

SISTEMI OPERATIVI/SISTEMI OPERATIVI E LAB.

(A.A. 23-24) – 30 MAGGIO 2024

Esercizio

Si realizzi un programma **concorrente** per UNIX che deve avere una parte in **Bourne Shell** (già svolta) e una parte in **C**.

TESTO PARTE C: ATTENZIONE LEGGERE ANCHE LA NOTA SEGUENTE AL TESTO!

La parte in C accetta un numero *fisso* di **due** parametri (*da controllare*) che rappresentano il primo un numero **N** intero strettamente positivo (*da controllare*) e il secondo il nome di un file relativo semplice (*da NON controllare*) che andrà creato (o sovrascritto in caso di sua esistenza). Il processo padre deve, *per prima cosa*, creare nella directory *corrente* un file con *nome* corrispondente al secondo parametro. Quindi, il processo padre deve generare **N** processi figli: ogni processo figlio **P_n** deve creare a sua volta un processo nipote **PP_n**; i processi figli e nipoti eseguono concorrentemente.

Il compito di ogni processo nipote **PP_n** è quello di ricavare l'elenco semplice dei processi associati alla sessione interattiva corrente, usando in modo opportuno il comando *ps* di UNIX/Linux.

Ogni processo figlio **P_n**, a sua volta, deve -filtrando quanto ricevuto dal proprio nipote **PP_n** - selezionare la linea del comando *ps* che corrisponde **al PID del processo nipote**, usando in modo opportuno il comando-filtro *grep* di UNIX/Linux. **Tale linea selezionata** deve contestualmente essere inviata al padre.

Il padre deve ricevere, *rispettando l'ordine dei figli*, ogni singola linea inviata dai figli; al termine della lettura di ogni singola linea, il padre deve scrivere tale linea (insieme con il terminatore di linea) sul file precedentemente creato.

Al termine, ogni processo figlio **P_n** torna al padre il risultato del comando-filtro *grep* eseguito e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato. Si inserisca un opportuno controllo nel caso questo valore non rappresenti il valore di successo del comando-filtro *grep*!

NOTA BENE NEL FILE C main.c SI USI OBBLIGATORIAMENTE:

- una variabile di nome **N** per il numero corrispondente al primo parametro;
- una variabile di nome **outfile** per il file descriptor usato dal padre per la creazione del file;
- una variabile di nome **n** per l'indice dei processi figli;
- una variabile di nome **buffer** per leggere, da parte del padre, via via le singole linee inviate dai figli.

OSSERVAZIONE SUL TESTO:

Ogni coppia costituita da un nipote e da un figlio realizza un piping di comandi, che sono quelli indicati nel testo. Rispetto ad un semplice piping di due comandi, in più ogni figlio si deve comportare in modo analogo al proprio nipote per quanto riguarda lo standard output, ma per mandare le informazioni al padre che agisce a livello di programmazione recuperando in modo opportuno quanto scritto da ogni figlio!

PRECISAZIONI DATE DURANTE LA VERIFICA:

- La variabile **buffer**, array di char usato per leggere le singole linee, si può considerare di dimensione **250**!
- Poiché la specifica indica "i processi figli e nipoti eseguono concorrentemente.", risulta opportuno che i figli **NON ASPETTINO** i nipoti!