

SISTEMI OPERATIVI/SISTEMI OPERATIVI E LAB.

(A.A. 23-24) – 6 NOVEMBRE 2024 (Riservato Laureandi)

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**: **si riporti su standard output il nome assoluto di tali directory**. In ognuna di tali directory, si deve verificare che esistano almeno 2 file e, in tal caso, si deve invocare la parte in C passando come parametri il nome della directory corrente e, quindi, i nomi relativi semplici dei file trovati; altrimenti, si deve dare una indicazione all'utente che non è stato possibile invocare la parte C!

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 **conteggio** per contare i file trovati;
- una variabile di nome **F** per identificare, via via, i singoli file delle directory trovate a livello **X**.

OSSERVAZIONE: 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**).

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; ogni processo figlio **Pn** deve creare a sua volta un processo nipote **PPn**: i processi nipoti **PPn** sono associati ad uno dei file **F1, ... FN (in ordine)**; i processi figli e nipoti eseguono concorrentemente.

Il compito di ogni processo nipote **PPn** è quello di ricavare tutte le informazioni associate al proprio file associato, compreso l'informazione del suo i-number, usando in modo opportuno il comando **ls** di UNIX/Linux.

Ogni processo figlio **Pn**, a sua volta, deve ricevere la linea inviata dal proprio processo nipote **PPn** e la deve scrivere (insieme con il terminatore di linea) sul file indicato precedentemente. Tale linea quindi deve 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, dopo avere trasformato la linea letta in una stringa, la deve scrivere (insieme con il terminatore di linea) sullo standard output insieme con l'indice del figlio, al nome del file cui le informazioni si riferiscono e al nome della directory.

Al termine, ogni processo figlio **Pn** torna al padre il risultato del comando-filtro **ls** eseguito dal proprio nipote **PPn** e 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 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 di ogni figlio e del padre, via via le singole linee inviate dai nipoti/figli; *si supponga che 250 caratteri siano sufficienti per ogni linea, compreso il terminatore di linea e di stringa.*

IMPORTANTE:

