

Universitat de Lleida
Escola Politècnica Superior

Administració i Manteniment de Sistemes i Aplicacions
Pràctica 8: Prioritat-NoHup

Sergi Puigpinós Palau
Jordi Rafael Lazo Florensa

21 de desembre de 2020

1 Pregunta 1

1. Feu un programa en C (`prioritza.c`) que accepti 1 argument (el pid d'un procés; p.e. del programa escriure de les transparències). El programa `prioritza.c` (el qual s'haurà d'executar com a root i en background), ha de fer el següent:
 - (a) Assignar-se a ell mateix una prioritat d'execució de 20 i una política de planificació `SCHED_RR`.
 - (b) Assignar al seu pare (el shell) una prioritat d'execució de 30 i una política de planificació `SCHED_RR`.
 - (c) Assignar al procés que s'haurà passat com argument una prioritat d'execució de 10 i una política de planificació `SCHED_RR`.

El codi del programa en C es el següent:

Programa en C: `prioritza.c`

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sched.h>
4 #include <unistd.h>
5
6 int main(int argc, char** argv){
7
8     if (argc != 2){
9         printf("Sisplau, escriu un argument\n");
10        exit(-1);
11    }
12
13    int pid = atoi(argv[1]);
14    struct sched_param self, parent, parameter;
15
16    self.sched_priority = 20;
17    parameter.sched_priority = 10;
18    parent.sched_priority = 30;
19
20    sched_setscheduler(getpid(), SCHED_RR, &self);
21    sched_setscheduler(pid, SCHED_RR, &parameter);
22    sched_setscheduler(getppid(), SCHED_RR, &parent);
23 }
```

En lliurar la pràctica haureu de dir com heu comprovat que funciona correctament.

1. S'obra una nova terminal i s'executa de fons el script `escriure.sh` amb la comanda:

```
# gnome-terminal -- /bin/bash -c 'bash escriure.sh'
```

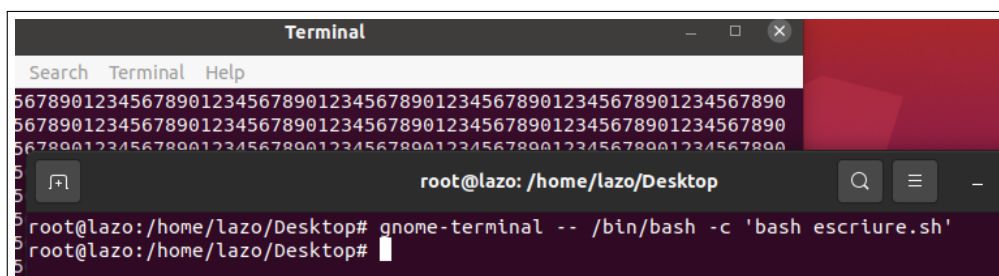
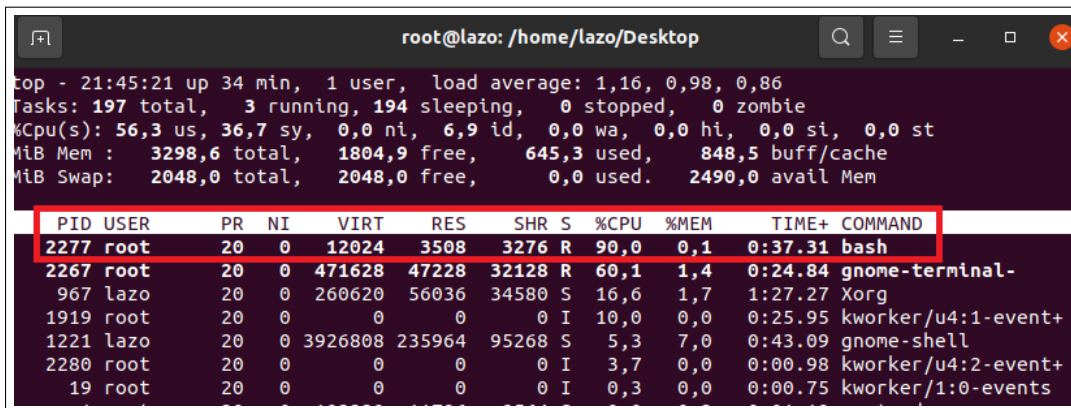


Figura 1: Execució del script `escriure.sh`.

2. S'obté el PID de tots els processos en execució utilitzant la comanda:

