

## Elaborato: 2 – System Call

Utilizzo delle system Call UNIX (Linux) tramite la programmazione C.

Realizzare una mini shell (programma eseguibile ottenuto dalla compilazione di codice C) che implementi le seguenti funzionalità:

- 1) Esecuzione in foreground di un processo (un qualunque eseguibile presente su disco), al termine viene ritornato alla propria shell il controllo
- 2) Intercettazione del comando Ctrl – C che non deve terminare la shell, ma deve avvisare di digitare exit per chiudere
- 3) Interpretazione del comando exit, che dovrà avvisare in caso di processi in background lanciati dalla shell che anche essi verranno terminati, e in caso di risposta affermativa, terminazione di tali processi e della shell stessa.
- 4) Implementazione di un comando per far partire in background un processo, (potrebbe essere un precomando o un'argomento da inserire alla fine)
- 5) Implementazione della PIPE tra 2 processi
- 6) Implementazione come comando interno del comando jobs (che non deve essere eseguito da disco, ma una funzionalità interna) e elenca i processi attivi, figli della shell che avete implementato
- 7) Il comando interno fg, che blocca la shell in attesa che il processo selezionato termini
- 8) Il comando kill (anche questo interno) che invia un segnale a un processo pid espresso dopo il comando

In DETTAGLIO:

- 1) Esecuzione in foreground di un processo:  
La shell creerà un figlio che poi farà l'exec di un programma letto da dal terminale (è titolo preferenziale gestire anche i parametri).
- 2) Ctrl - C:  
Deve essere essere intercettato il comando Ctrl – C che non deve terminare la shell ma segnalare la corretta procedura
- 3) exit:  
Exit, comporta la chiusura della shell, il comando deve avvisare l'utente di eventuali processi attivi lanciati dalla vostra shell e se l'utente decide di proseguire verranno terminati.
- 4) Background:  
Implementazione dell'esecuzione in background di comandi, può essere usato un comando speciale prima o un simbolo dopo &. Si tenga presente che la shell deve avere sempre "coscienza" di quanti processi sono in esecuzione per poterli poi terminare, il sistema dovrebbe supportare almeno l'esecuzione in background di 10 o più processi.
- 5) Pipe:  
Deve essere implementata la | di unix.
- 6) Jobs:  
Deve essere un comando interno (non eseguire file esterni) che elencano i processi lanciati dalla shell e ancora attivi.
- 7) fg:  
fg seguito da un pid o proces id a vostra scelta deve portare in foreground un dato processo e bloccare la shell fino alla terminazione del processo
- 8) kill:  
Il comando kill deve essere implementato come system call interna per inviare un dato segnale a un processo

La mancanza di alcune funzionalità non comportano il non superamento dell'esame, ma una penalizzazione in termini di punteggio.

Il programma deve girare senza errori...

È titolo preferenziale per la valutazione (ma non obbligatorio) il controllo dei casi particolare

In calce al file come commento deve essere riportata:

/\*Matricola

Nome e cognome

Data di realizzazione

Titolo esercizio

\*/