

Pràctica 4: Encripta-RR

Sistemes operatius 2023-24

Termini d'entrega del qüestionari pel moodle:

14 de maig → grups 1, 3, i 4

8 de maig → grup 2, 5, 6, 7 i 8

Encripta-RR (round robin)

- Tindrem dos programes diferents que bàsicament generaran nous *passwords* (força dolents, per cert) a partir de l'enciptació de paraules introduïdes per teclat. .



- Exemples:

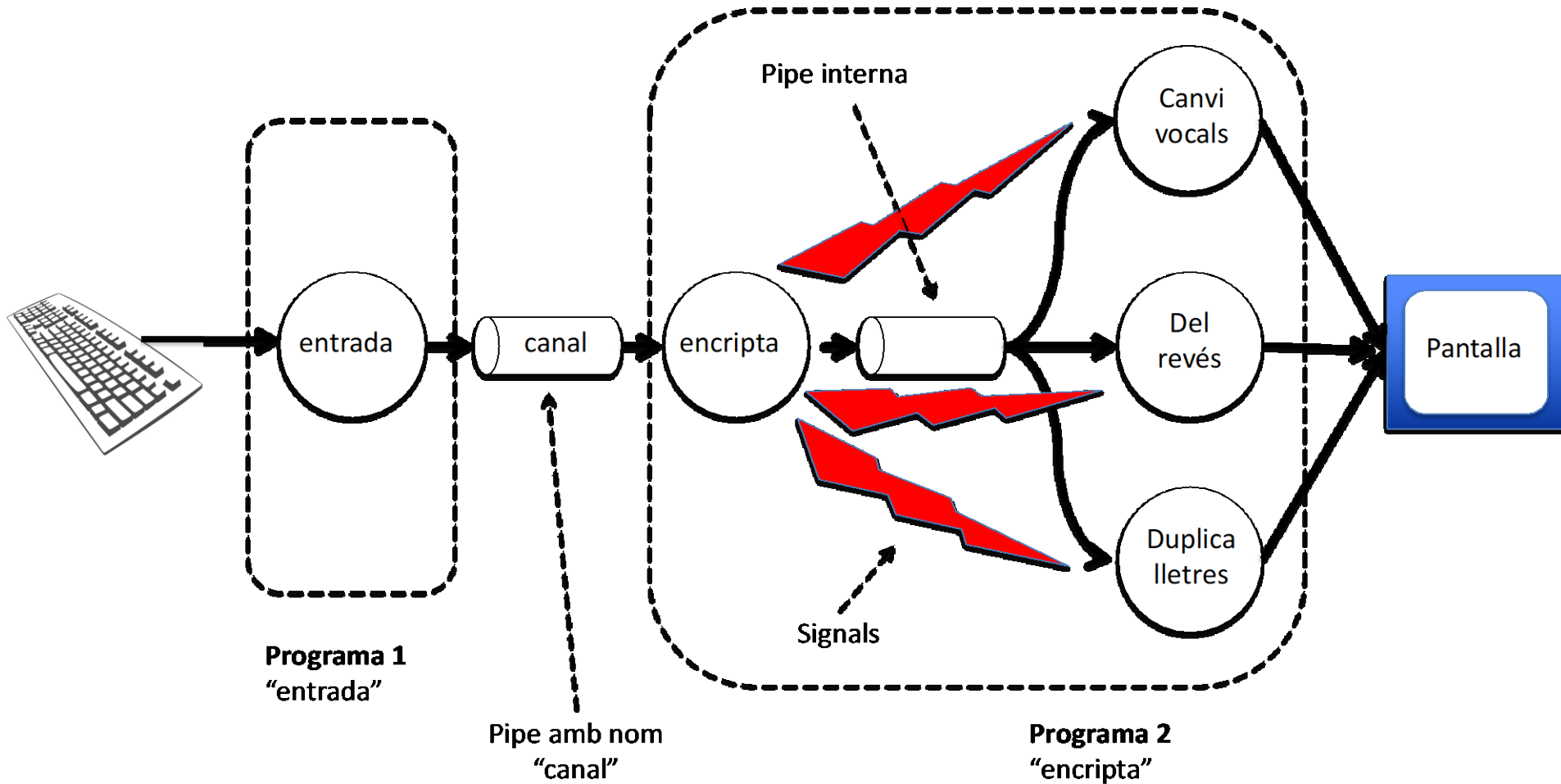


Detall

- Es tracta de fer una aplicació “**entrada**” que accepta cadenes de text per teclat i les envia per una *pipe* amb nom a una altra aplicació “**encripta**” que encripta aquestes cadenes seguint diferents mètodes. Concretament, com que encriptar pot ser una tasca feixuga que trigui un cert temps, el que fa és tenir 3 processos fills que s’encarreguen d’aquesta feina i els va passant les cadenes per encriptar a través d’una pipe interna. Es segueix una política de Round-Robin per anar passant la feina als processos fills.
- L’estructura (es mostra a la següent transparència):
 - El programa “entrada” anirà acceptant cadenes de text d’entrada i les anirà escrivint a la *pipe* amb nom fins que l’entrada sigui dos caràcters “XX”. Això indica el final del procés d’entrada, que tanca la *pipe* amb nom i finalitza. La cadena “XX” NO s’ha d’enviar per la *pipe* ja que no és cap cadena a encriptar.
 - El programa “encripta” obre la *pipe* amb nom “canal” i espera a rebre cadenes de text per encriptar-les. Per cada cadena que rep, l’envia a través d’una **única** *pipe* normal a un dels processos que es dediquen a encriptar seguint una tècnica Round-Robin. El procés “encripta” avisa al procés que li ha tocat la feina enviant-li un signal **SIGUSR1**. Llavors aquell procés es desperta, llegeix de la pipe interna la cadena i es posa a fer la feina d’encriptar. Quan acaba mostra el resultat per pantalla.

