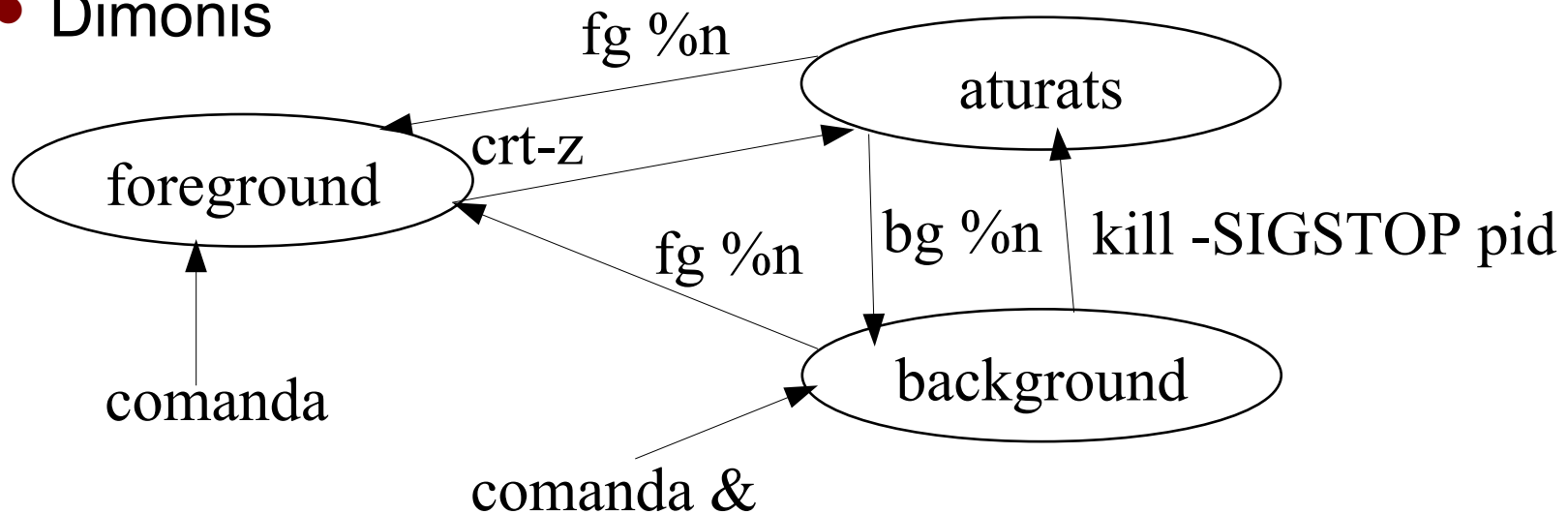
## 1.2 Atributs dels processos

- ✓ PID: nº identificador del procés.
- ✓ PPID: identificador del pare del procés.
- ✓ RUID: identificador real = UID propietari.
- ✓ EUID: identificador efectiu.
  - Normalment EUID= RUID, excepte quan setuid activat.
- ✓ RGID: identificador del grup real = GID propietari
- ✓ EGID: identificador del grup efectiu.
  - Normalment EGID= RGID, excepte quan setgid activat.
- ✓ NICE: prioritat d'execució d'inici.
- ✓ TTY: dispositiu terminal associat al procés

Tot procés té un procés  
pare, excepte el procés init  
que és el procés inicial.

