

# Sistemes Operatius 1

## Sessió de problemes 8 – Mayo del 2022

### Objectius

L'objectiu d'aquesta sessió és analitzar dues famílies de funcions que s'han vist a classe de teoria, en particular, `read/write` i `fread/write`. Totes dues famílies serveixen per llegir i escriure dades de disc. Les primeres són crides a sistema i les segones són crides a una llibreria d'usuari que internament fan les crides a sistema a `read/write`.

En aquesta sessió l'objectiu és analitzar la "eficiència" de cadascuna de les funcions. En particular, la signatura de la funció `read` és

```
read(fd, vector, nbytes);
```

on `fd` és el descriptor de fitxer, `vector` és lloc de memòria on es volen emmagatzemar les dades llegides i `nbytes` és el nombre de bytes a llegir, que aquí anomenarem "la mida de bloc". El descriptor de fitxer s'obté mitjançant la crida a sistema `open`.

La signatura de la funció `fread` és

```
fread(vector, mida_element, nombre_elements, fp);
```

on `vector` és el lloc de memòria es volen emmagatzemar les dades llegides, `mida_element` és la mida de cadascun dels elements a llegir, `nombre_elements` és el nombre d'elements a llegir i `fp` és el fitxer obert amb la crida a `fopen`. Aquesta funció llegeix un total de `mida_element x nombre_elements` bytes de disc.

Ens podem realitzar diverses preguntes a l'hora d'utilitzar aquestes funcions: per exemple, si faig servir les crides a sistema, quina és la eficiència associada si es llegeix un fitxer de byte en byte o en blocs de mida de 4096 bytes? Triga el mateix? De forma equivalent, si faig servir la funció `fread` és igual si llegeixo un fitxer de byte en byte que en blocs de 4096 bytes?

### Eines que es proporcionen

Es demana, primer de tot, executar el codi `create_big_file.c`. Aquest programa generarà, al vostre ordinador, un fitxer de 4GBytes anomenat `big_file.bin`. Sí, anem a crear un fitxer realment gran per poder mesurar l'eficiència dels dos tipus de crida que disposem per llegir o escriure a un fitxer.

Es proporcionen a més dos programes en C, un que utilitza crides a sistema i l'altre les d'usuari, per llegir/escriure un fitxer. Els dos programes utilitzen un paràmetre d'execució per especificar el nombre de bytes, la mida de bloc, a llegir a cada iteració fins que s'ha llegit tot el fitxer. L'objectiu és doncs veure quina és la influència del nombre de bytes que es llegeix a cada iteració en el temps d'execució de l'aplicació.

Observeu els dos codis. Els dos codis llegeixen de l'entrada estàndard i escriuen a la sortida estàndard fins que s'ha llegit tot el fitxer.

Per executar cadascun dels experiments feu servir la següent instrucció

```
time ./io_system_call 1024 < big_file.bin > /dev/null
```

En aquest cas `io_system_call` és el codi que utilitza crides a sistema per accedir a disc. El valor 1024 és la mida de bloc amb la qual s'executarà l'aplicació (podeu posar qualsevol sencer com, per exemple, 64, 512, 1024, ...).

Observeu que es fa servir redirecció per indicar que l'entrada estàndard és el fitxer `big_file.bin`. La sortida estàndard és `/dev/null`, una mena de "forat negre". Tot el que s'hi escriu es perd. Escrivint tot en aquest "forat negre" no es crea cap fitxer a disc cosa que pot esbiaixar els resultats dels experiments (ja que no podem assegurar que a cada experiment el fitxer es creï a les mateixes posicions de disc).

La instrucció `time` mesura el temps d'execució de l'aplicació. Per exemple, ens pot donar un resultat com

```
real 0m34.055s
user 0m12.902s
sys 0m21.145s
```

En aquest cas ens diu que el temps d'execució en mode usuari és de 12.9 segons i que el temps d'execució en el sistema operatiu (en mode nucli) és de 21.14 segons.