Estructura

- Hi ha 2 programes i 5 processos en total:



Detall

- Quan el procés “encripta” detecta que el procés “entrada” ha finalitzat, ha d’avisar a tots els seus processos fills amb un senyal **SIGUSR2** perquè també finalitzin, i un cop han finalitzat tots llavors també acabarà el procés “encripta”. El procés “encripta” és l’encarregat de crear la *pipe* amb nom “canal” a l’inici de la seva execució i de tancar-la i esborrar-la al final del programa.
- Els processos fills són els encarregats de fer l’encriptació real. El primer canvia totes les vocals de la cadena per caràcters numèrics segons el següent: a→4, e→3, i→1, o→0, u→1. Per exemple, “hola” queda encriptat com “h0l4”. El segon procés gira la cadena del revés. Per exemple “hola” queda encriptat com “aloh”. El tercer procés duplica les lletres. Per exemple “hola” queda encriptat com “hhoollaa”. La seva tasca és doncs esperar a rebre un *signal* per llegir de la *pipe* interna una cadena i encriptar-la i mostrar-la per pantalla.

Notes

- Cal obrir primer el programa “encripta” (es pot deixar en *background* o utilitzar 2 sessions) i després el programa “entrada”.
- L’entrada serà sempre com a **cadena** o **string**. Podem suposar una mida màxima de 30 caràcters.
- Es recomana utilitzar les crides `read()` i `write()` per llegir/escriure de teclat, pipe amb nom, pipe normal, pantalla. La crida `read()` retorna el número de caràcters llegits i és molt important tenir-los en compte. Aquest valor es pot utilitzar a la crida `write()` on cal especificar quants caràcters del buffer es volen escriure.
- En aquesta pràctica **NO cal redirigir** els canals d’entrada i sortida estàndards cap o des de les *pipes*.
- En aquesta pràctica no hi ha cap part opcional.