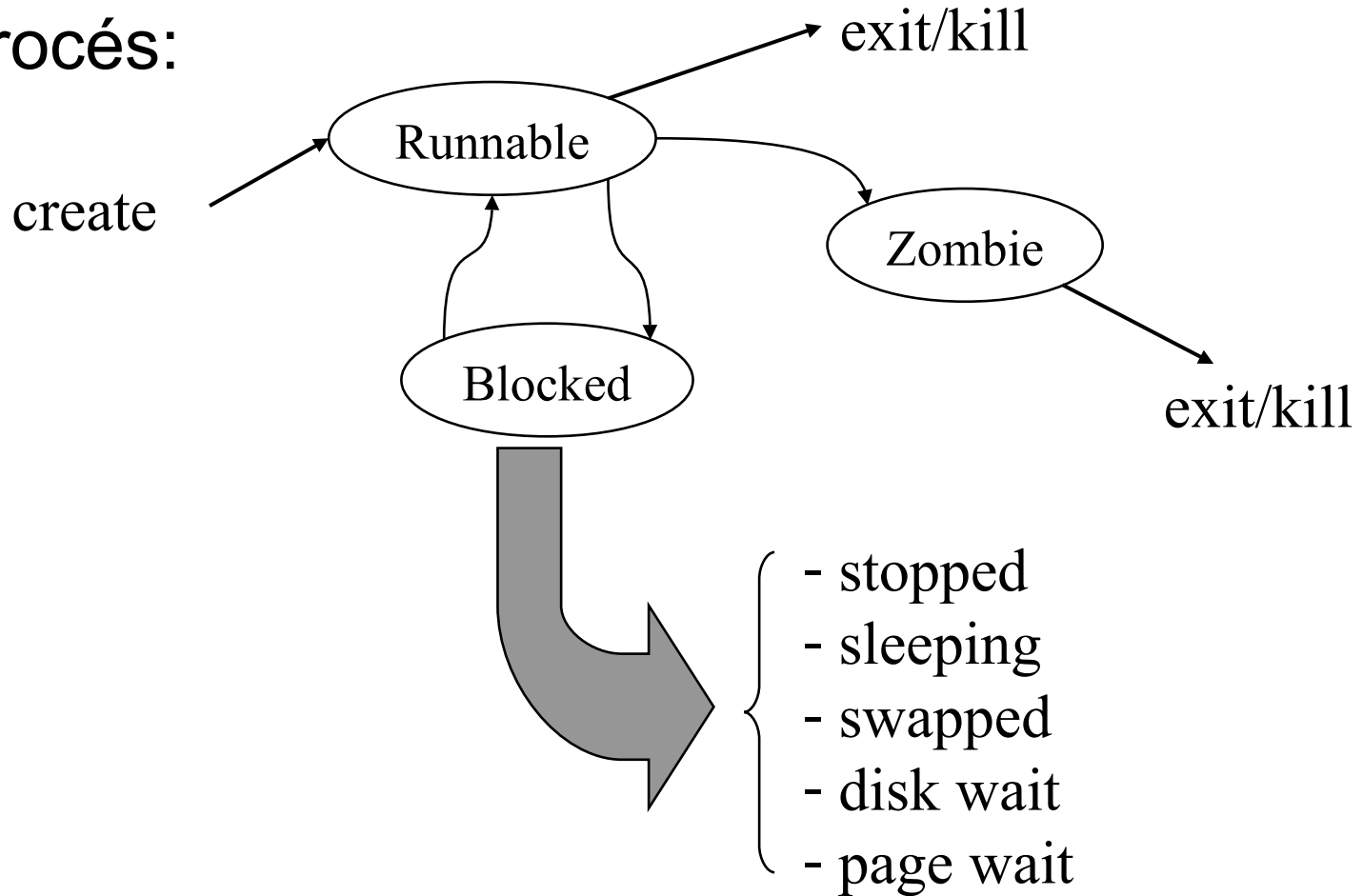
## 2 Tipus de processos

- Processos interactius i no interactius
- S'executen en foreground o background (&)
- Control de jobs (cjt de processos): **jobs**
- Processos batch(processos consumidors de molts recursos, que s'executen d'un en un)
- Dimonis



### 3 Planificació de processos (1/3)

Estats d'un  
procés:



### 3 Planificació de processos (2/3)

- Planificació Round Robin apropiatiu amb prioritats dinàmiques. La prioritat d'execució depèn de:
  - ✓ prioritat base del procés
  - ✓ prioritat d'inici del procés (NICE)
  - ✓ Temps que porta en execució, temps d'espera
- En Solaris i AIX: paràmetres configurables(quantum)
- Prioritat dels processos :
  - + ↑ processos amb prioritat Temps Real
  - dimonis i processos de root
  - processos interactius
  - ↓ processos batch



### 3 Planificació de processos (3/3)

- La prioritat d'entrada (NICE) modificable per:
  - ✓ el propietari del procés
  - ✓ l'administrador
    - `nice` prioritat comanda
    - `renice` prioritat `-p` pid
- Segons la implementació, rangs del nice:
  - ✓ BSD: nice : [-20.. 20]
  - ✓ System V: nice : [0.. 39]  
(número de nice petit -> major prioritat)