```
# top
```



```

top - 21:45:21 up 34 min, 1 user, load average: 1,16, 0,98, 0,86
Tasks: 197 total, 3 running, 194 sleeping, 0 stopped, 0 zombie
%Cpu(s): 56,3 us, 36,7 sy, 0,0 ni, 6,9 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 3298,6 total, 1804,9 free, 645,3 used, 848,5 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 2490,0 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 2277 root        20   0 12024  3508  3276 R  90,0   0,1   0:37.31  bash
 2267 root        20   0 471628 47228 32128 R  60,1   1,4   0:24.84  gnome-terminal-
  967 lazo        20   0 260620  56036 34580 S  16,6   1,7   1:27.27  Xorg
 1919 root        20   0      0      0      0 I  10,0   0,0   0:25.95  kworker/u4:1-event+
 1221 lazo        20   0 3926808 235964 95268 S   5,3   7,0   0:43.09  gnome-shell
 2280 root        20   0      0      0      0 I   3,7   0,0   0:00.98  kworker/u4:2-event+
    19 root        20   0      0      0      0 I   0,3   0,0   0:00.75  kworker/1:0-events
  
```

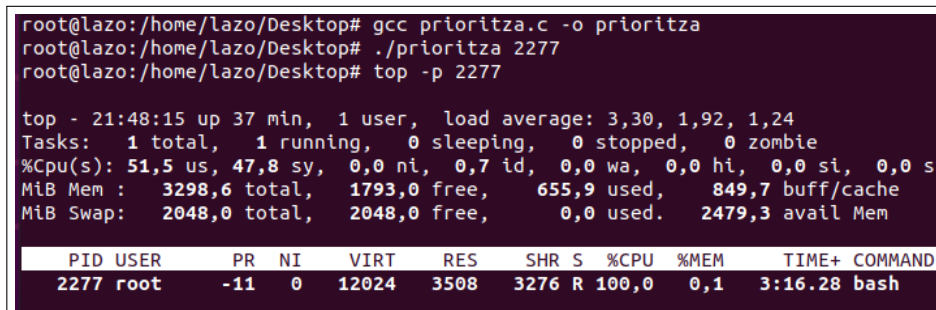
Figura 2: Resultat de la comanda *top*.

3. Finalment s'executen les comandes següents per comprovar que el programa ha disminuït la prioritat del PID correctament:

```
# gcc prioritza.c -o prioritza
```

```
# ./prioritza
```

```
# top -p <pid>
```



```

root@lazo:/home/lazo/Desktop# gcc prioritza.c -o prioritza
root@lazo:/home/lazo/Desktop# ./prioritza 2277
root@lazo:/home/lazo/Desktop# top -p 2277

