

# Pràctica 3: Descobrint el /proc

Sistemes operatius 2023-24

Termini d'entrega del qüestionari pel moodle:

16 d'abril → grups 1, 3, i 4

17 d'abril → grup 2, 5, 6, 7 i 8

# Introducció

- En aquesta pràctica es tracta d'executar comandes del shell però des d'un programa fet en "C" i realitzant les adequades **crides al sistema** per tal d'executar-les tal i com es desitja.
- S'aprofita aquesta pràctica per "aprendre" nous conceptes pràctics de Linux. En aquesta ocasió s'estudiarà el que hi ha a la carpeta del sistema **/proc** . Aquesta carpeta és un sistema de fitxers "virtual" (no existeixen, no són al disc) on s'hi pot trobar informació detallada del funcionament del sistema i de tots els processos que s'estan executant. Fent un **ls -l /proc** es pot veure:
  - Carpetes i fitxers amb "nom" → mostren informació del sistema
  - Carpetes amb un número → contenen informació del procés amb el PID corresponent (només es poden visualitzar els propis processos per un tema de permisos)
- Busca més informació fent **man proc** o **man 5 proc** o bé visita llocs web, per exemple <http://www.linuxfocus.org/Castellano/January2004/article324.shtml>

# Estudi previ

- Mira quin és el PID del teu procés **bash**. Pots fer per exemple:

```
ps -ef | grep u2000779
```

```
u2000779 15735 15723 0 10:13 pts/0 00:00:00 -bash
u2000779 16053 15735 0 10:26 pts/0 00:00:00 ps -ef
u2000779 16054 15735 0 10:26 pts/0 00:00:00 grep u2000779
```

- (NOTA → Cal substituir u2000779 pel teu número d'usuari de bas)
- i llavors es tracta de mirar la carpeta **/proc/15735/** i veure quins fitxers hi ha. Entre d'altres per exemple hi ha el fitxer **status** que conté informació de l'estat del procés. Per exemple amb la següent comanda es pot veure el consum de memòria del procés:

```
cat /proc/15735/status | grep Vm | sort -nrk2 >> sortida.txt
```

- Aquesta és la comanda que cal implementar en C i crides (amb el número de procés corresponent).

# Es demana

- Implementar un programa en C que utilitzant crides al sistema faci exactament el mateix que les següents comandes bash (amb el número de procés corresponent):

```
cat /proc/15735/status | grep Vm | sort -nrk2 >> sortida.txt
```

- Aquesta pràctica cal realitzar-la utilitzant les crides al sistema *fork* (per crear processos fill que executin cada part), *exec/* (per executar les comandes des del programa), *pipe* (per comunicar els canals del diferents processos), *close* (per tancar canals), *dup* (per duplicar canals), *wait* (per sincronitzar l'acabament), *open* (per crear/obrir un fitxer), *lseek* (per desplaçar-se dins d'un fitxer), etc. Les que necessiteu.
- OPCIONAL: Passar com a paràmetre del programa el número del procés (pas de paràmetres en C)

# Exemples de crides

- Per exemple: Aquesta crida executa la comanda **cat** amb aquestes opcions.

```
execl ("/bin/cat", "cat", "/proc/15735/status", (char *)0 );
```

- Per exemple: Aquesta crida crea un fitxer "out.txt" al directori actual per només escriptura i si ja existia el sobreescriu. El fitxer tindrà els permisos "600".

```
fd_fitxer = open("./out.txt", O_WRONLY | O_CREAT | O_TRUNC, 00600);
```

# Aprendre C és “fàcil”

<http://www.cprogramming.com/>

[https://en.wikibooks.org/wiki/C\\_Programming](https://en.wikibooks.org/wiki/C_Programming)

<http://www.tutorialspoint.com/cprogramming/>

... + 100.000 tutorials i llibres

Història, descripció i estàndards del C:

[http://en.wikipedia.org/wiki/C\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/C_%28programming_language%29)

Descripció de les llibreries estàndard de C:

<http://www.mers.byu.edu/docs/standardC/index.html>

**Ull, que fem C, NO C++**