## 4 Programació de l'execució de processos

### 4.1 Processos periòdics (1/4)

- Procés servidor (daemon): *cron* / *crond*
- Envia a executar les tasques que estan en el seu fitxer de configuració */etc/crontab*, */var/spool/cron/crontab*
- Segons implementació té diferents modes de funcionament:
  - ✓ cada minut es desperta (versió antiga)
  - ✓ es desperta quan és necessari

## 4.1 Processos periòdics (2/4)

- En BSD:
  - ✓ només un fitxer (ascii) de tasques: `/usr/lib/crontab` o `/etc/crontab`
- En System V:
  - ✓ un fitxer de tasques (no ascii) per usuari
    - `/var/spool/cron/crontabs/nom_usuari`
  - ✓ per instal·lar les tasques:
    - `crontab -l ; crontab -r;`
    - `crontab -u nom_user fitxer`
    - Editar fitxer de tasques `crontab`

1) Introduir tasques en fitxer

2) Instal·lar tasques

*crontab* fitxer

1) `crontab -e`

## 4.1 Processos periòdics (3/4)

- En Linux: implementades les dues variants
  - ✓ `/var/spool/cron/nom_usuari` (System V)
  - ✓ `/etc/crontab` (BSD)
- S'han de definir variables d'entorn:
  - PATH, SHELL, MAILTO, etc.
- Control d'accés a la utilitat crontab:
  - ✓ En BSD: només root té accés.
  - ✓ En System V i Linux: mitjançant un dels fitxers

*/etc/cron.allow*

*/etc/cron.deny*

## 4.1 Processos periòdics (4/4)

- Tasques executades des de la SHELL i amb l'entorn creat pel script
- Fitxer de tasques periòdiques:

✓ En System V:

<i>minut</i>	<i>hora</i>	<i>dia</i>	<i>mes</i>	<i>dia_setmana</i>	<i>script</i>
0..59	0..23	1..31	1..12	0..6 (1..7)	comanda

✓ En BSD:

<i>minut</i>	<i>hora</i>	<i>dia</i>	<i>mes</i>	<i>dia_setmana</i>	<i>usuari</i>	<i>script</i>
0..59	0..23	1..31	1..12	0..6 (1..7)	user	comanda

Ex: 45	10	*	*	1	/etc/script1
0/15	10	*	*	1-5,7	/home/milax/script2
45-60	10	*	*	1,5	/usr/local/script3
0,30	*	13	*	5	/etc/script4

## 4.2 Processos no periòdics

- Maneres d'enviar processos a executar:

- 1) Immediatament

- comanda
- comanda &
- nohup comanda & (sortida estàndart nohup.out)

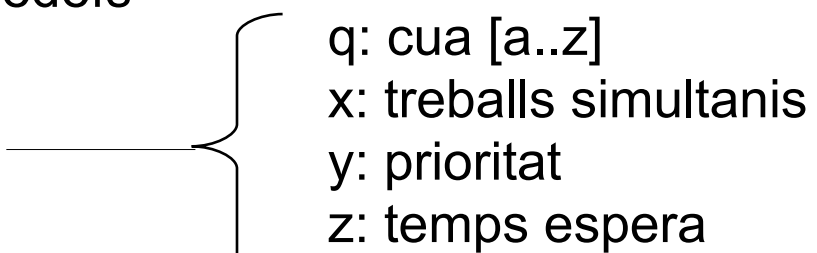
- 2) Sense especificar quan s'executarà

- `batch` comanda

- 3) Especificant el moment concret en que s'executarà

- `at [-f fitxer] data` (dades entrada estàndard)
- El dimoni *atd* gestiona les tasques.
- Es controla l'accés a la utilitat amb:
  - `/etc/at.allow`
  - `/etc/at.deny`

## 4.3 Característiques generals

- Per les utilitats crontab, at i batch:
  - La sortida estàndard i l'error estàndard redireccionada cap a la bústia de correu del propietari de la tasca.
  - Programació de processos implementat amb cues:
    - ✓ cua a: comanda at
    - ✓ cua b: comanda batch
    - ✓ cua c: utilitat crontab
    - ✓ definició de cues addicionals:
      - fitxer /etc/queuedefs
        - `q.xjynzw`
        - `at -q cua`
- 
- q: cua [a..z]  
x: treballs simultanis  
y: prioritat  
z: temps espera

## 5 Monitorització de processos (1/4)

- Eines:

- ✓ ps

- nom usuari corresponent al EUID.
    - identificador del procés (PID).
    - estat del procés.
    - hora inici procés.
    - temps CPU consumit, % ús de la CPU
    - prioritat d'entrada (NICE) i prioritat execució (PRI).
    - tty associada.
    - memòria i memòria virtual usada.
    - . . .