top - 21:48:15 up 37 min, 1 user, load average: 3,30, 1,92, 1,24
Tasks: 1 total, 1 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 51,5 us, 47,8 sy, 0,0 ni, 0,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 3298,6 total, 1793,0 free, 655,9 used, 849,7 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 2479,3 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
 2277 root       -11   0 12024  3508  3276 R 100,0   0,1   3:16.28  bash
  
```

Figura 3: Resultat de la prioritat del PID després d'executar el programa *prioritzar.c*.

2 Pregunta 2

1. Proveu l'script Nohup dels apunts. Hi ha alguna diferència amb la comanda `nohup`?

La comanda *nohup* anteposada a una comanda en la terminal evita que es cancel·li automàticament quan es desconnecta o surt del shell.

En la Figura 4 s'executa la comanda *nohup* i aquesta crea un arxiu per defecte *nohup.out* on s'emmagatzemaran les comandes que s'executin amb *nohup*. Aquest arxiu *.out* només es pot accedir amb *root*.

En canvi utilitzant el *script* *nohup* aquest crea un arxiu *nohup_script.out* i redirigeix les comandes *nohup* al nou arxiu creat.

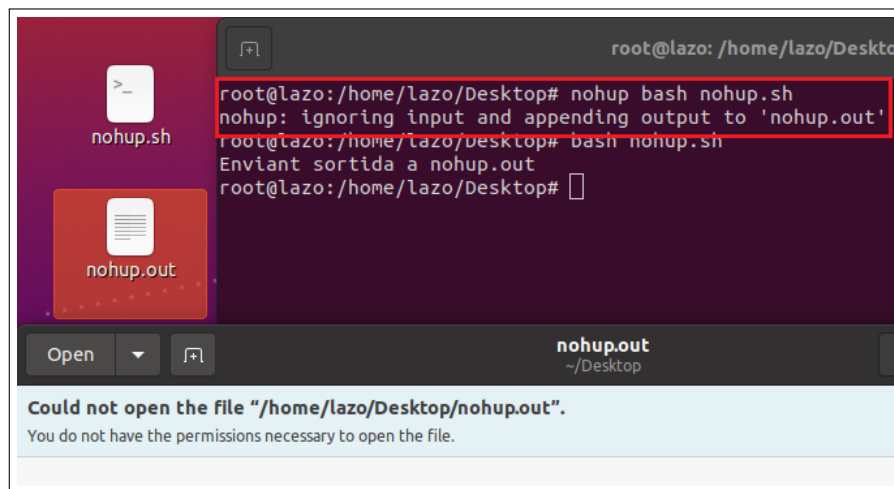


Figura 4: Resultat de l'execució de la comanda *nohup*.

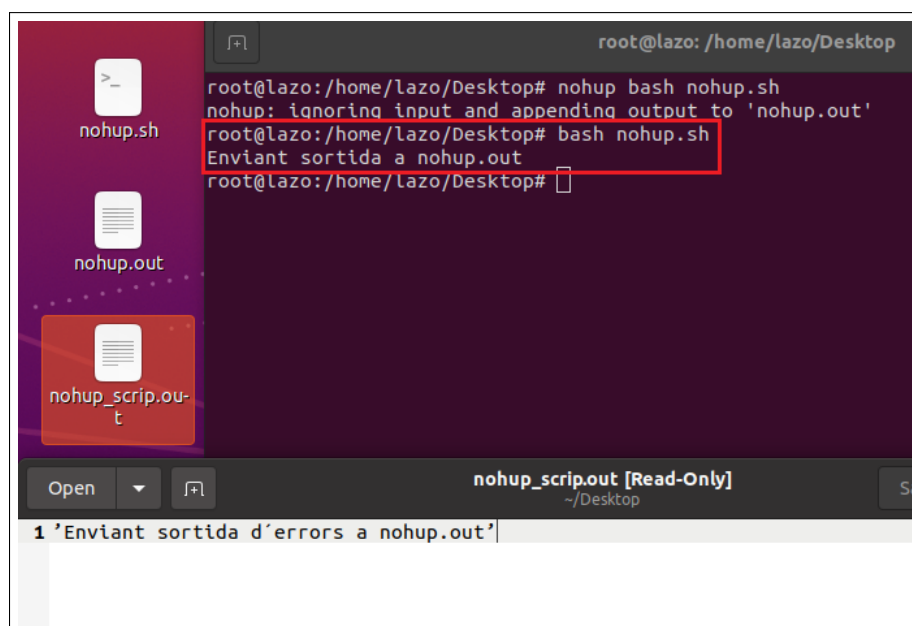


Figura 5: Resultat del script *nohup*.

3 Pregunta 3

1. MySQL. Realitzeu les tasques següents:

- Creeu una BBDD (Base de Dades) en mysql amb 3 taules. Empleneu les taules amb 10 registres cadascuna.
- Creeu un RAMDISC i monteu-hi la BBDD creada en l'apartat a.

3.1 (a)

1. Instalar *mysql* en linux:

```
# sudo apt install mysql-server
```

2. S'inicialitza amb:

```
# mysql
```

3. Es crea la BBDD:

```
-> CREATE DATABASE <nom_BBDD>;
```

4. Es comprova que s'ha creat la BBDD amb la comanda:

