

SISTEMI OPERATIVI E LAB.

(A.A. 21-22) – 7 SETTEMBRE 2022

IMPORTANTE:

SEGUIRE TUTTE LE REGOLE FORNITE PRIMA DELLO SVOLGIMENTO DELL'ESAME!

Esercizio

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

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

La parte in Shell deve prevedere un numero variabile di parametri **Q+1** (con **Q** maggiore o uguale a **2**): il primo parametro deve essere considerato un numero **intero strettamente positivo e dispari (X)**, mentre gli altri **Q** devono essere **nomi assoluti di directory** che identificano **Q** gerarchie (**G1, G2, ...**) all'interno del file system. Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in **Q** fasi, una per ogni gerarchia.

Il programma, per ognuna delle **Q** fasi, deve esplorare la gerarchia **G** corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e deve cercare in ogni directory della gerarchia **G** i file *leggibili* **F** la cui lunghezza in caratteri sia esattamente uguale a **X**: **nel caso tali file F siano in posizione dispari*** si deve riportare il loro nome assoluto sullo standard output. In ogni directory dove è stato trovato *almeno* un file **(CHE SODDISFA LE CONDIZIONI PRECEDENTI)**, si deve invocare la parte in C, passando come parametri i nomi dei file trovati (**F1, F2, ...**) che soddisfano la condizione precedente.

NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:

- una variabile di nome **X** per contenere il primo parametro di FCP.sh;
- una variabile di nome **G** per le singole gerarchie di ognuna delle **Q** fasi;
- una variabile di nome **F** per identificare, via via, i singoli file delle directory esplorate.

OSSERVAZIONE: se per provare la parte shell, si commenta la chiamata alla parte C, ricordarsi di togliere il commento prima della consegna!

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

La parte in C accetta un numero variabile di parametri **N** maggiore o uguale a **2** che rappresentano nomi di file (**F1, ...FN**), che hanno tutti la stessa lunghezza dispari (*da non controllare*). Il processo padre deve, per prima cosa, creare nella directory corrente un file **fcreato** con nome corrispondente al proprio COGNOME (tutto scritto in maiuscolo, in caso di più cognomi se ne usi solo uno, inserendo un opportuno commento).

Il processo padre deve generare un numero di **processi figli** pari a **N**: ogni processo figlio **Pn** è associato ad uno dei file **F1, ...FN** (*in ordine*). Ognuno di tali processi figli **Pn** esegue concorrentemente e legge tutti i caratteri del proprio file associato operando una opportuna selezione come indicato nel seguito.

I processi figli **Pn** devono usare uno schema di comunicazione a pipeline: il figlio **P0** comunica con il figlio **P1** che comunica con il figlio **P2** etc. fino al figlio **PN-1** che comunica con il padre. Questo schema a pipeline deve prevedere l'invio in avanti, per ognuno dei caratteri dispari, di un array di grandezza **N** e in cui ogni elemento dell'array corrisponde al carattere corrente **car di posizione dispari** (*il primo carattere ha posizione 1*) letto dal corrispondente processo figlio **Pn**. Quindi, il generico processo **Pn**, dopo aver letto il carattere dispari corrente, deve ricevere dal figlio precedente (a parte il processo **P0**) l'array di caratteri dispari e, dopo aver inserito il proprio carattere dispari corrente nella posizione giusta dell'array, deve inviare l'array di caratteri dispari al figlio successivo, con **PN-1** che manda al padre. Quindi al padre deve arrivare, per ognuno dei caratteri dispari, un array di caratteri dispari, ognuno dei quali deve essere scritto sul file **fcreato**. Al termine dell'esecuzione, ogni figlio **Pn** ritorna al padre l'ultimo carattere dispari letto dal proprio file (*sicuramente minore di 255*); il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

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

- una variabile di nome **N** per il numero di processi figli;
- una variabile di nome **n** per l'indice dei processi figli;
- una variabile **fcreato** per il file descriptor del file creato;
- una variabile di nome **car** per il carattere letto correntemente dai figli dal proprio file;
- una variabile di nome **cur** per l'array da passare fra i vari figli fino al padre.

* N.B. Se una dir contiene: d1 (dir), d2 (dir), f1 (file leggibile di lunghezza X), f2 (idem), f3 (idem), f4 (idem), prova.txt (file leggibile di lunghezza diversa da X), allora i file di posizione dispari che sarebbero selezionati sarebbero f1 e f3.