## 5 Monitorització de processos (2/4)

✓ `top`

- Idem que `ps`, però és interactiva i permet:
  - canviar temps actualització
  - matar processos
  - canviar prioritat processos
  - ordenar per diferents camps
- Només s'ha d'usa per debugger o monitorització en un moment concret, consumeix molt temps de processador

✓ `ps tree`

- Per veure la jerarquia dels processos des del procés init
- Per veure jerarquia de processos d'un usuari

## 5 Monitorització de processos (2/4)

✓ `top`

- Idem que `ps`, però és interactiva i permet:
  - canviar temps actualització
  - matar processos
  - canviar prioritat processos
  - ordenar per diferents camps
- Només s'ha d'usa per debugger o monitorització en un moment concret, consumeix molt temps de processador

`pgrep`

✓ `ps tree`

- Per veure la jerarquia dels processos des del procés init
- Per veure jerarquia de processos d'un usuari

## 5 Monitorització de processos (3/4)

- En debian: `/etc/init.d/psacct start`
- En ubuntu: `sudo service acct start`
- `Lastcomm`
  - ✓ nom de comanda
  - ✓ flags d'execució
    - S -- comanda generada pel super-user
    - F -- comanda generada per un fork, sense un exec
    - C -- comanda executada usant compatibilitat PDP-11
    - D -- comanda acabada generant un fitxer core
    - X -- comanda acabada amb un signal SIGTERM
  - ✓ nom de l'usuari efectiu
  - ✓ temps d'inici del procés

*normalment usa /var/log/account*

## 5 Monitorització de processos (4/4)

### **vmstat**

Procs:r,b (número)

Memory:swpd, free, buff, cache (quantitat)

Opció -a: inact, active

Swap:si, so (quantitat memoria/s)

IO:bi, bo (blocks/s).

System:in, cs (quantitat/s)

CPU:us, sy, id, wa, st (percent total temps)

## 6 Senyals (1/2)

- Interrupcions software per informar d'un succés
- S'envien entre processos.
- Quan un procés rep una senyal pot:
  - ✓ cridar a una rutina per defecte
  - ✓ cridar a una rutina pròpia
  - ✓ Ignorar-la
  - ✓ bloquejar-la / desbloquejar-la
- Processos que no moren al rebre-la en queden en estat:
  - ✓ Zombie
  - ✓ bloquejats en espera de E/S o NFS

## 6 Senyals (2/2)

- Cada senyal té un nom i un número, diferents segons implementació (BSD:30, System V:19)
  - ✓ `kill -nº_senyal pid`
  - ✓ `killall -nº_senyal comanda`
  - ✓ `kill -l -> llista tots els senyals`
  - ✓ `Pkill -> sudo pkill -u root sshd`
- Tipus de senyals:
  - ✓ SIGKILL: termina el procés.
  - ✓ SIGSTOP: atura el procés 2 CTRL Z.
  - ✓ SIGCONT: reempren execució procés.
  - ✓ SIGHUP: desconnecta la terminal dels processos, avisa a dimoni relectura fitxer configuració
  - ✓ SIGINT: acaba el procés foreground 2 CTRL C
  - ✓ SIGQUIT: idem 2 CTRL \, genera core
  - ✓ SIGPWR: reducció alimentació equip

## 7 Directori /proc (1/3)

- Conté una jerarquia d'arxius especials(virtuals) que representen l'estat actual del nucli, permetent a les aplicacions i usuaris veure i modificar el seu estat.
- Informació sobre el maquinari del sistema
- Informació dels processos que s'estan executant
- Alguns fitxers poden ser manipulats per a comunicar al nucli canvis en la configuració
- S'agrupen en directoris virtuals i subdirectoris segons la seva funcionalitat
- Per extreure/posar informació dels arxius virtuals i mostrar la de manera útil hi ha:
  - ✓ cat, lspci, apm, free, top, ...
  - ✓ Redirecció de la sortida: >

