

Projecte 2

Mòdul de Kernel

Decisions preses:

Els mòduls que hem dissenyat intenten ser tan independents com sigui possible. És per això que totes les operacions d'accés a les estadístiques les fa el mòdul 1 i permet el control i consulta d'aquestes al mòdul dos mitjançant les següents funcions (les quals són exportades a la taula de símbols del mòdul):

```
modul_stats_catch_syscall (activa/desactiva el control de les syscalls)
modul_stats_reset_process (reinicia les estadístiques del processus)
modul_stats_get_stats      (obté les estadístiques del procés)
```

Així doncs el mòdul 2 depèn del mòdul 1, i no és possible carregar-lo sense carregar abans el mòdul 1.

El mòdul 2 crea un manegador que, en ser llegit, escriu les estadístiques en format text. Hem implementat el manegador de manera que si es crida a `ioctl` sobre el dispositiu el buffer es reinicia, és a dir que si es torna a fer un `read` es llegeix el text del buffer des del principi. Això ho hem fet així per evitar obrir i tancar el mòdul cada vegada que volem llegir de nou les estadístiques actualitzades.

A més tots els wrappers de les syscalls estan fets amb macros i taules per a poder afegir una nova syscall sense tenir que modificar massa codi (només constants i declaracions).

Jocs de prova:

El joc de prova que presentem treballa sobre un fitxer (el manegador del dispositiu que hem creat) com argument 1 i un número com argument 2. El programa executa aquest bucle:

```
for (i = 0; i < numw; i++) {
    write(fd2,&mychar,1);
    write(fd2,&mychar,1);
    write(fd2,&mychar,-1);
}
```

És a dir, escriu dues vegades correctament i una malament (count negatiu). D'aquesta manera provem la crida `write`, `open` i `close`. I també provem el comptador del temps (que ens diu quan de temps em estat a dins d'una crida).