

Elaborato 2: Utilizzo della system call fork, exec, signal, fifo, pipe.

Consegna: entro il 05 Giugno 2006 ore 23:59

Modalità di consegna:

- 1) Rinominare il file contenente l'elaborato con il proprio numero di matricola. Si ricorda che la consegna è individuale, pertanto ogni studente dovrà consegnare una copia dell'elaborato.
- 2) Riportare in calce al file contenente l'elaborato un commento che includa: matricola, nome e cognome, data di consegna, titolo dell'elaborato.
- 3) Fare l'upload del file su <http://amarena.sci.univr.it/>
 - a. Seguire i link: Accesso pubblico Sistemi Operativi LABORATORIO 2006 Elaborati "Nome_docente_del_corso" Elaborato 2
 - b. A questo punto dovrete trovarvi all'interno di anonymous / Sistemi Operativi LABORATORIO 2006 Membri / Elaborati / Nome_docente_del_corso / Elaborato 2
 - c. Cliccare sulla freccia alla destra della voce Elaborato 2, quindi su Nuovo Documento
 - d. Compilare i campi del form che appare inserendo il file di cui fare l'upload in "File locale", il vostro nome, cognome e n° di matricola su "Nome del documento".
 - e. Premere OK
- 4) Si ricorda inoltre che non si potranno né modificare né visualizzare i file di cui è stato fatto l'upload.
- 5) Per qualunque problema durante la sottomissione dell'elaborato contattare il docente del relativo corso (Drago per Informatica, Pravadelli per Informatica Multimediale).
- 6) Dopo la scadenza del 05/06 non sarà più possibile effettuare l'upload dell'elaborato. Chi non avrà consegnato perderà definitivamente il diritto di fare l'esame nella modalità orale.

Testo dell'elaborato

Realizzare due programmi che, una volta lanciati su 2 shell separate, saranno in grado di interagire secondo le seguenti specifiche.

Programma 1

Il programma 1 deve creare 2 processi: un client e un server. Il client deve visualizzare un menu e mettersi in attesa che l'utente scelga di eseguire una delle operazioni presenti nel menu.

Il server invece deve mettersi in attesa che il client gli chieda di fare qualcosa.

Il menu visualizzato dal client dovrà essere il seguente:

1. ps -aux | grep root
2. firefox &
3. exit

Il client, dopo aver letto da tastiera la scelta fatta dall'utente, deve inviare un segnale al server (tramite la system call kill) per chiederli di eseguire l'operazione desiderata, nel caso in cui l'utente abbia premuto 1 o 2. Nel caso in cui l'utente scelga la voce di menu 3, il client deve fare in modo che il server termini la sua esecuzione, e quindi, dopo aver rilasciato eventuali risorse allocate, deve terminare a sua volta.

Il server deve intercettare l'arrivo dei segnali da parte del client, creare uno o più nuovi processi per eseguire le operazioni richieste dal client e rimettersi subito in attesa di ulteriori richieste. Ciò significa che il server non deve eseguire lui stesso i comandi, ma deve demandarli a processi figli. L'output dei comandi eseguiti dai processi figli deve essere inviato tramite FIFO al Programma 2 per la visualizzazione. La FIFO deve essere creata dal server.

Quando il server riceve dal client il segnale di terminazione, il server deve prima far terminare il Programma 2, poi rilasciare eventuali risorse allocate, e infine terminare a sua volta.

Programma 2

Il programma 2 deve dare origine a un unico processo per la visualizzazione dei dati inviati dal Programma 1. I dati da visualizzare devono essere letti dalla FIFO creata dal server del Programma 1 e mostrati su stdout.

Il Programma 2 deve terminare in risposta ad un segnale SIGKILL inviato dal server del Programma 1. Si ricorda che il server del Programma 1 deve conoscere il PID del Programma 2 per inviargli il segnale SIGKILL. Tuttavia, visto che Programma 1 e Programma 2 non sono “parenti”, è necessario trovare un modo affinché il server del Programma 1 possa conoscere il PID del Programma 2. Si lascia libertà allo studente di proporre la soluzione desiderata.