

SISTEMI OPERATIVI E LAB.

(A.A. 19-20) – 15 LUGLIO 2020

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 **L**, strettamente positivo, 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 **Gg** corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e deve contare **globalmente per ogni singola gerarchia Gg** tutti i file leggibile che soddisfino le seguenti specifiche:

- la lunghezza in linee sia esattamente uguale a **L**;
- la lunghezza media delle sue linee sia strettamente maggiore di **10**.

Al termine di OGNUNA delle Q fasi, si deve riportare sullo standard output il numero di file trovati in tale gerarchia Gg e, per ogni file trovato, si deve invocare la parte in C, passando come parametri F e L.

NOTA BENE NEI DUE FILE COMANDI SI USI OBBLIGATORIAMENTE:

- una variabile di nome **L** per il primo parametro;
- il nome **/tmp/nomiAssoluti\$\$** per il file temporaneo
- una variabile di nome **F** per identificare, via via, i singoli file per i quali si deve invocare la parte C.

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 due parametri **F e L** che rappresentano rispettivamente le seguenti informazioni: un nome assoluto di file e un numero strettamente positivo e minore o uguale di 255 (da controllare) che rappresenta il numero di linee del file. Il processo padre deve generare un numero di **processi figli** pari a **L**: ogni processo figlio **Pq** è associato ad una linea di **F**: la numerazione delle linee deve essere fatta partire da 1. Ognuno di tali processi figli **Pq** esegue concorrentemente e legge la propria linea associata (supposta avente lunghezza massima di 250 caratteri, compreso il terminatore di linea). Dopo la lettura della propria linea, ogni processo figlio **Pq** seleziona il secondo e il penultimo carattere della linea (*considerando il terminatore di linea come ultimo carattere*) e comunica al padre una struttura contenente tre campi: *c1* che deve contenere il pid del processo figlio; *c2* che deve contenere il secondo carattere e *c3* che deve contenere il penultimo carattere della linea. Il padre deve ricevere, rispettando l'ordine delle linee, ognuna delle strutture inviate dai figli e per ognuna deve controllare se i caratteri corrispondenti ai campi *c2* e *c3* sono **uguali: in tal caso**, deve stampare su standard output tutti i campi della struttura corrente indicando esplicitamente il loro significato e, inoltre, deve riportare anche il numero d'ordine del processo figlio che ha inviato l'informazione, il numero della linea cui tale processo era associato e il nome del file **F**; *in caso contrario, il padre non deve stampare nulla*.

Al termine dell'esecuzione, ogni figlio **Pq** ritorna al padre il numero della propria linea associata; 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 **L** per il numero di processi figli;
- una variabile di nome **q** per l'indice dei processi figli;
- una variabile di nome **linea** per memorizzare la linea associata e quindi letta da ogni figli