## 7 Processos del directori /proc (2/3)

- **/proc/PID** : <http://manpages.ubuntu.com/manpages/hardy/es/man5/proc.5.html>
  - ✓ Hi ha un subdirectori numèric per a cada procés en execució
  - ✓ El nom del subdirectori és el ID del procés.
  - ✓ Cada un de ells conté entre altres:
    - cmdline: comanda completa, res si zombie o swap out.
    - cwd : enllaç al directori de treball
    - environ : entorn de treball
    - exe: enllaça simbòlic al fitxer executable
    - fd: una entrada per cada fitxer obert
    - maps: una entrada per a cada zona de memòria que té assignada
    - stat: l'estat del procés (usat per la comanda ps)
    - statm: informació de les pàgines de memòria que usa el procés



## 7 Fitxers del directori /proc (3/3)

- **Exemples d'altres fitxers virtuals**

- ✓ /proc/scsi

- conté informació dels dispositius scsi

- ✓ /proc/cpuinfo

- conté informació del processador

- ✓ /proc/sys

- Fitxers virtuals de configuració del nucli

- ✓ /proc/uptime

- conté informació del temps que porta encès el sistema des del últim reinici. Les dades de /proc/uptime es mínima:

- 350735.47 234388.90

- El primer número indica el número total de segons que el sistema ha estat en funcionament. El segon indica quan d'aquest temps, la màquina ha estat inactiva.

## Ex: /etc/crontab (Linux)

```
SHELL=/bin/bash
```

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

```
MAILTO=root
```

```
HOME=/
```

```
# run-parts
```

```
01 * * * * root run-parts /etc/cron.hourly
```

```
02 4 * * * root run-parts /etc/cron.daily
```

```
22 4 * * 0 root run-parts /etc/cron.weekly
```

```
42 4 1 * * root run-parts /etc/cron.monthly
```

## \$ps aux (a un Linux)

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	1372	204	?	S	Aug16	0:44	init [3]
root	2	0.0	0.0	0	0	?	SW	Aug16	0:00	[keventd]
root	5	0.0	0.0	0	0	?	SW	Aug16	6:11	[kswapd]
root	6	0.0	0.0	0	0	?	SW	Aug16	0:01	[bdf flush]
root	7	0.0	0.0	0	0	?	SW	Aug16	1:49	[kupdated]
root	586	0.0	0.0	1696	220	?	S	Aug16	4:38	syslogd -m 0
root	591	0.0	0.0	1364	176	?	S	Aug16	0:00	klogd -x
rpc	609	0.0	0.2	1700	564	?	S	Aug16	0:04	portmap
rpcuser	637	0.0	0.0	1600	252	?	S	Aug16	0:00	rpc.statd
root	712	0.0	0.2	1892	724	?	S	Aug16	0:00	rpc.mountd
root	717	0.0	0.0	0	0	?	SW	Aug16	1:18	[nfsd]
root	726	0.0	0.0	0	0	?	SW	Aug16	0:00	[lockd]
root	727	0.0	0.0	0	0	?	SW	Aug16	0:00	[rpciod]
daemon	822	0.0	0.0	1404	176	?	S	Aug16	0:00	/usr/sbin/atd
root	840	0.0	0.3	10144	776	?	S	Aug16	18:04	xinetd -stayalive
root	976	0.0	0.0	1400	160	?	S	Aug16	0:04	gpm -t ps/2 -m /d
root	1149	0.0	0.0	1576	232	?	S	Aug16	0:08	crond
root	1349	0.0	0.2	5172	628	?	S	Aug16	3:49	nmbd -D
root	1377	0.0	0.1	2620	416	?	S	Aug16	0:58	/usr/sbin/sshd
root	1385	0.0	0.0	1344	136	tty3	S	Aug16	0:00	/sbin/mingetty tt
root	1386	0.0	0.0	1344	136	tty4	S	Aug16	0:00	/sbin/mingetty tt
lp	23818	0.0	0.2	11296	588	?	S	Aug23	1:18	lpd Waiting
caliagas	13602	0.0	0.4	2792	1156	pts/1	S	Sep30	0:00	-csh
caliagas	13637	0.0	7.4	46864	18948	pts/1	SN	Sep30	1:35	/usr/lib/netscape
jbanus	29283	0.0	0.5	2424	1332	pts/2	S	15:10	0:00	-zsh
caliagas	30636	0.0	0.8	4432	2152	pts/0	S	15:52	0:00	mutt
salvarez	30831	0.0	0.4	2284	1248	pts/4	S	15:58	0:00	-bash
salvarez	31318	0.0	0.2	2584	668	pts/4	R	16:12	0:00	ps aux

