SISTEMI OPERATIVI/SISTEMI OPERATIVI E LAB.
(A.A. 23-24) – 11 SETTEMBRE 2024

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** (**X**) **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 **G** corrispondente - tramite un file comandi ricorsivo, **FCR.sh** – e deve cercare tutte le directory che si trovano al livello **X**; in ognuna di tali directory, si devono selezionare i file *leggibili*, *non VUOTI* (salvando contestualmente, in modo opportuno, i loro nomi relativi semplici). Al termine di tutte le **Q** fasi, si deve riportare il numero di directory trovate e poi, spostandosi via via in ognuna delle directory trovate (usando il loro nome assoluto), si deve invocare la parte in C passando come parametri il nome della directory trovata corrente e, quindi, i nomi relativi semplici dei file selezionati in tutte le directory trovate (salvati in modo opportuno, come indicato precedentemente).

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;
- la stringa **/tmp/nomi** per la parte iniziale dei nomi dei file temporanei;
- una variabile di nome **F** per identificare, via via, i singoli file delle directory trovate a livello **X**.

OSS: a) si può supporre, senza fare alcun controllo, che i nomi dei file selezionati siano tutti diversi per non avere problemi nella parte C; b) se per provare la parte shell, si commenta la chiamata alla parte C, ricordarsi di togliere il commento!

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 **3** che rappresentano: il primo il nome assoluto di una directory e gli altri nomi relativi semplici di file (**F1, …FN**); *si consideri l'OSS. a)*.

Il processo padre deve, *per prima cosa*, creare nella directory di sistema */tmp* un file con *nome RISULTATO*, se non esiste, altrimenti deve aprire tale file in append e scrivere -in una linea separata- il nome assoluto della directory passata come primo parametro. Quindi, il processo padre deve generare **N** processi figli: i processi figli **Pn** sono associati ad uno dei file **F1**, ... **FN** (*in ordine*).; ogni processo figlio **Pn** deve creare a sua volta un processo nipote **PPn**; i processi figli e nipoti eseguono concorrentemente.

Il compito di ogni processo nipote **PPn** è quello di ricavare l'elenco dei file/directory della directory corrente, considerando tutte le informazioni associate ai file/directory compreso l'informazione del loro i-number, usando in modo opportuno il comando ***ls*** di UNIX/Linux**.**

Ogni processo figlio **Pn**, a sua volta, deve -filtrando quanto ricevuto dal proprio nipote **PPn** - selezionare la linea del comando ***ls*** che corrisponde al nome del proprio file associato, 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 indicato precedentemente.

Al termine, ogni processo figlio **Pn** 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 di file;
- una variabile di nome **outfile** per il file descriptor usato dal padre per la creazione/apertura del file *RISULTATO*;
- una variabile di nome **n** per l'indice dei processi figli;
- una variabile di nome **linea** *per l'array di 250 caratteri* usato per leggere, da parte del padre, via via le singole linee inviate dai figli; *si supponga che 250 caratteri siano sufficienti per ogni linea, compreso il terminatore di linea*.

OSSERVAZIONE: 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!

IMPORTANTE:

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

ESEMPIO DI ESECUZIONE:

```
$ FCP.sh 2 `pwd`/ger `pwd`/ger1
Il numero totale di directory che soddisfano la specifica = 5
DEBUG-Le directory che abbiamo trovato a livello 2 sono le seguenti:
/home/soELab/prova/11Set24/ger/d1
/home/soELab/prova/11Set24/ger/d2
/home/soELab/prova/11Set24/ger/d3
/home/soELab/prova/11Set24/ger1/d1
/home/soELab/prova/11Set24/ger1/d3
DEBUG-chiamo la parte in C con F1 F3 F4 prova1.txt prova2.txt prova3.txt FFF F2 prova20.txt prova6.txt prova7.txt F5
DEBUG-Sono il processo padre con pid 180030 e creero' 12 processi figli per la directory /home/soELab/prova/11Set24/ger/d1
Il figlio con pid=180031 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180032 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180033 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180035 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180037 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180038 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180040 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180041 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180042 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180043 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180039 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
Il figlio con pid=180034 ha fallito l'esecuzione del comando-filtro grep oppure se 255 ha avuto dei problemi (ritorno=1)
DEBUG-chiamo la parte in C con F1 F3 F4 prova1.txt prova2.txt prova3.txt FFF F2 prova20.txt prova6.txt prova7.txt F5
```

... <omessa stampa per le altre 4 directory trovate!>

CONTENUTO DEL FILE CREATO IN /tmp DALLA PARTE C

```
/home/soELab/prova/11Set24/ger/d1
/home/soELab/prova/11Set24/ger/d2
4199208 -rw-r--r-- 1 soELab users   80 Jul 18 15:03 F1
4199220 -rw-r--r-- 1 soELab users   44 Jul 18 15:03 F3
4199209 -rw-r--r-- 1 soELab users   84 Jul 18 15:03 F4
4199217 -rw-r--r-- 1 soELab users  344 Jul 18 15:03 prova1.txt
4199218 -rw-r--r-- 1 soELab users   70 Jul 18 15:03 prova2.txt
4199219 -rw-r--r-- 1 soELab users   98 Jul 18 15:03 prova3.txt
/home/soELab/prova/11Set24/ger/d3
4199206 -rw-r--r-- 1 soELab users 26 Jul 18 15:03 FFF
/home/soELab/prova/11Set24/ger1/d1
4199242 -rw-r--r-- 1 soELab users   44 Jul 18 15:03 F2
4199241 -rw-r--r-- 1 soELab users   72 Jul 18 15:03 prova20.txt
4199239 -rw-r--r-- 1 soELab users  292 Jul 18 15:03 prova6.txt
4199240 -rw-r--r-- 1 soELab users  438 Jul 18 15:03 prova7.txt
/home/soELab/prova/11Set24/ger1/d3
4199259 -rw-r--r-- 1 soELab users   44 Jul 18 15:03 F5
```

LE PARTI IN EVIDENZIATO GIALLO SONO STATE CHIARITE DURANTE L'ESAME COMUNICANDOLO A VOCE ALTA!