El valor "real" ens indica el temps cronològic que ha trigat l'aplicació en executar-se: des que s'ha començat a executar fins que ha finalitzat d'executar-se, incloent el fet que hi ha hagut canvis de context a altres processos pel mig, el temps que el procés ha estat bloquejat, etc.

Per fer les proves es proposa que feu servir mides de bloc de 64 bytes, 128 bytes, 512 bytes, 1024 bytes, 4096 bytes, 16384 bytes, 65536 bytes i 262144 bytes.

És convenient que la mida del fitxer sigui més gran que la que RAM que tingueu associada a la vostra màquina virtual (o la màquina en què executeu aquestes proves en cas que les executeu directament a la màquina nativa). Això és perquè, en llegir per primer cop el fitxer, les dades associades s'emmagatzemen al buffer intern del sistema operatiu, típicament anomenada "caché de disc". El sistema operatiu aprofita tota la RAM disponible que pot com a buffer intern. Si la mida del fitxer és inferior a la RAM disponible, les següents vegades que es llegeixi del fitxer ja no caldrà llegir-lo de disc sinó que s'accedirà directament a la còpia que hi ha memòria. En canvi, si la mida del fitxer és més gran que la RAM disponible, el sistema operatiu haurà d'accedir a disc per llegir les dades.

## Exercici a entregar

Es demana executar els experiments amb diferents mides de bloc per veure quina és la influència de la mida del bloc en la lectura del disc. Feu-ho pel codi que utilitza crides a sistema com el que fa crides a la llibreria d'usuari. Feu una taula perquè es vegin clar les diferències d'execució en el temps d'usuari, de nucli i temps real als experiments que feu.

Per fer les proves feu servir mides de bloc de 64 bytes, 128 bytes, 512 bytes, 1024 bytes, 4096 bytes, 16384 bytes, 65536 bytes i 262144 bytes. En fer les proves es recomana executar

diverses vegades el codi per a un mateix valor per veure els marges de temps d'execució que s'obtenen.

*Quines diferències de temps d'execució (temps real, temps d'usuari i de sistema) veieu entre fer les proves amb les funcions read/write i fread/write? Raoneu aquestes diferències. Comenteu, entre altres coses, per què en alguns casos hi ha diferències clares de temps d'execució entre les dues formes de llegir les dades de disc? Per què el temps real és més gran que la suma del temps d'usuari i el temps de sistema?*

## Avaluació

Es demana entregar un **petit informe de màxim tres pàgines** (sense incloure la portada/index en cas que els vulgueu incloure) que inclogui, en taules, els temps execució d'usuari, de sistema i real obtinguts per a les funcions read/write i fread/write fent servir les mides de bloc esmentades abans. Recordeu buidar la memòria cau per a cada experiment. Comenteu els resultats obtinguts.

Executeu tots els experiments en un únic ordinador. En cas que tingueu accés a ordinadors amb discs diferents (per exemple, un disc magnètic i un disc SSD) també és interessant que feu proves amb aquests dos discos. A nivell d'avaluació, però, és més que suficient fer les proves amb un únic ordinador.

A l'informe se us demana incloure informació sobre:

- a) La plataforma en què heu executat les proves (màquina virtual, màquina nativa)
- b) RAM de la màquina virtual (o la màquina nativa) en què s'han executat les proves. Indiqueu la mida del fitxer generat.
- c) Informació (si la coneixeu) respecte el disc en què s'han fet les proves: magnètic, SSD

Sigueu breus i concisos a en comentar els resultats obtinguts.

**Entregueu un únic informe en format PDF per parella (grupXX.pdf**, on XX és el nombre del vostre grup de pràctiques). L'avaluació per part del professor serà de 0, 2.5, 5, 7.5 o 10. Per fer aquesta avaluació es faran servir criteris equivalents als que es fan als informes que entregueu a pràctiques.

Es valorarà molt més el fet que interpreteu i comenteu els resultats obtinguts amb un únic ordinador que el fet que feu experiments amb múltiples ordinadors i no feu comentaris concloents respecte els resultats obtinguts.