```
-> SHOW DATABASES;
```

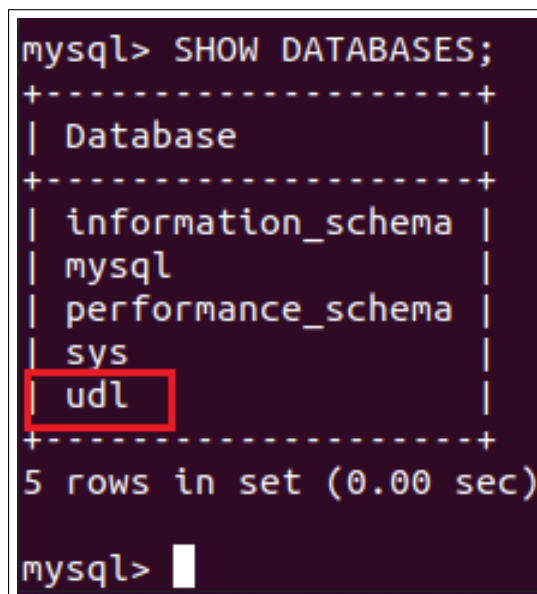


Figura 6: Creació de la base de dades *udl*.

5. S'accedeix a la base de dades *udl* amb la comanda:

```
-> USE <nom_BBDD>;
```

6. Es crea les 3 taules amb la comanda següent:

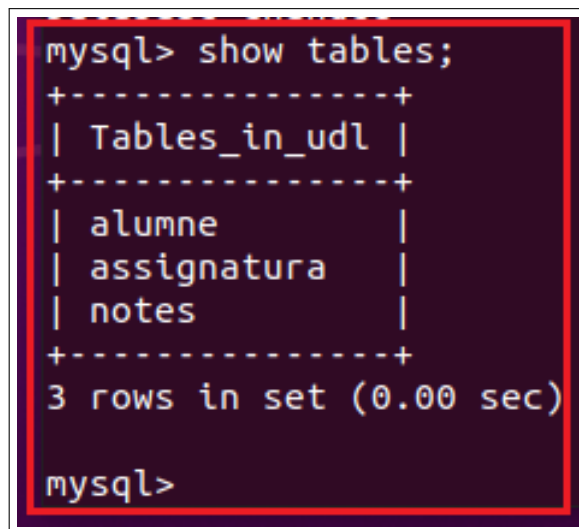
```
mysql -> CREATE TABLE alumne (  
-> idalumne SERIAL PRIMARY KEY  
-> nom varchar(20),  
-> edat int,  
-> ciutat varchar(15) );
```

```
mysql -> CREATE TABLE assignatura (  
-> idassignatura SERIAL,  
-> nom varchar(20),  
-> numalumnes int,  
-> PRIMARY KEY (idassignatura) );
```

```
mysql -> CREATE TABLE notes (  
-> idalumne int,  
-> idassignatura int REFERENCES assignatura (idassignatura),  
-> nota int,  
-> PRIMARY KEY (idalumne,idassignatura) );
```

7. Es visualitza que les 3 taules han estat creades correctament amb la comanda:

```
-> SHOW tables;
```



```
mysql> show tables;  
+-----+  
| Tables_in_udl |  
+-----+  
| alumne        |  
| assignatura    |  
| notes         |  
+-----+  
3 rows in set (0.00 sec)  
  
mysql>
```

Figura 7: Creació de les 3 taules en la base de dades *udl*.

8. Es comprova que els camps de les taules creades estan correctes amb la comanda:

```
-> DESCRIBE <nom_taula>;
```

```
mysql> DESCRIBE notes;
+-----+-----+-----+-----+-----+-----+
| Field      | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idalumne   | int  | NO   | PRI | NULL    |       |
| idassignatura | int  | NO   | PRI | NULL    |       |
| nota       | int  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DESCRIBE alumne;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idalumne   | bigint unsigned | NO   | PRI | NULL    | auto_increment |
| nom        | varchar(20)    | YES  |     | NULL    |               |
| edat       | int            | YES  |     | NULL    |               |
| ciutat     | varchar(15)    | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> DESCRIBE assignatura;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idassignatura | bigint unsigned | NO   | PRI | NULL    | auto_increment |
| nom          | varchar(20)    | YES  |     | NULL    |               |
| numalumnes   | int            | YES  |     | NULL    |               |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Figura 8: Camps creats en les 3 taules de la base de dades de *udl*.