\$ps -ef (a solaris)

	UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	0	0	0		Jun 15	?	0:01	sched
root	1	0	0		Jun 15	?	0:14	/etc/init -
root	2	0	0		Jun 15	?	0:00	pageout
root	3	0	1		Jun 15	?	1018:57	fsflush
root	237	1	0		Jun 15	?	0:00	/usr/sbin/rpcbind
root	215	1	0		Jun 15	?	0:00	/usr/lib/nfs/nfsd -a 16
root	266	1	0		Jun 15	?	0:00	/usr/sbin/inetd -s
root	315	1	0		Jun 15	?	0:11	/usr/sbin/cron
daemon	278	1	0		Jun 15	?	0:00	/usr/lib/nfs/statd
root	313	1	0		Jun 15	?	0:09	/usr/sbin/syslogd
root	423	417	0		Jun 15	?	0:00	/usr/lib/saf/listen tcp
salvarez	27841	27839	0	16:15:56	pts/5		0:00	-bashnull
root	410	1	0		Jun 15	?	0:15	/usr/local/sbin/sshd
root	360	1	0		Jun 15	?	0:01	/usr/lib/utmpd
dbc.ge	28372	28349	0	Sep 06	pts/7		0:00	ftp 80.32.46.96
root	27839	410	0	16:15:56	?		0:01	/usr/local/sbin/sshd
oracle	16671	1	1	Jul 03	?		0:34	ora_s000_doc
oracle	16661	1	0	Jul 03	?		0:25	ora_dbw0_doc
oracle	19210	1	0	Aug 06	?		0:00	./tnslsnr
oracle	16673	1	0	Jul 03	?		0:01	ora_d000_doc
dbc.ge	28349	28346	0	Sep 06	pts/7		0:00	/usr/local/bin/bashnull
oracle	16667	1	0	Jul 03	?		5:46	ora_smon_doc
oracle	16659	1	0	Jul 03	?		0:02	ora_pmon_doc
dbc.ge	28346	1	0	Sep 06	??		0:00	/usr/openwin/bin/shelltool
mvp.is	10362	10361	0	Aug 01	pts/2		0:00	ssh alumne.etse.urv.es
nobody	8434	8433	0				0:00	<defunct>
root	27849	27841	0	16:16:52	pts/5		0:00	ps -ef
mvp.is	9622	1	0				0:00	<defunct>
cvg.si	27824	27822	0	16:11:52	pts/4		0:00	-bashnull

## \$pstree (a un Linux)

```
init--and
  |-atd
  |-bdf flush
  |-crond
  |-gpm
  |-identd---identd---3*[identd]
  |-keventd
  |-khubd
  |-klogd
  |-kswapd
  |-kupdated
  |-lockd---rpciod
  |-lpd
  |-6*[mingetty]
  |-8*[nfsd]
  |-nmbd
  |-portmap
  |-3*[raidld]
  |-rhnsd---rhn_check
  |-rpc.mountd
  |-rpc.statd
  |-sendmail
  |-sshd--sshd---csh
  |   |-sshd---csh---netscape-commun---netscape-commun
  |   |-sshd---bash---tcsh---runalg---runtime---runttest---gaborClass
  |   |-sshd---bash---xfig
  |   |-sshd---bash---pstree
  |   `--sshd---sftp-server
  |-syslogd
  |-xfs
  `--xinetd---3*[smbd]
```

## Exercicis

- ✓ El script `/sbin/estadistica` l'ha d'executar el sistema el primer dia del mes i tots els diumenges.
- ✓ El joan vol que s'executi el programa `innocent` el dia 28 de desembre. A més cada divendres s'ha d'executar el script `/home/joan/copia`.
- ✓ Obtenir el noms dels processos adormits que es van enviar a executar el dia que ens indiquen per paràmetre.

## ***Bibliografia***

Cap 8 “Essential System Administration”

Cap 4 i 9 “Unix System Administration Handbook

Cap 2 i 13 “Administración Unix. System V. Redes TCP/IP”