9. S'executen les següents comandes (10 vegades cadascuna) pero omplir les taules crades:

```
-> INSERT INTO alumne (nom,edat,ciutat) VALUES ('Joan',19,'Lleida');
```

```
-> INSERT INTO assignatura (nom,numalumnes) VALUES ('IBD',60);
```

```
-> INSERT INTO notes VALUES (2,1,8);
```

10. Per últim es comprova que els valors afegits s'han guardat en les taules amb la comanda:

```
-> SELECT * FROM <nom_taula>;
```

```
mysql> SELECT * FROM notes
-> ;
+-----+-----+-----+
| idalumne | idassignatura | nota |
+-----+-----+-----+
| 2 | 1 | 8 |
| 2 | 2 | 3 |
| 3 | 1 | 4 |
| 3 | 2 | 3 |
| 3 | 6 | 4 |
| 4 | 1 | 10 |
| 4 | 2 | 8 |
| 4 | 3 | 3 |
| 4 | 5 | 10 |
| 4 | 6 | 9 |
+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> SELECT * FROM alumne;
+-----+-----+-----+-----+
| idalumne | nom | edat | ciutat |
+-----+-----+-----+-----+
| 1 | Joan | 19 | Lleida |
| 2 | Teresa | 20 | Barcelona |
| 3 | Marta | 18 | Lleida |
| 4 | Juli | 22 | Manresa |
| 5 | Montse | 17 | Barcelona |
| 6 | Sergi | 19 | Lleida |
| 7 | Ian | 18 | Manresa |
| 8 | Pau | 20 | Barcelona |
| 9 | Oriol | 18 | Lleida |
| 10 | Quimetix | 22 | Manresa |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)

mysql> SELECT * FROM assignatura;
+-----+-----+-----+
| idassignatura | nom | numalumnes |
+-----+-----+-----+
| 1 | IBD | 60 |
| 2 | EDALG | 50 |
| 3 | ESOFI | 30 |
| 4 | EDI | 75 |
| 5 | LABOSOFT | 30 |
| 6 | TC | 40 |
| 7 | IBBB | 60 |
| 8 | EDBBALG | 50 |
| 9 | BESOFI | 30 |
| 10 | BBEDI | 75 |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

Figura 9: Dades afegides a les 3 taules creades.

3.2 (b)

1. Es crea el script bash en el *init* amb la comanda:

```
# gedit /etc/init.d/ramdisc.sh
```



```
1  #!/bin/bash
2
3  ROOTUSER_NAME=root
4  MOUNTPT=/var/lib/mysql
5  SIZE=102400
6  BLOCKSIZE=1024
7  DEVICE=/dev/ram0
8
9      service mysql stop
10     cp -r /var/lib/mysql/* /home/lazo/Desktop/sql/
11     rm -Rf /var/lib/mysql/
12
13     username='id -nu '
14     [ "$username" != "$ROOTUSER_NAME" ] && echo "no autoritzat" && exit
15     1
16     [ ! -d "$MOUNTPT" ] && mkdir $MOUNTPT
17     dd if=/dev/zero of=$DEVICE count=$SIZE bs=$BLOCKSIZE
18     mkfs -t ext4 $DEVICE
19     mount $DEVICE $MOUNTPT
20     chmod 777 $MOUNTPT
21     echo $MOUNTPT " disponible"
22
23     cp -r /home/lazo/Desktop/sql/* /var/lib/mysql/
24     chown -R mysql:mysql /var/lib/mysql/
25     service mysql start
26 exit 0
```

2. Es crea el directori on es muntarà el ramdisc:

```
# mkdir -p /tmp/ramdisk
```

3. Es crea el sistema de fitxers:

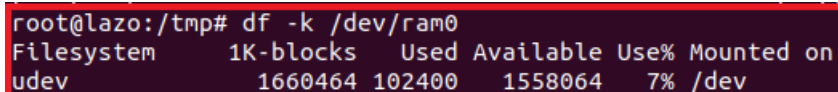
```
# mkfs -t ext2 /dev/ram0
```

4. Es munta el ramdisc en el directori creat anteriorment:

```
# mount /dev/ram0 /tmp/ramdisk
```

5. Finalment es comprova que s'ha creat correctament:

```
# df -k /dev/ram0
```



Filesystem	1K-blocks	Used	Available	Use%	Mounted on
udev	1660464	102400	1558064	7%	/dev

Figura 10: Dades afegides a les 3 taules creades.