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

ESEMPIO DI ESECUZIONE:

```
$ FCP.sh 2 `pwd`/ger `pwd`/ger1
TROVATA directory /home/soELab/6Nov24/ger/d1 che si trova a livello GIUSTO 2
Nella dir /home/soELab/6Nov24/ger/d1 NON sono stati trovati almeno 2 file dato che files= f6 e quindi NON chiamiamo la parte C
TROVATA directory /home/soELab/6Nov24/ger/d2 che si trova a livello GIUSTO 2
DEBUG-chiamiamo la parte C passando come parametri /home/soELab/6Nov24/ger/d2 e F1 F3 F4 f4 f5 prova1.txt prova2.txt prova3.txt
La linea comunicata dal figlio di indice 0 e relativa al file F1 della directory /home/soELab/6Nov24/ger/d2 e' la seguente:
4199293 -rw-r--r-- 1 soELab users 80 Oct 10 17:27 F1
La linea comunicata dal figlio di indice 1 e relativa al file F3 della directory /home/soELab/6Nov24/ger/d2 e' la seguente:
4199305 -rw-r--r-- 1 soELab users 44 Oct 10 17:27 F3
La linea comunicata dal figlio di indice 2 e relativa al file F4 della directory /home/soELab/6Nov24/ger/d2 e' la seguente:
4199294 -rw-r--r-- 1 soELab users 84 Oct 10 17:27 F4
La linea comunicata dal figlio di indice 3 e relativa al file f4 della directory /home/soELab/6Nov24/ger/d2 e' la seguente:
4199300 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f4
La linea comunicata dal figlio di indice 4 e relativa al file f5 della directory /home/soELab/6Nov24/ger/d2 e' la seguente:
4199301 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f5
La linea comunicata dal figlio di indice 5 e relativa al file prova1.txt della directory /home/soELab/6Nov24/ger/d2 e' la seguente:
4199302 -rw-r--r-- 1 soELab users 344 Oct 10 17:27 prova1.txt
La linea comunicata dal figlio di indice 6 e relativa al file prova2.txt della directory /home/soELab/6Nov24/ger/d2 e' la seguente:
4199303 -rw-r--r-- 1 soELab users 70 Oct 10 17:27 prova2.txt
La linea comunicata dal figlio di indice 7 e relativa al file prova3.txt della directory /home/soELab/6Nov24/ger/d2 e' la seguente:
4199304 -rw-r--r-- 1 soELab users 98 Oct 10 17:27 prova3.txt
Il figlio con pid=66775 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66776 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66777 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66778 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66780 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66782 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66784 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66781 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
TROVATA directory /home/soELab/6Nov24/ger/d3 che si trova a livello GIUSTO 2
Nella dir /home/soELab/6Nov24/ger/d3 NON sono stati trovati almeno 2 file dato che files= FFF e quindi NON chiamiamo la parte C
TROVATA directory /home/soELab/6Nov24/ger1/d1 che si trova a livello GIUSTO 2
DEBUG-chiamiamo la parte C passando come parametri /home/soELab/6Nov24/ger1/d1 e F2 f4 f5 prova20.txt prova6.txt prova7.txt
La linea comunicata dal figlio di indice 0 e relativa al file F2 della directory /home/soELab/6Nov24/ger1/d1 e' la seguente:
4199327 -rw-r--r-- 1 soELab users 44 Oct 10 17:27 F2
La linea comunicata dal figlio di indice 1 e relativa al file f4 della directory /home/soELab/6Nov24/ger1/d1 e' la seguente:
4199322 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f4
La linea comunicata dal figlio di indice 2 e relativa al file f5 della directory /home/soELab/6Nov24/ger1/d1 e' la seguente:
4199323 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f5
La linea comunicata dal figlio di indice 3 e relativa al file prova20.txt della directory /home/soELab/6Nov24/ger1/d1 e' la seguente:
4199326 -rw-r--r-- 1 soELab users 72 Oct 10 17:27 prova20.txt
La linea comunicata dal figlio di indice 4 e relativa al file prova6.txt della directory /home/soELab/6Nov24/ger1/d1 e' la seguente:
4199324 -rw-r--r-- 1 soELab users 292 Oct 10 17:27 prova6.txt
La linea comunicata dal figlio di indice 5 e relativa al file prova7.txt della directory /home/soELab/6Nov24/ger1/d1 e' la seguente:
4199325 -rw-r--r-- 1 soELab users 438 Oct 10 17:27 prova7.txt
Il figlio con pid=66813 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66816 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66817 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66818 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66819 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66814 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
TROVATA directory /home/soELab/6Nov24/ger1/d3 che si trova a livello GIUSTO 2
DEBUG-chiamiamo la parte C passando come parametri /home/soELab/6Nov24/ger1/d3 e F5 f6
La linea comunicata dal figlio di indice 0 e relativa al file F5 della directory /home/soELab/6Nov24/ger1/d3 e' la seguente:
4199344 -rw-r--r-- 1 soELab users 44 Oct 10 17:27 F5
La linea comunicata dal figlio di indice 1 e relativa al file f6 della directory /home/soELab/6Nov24/ger1/d3 e' la seguente:
4199343 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f6
Il figlio con pid=66837 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
Il figlio con pid=66838 ha ritornato 0 (se 255 problemi o nel figlio o nel nipote)
```

CONTENUTO DEL FILE CREATO IN /tmp

```
/home/soELab/6Nov24/ger/d2
4199304 -rw-r--r-- 1 soELab users 98 Oct 10 17:27 prova3.txt
4199302 -rw-r--r-- 1 soELab users 344 Oct 10 17:27 prova1.txt
4199300 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f4
4199293 -rw-r--r-- 1 soELab users 80 Oct 10 17:27 F1
4199294 -rw-r--r-- 1 soELab users 84 Oct 10 17:27 F4
4199305 -rw-r--r-- 1 soELab users 44 Oct 10 17:27 F3
4199303 -rw-r--r-- 1 soELab users 70 Oct 10 17:27 prova2.txt
4199301 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f5
/home/soELab/6Nov24/ger1/d1
4199325 -rw-r--r-- 1 soELab users 438 Oct 10 17:27 prova7.txt
4199327 -rw-r--r-- 1 soELab users 44 Oct 10 17:27 F2
4199322 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f4
4199324 -rw-r--r-- 1 soELab users 292 Oct 10 17:27 prova6.txt
4199323 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f5
4199326 -rw-r--r-- 1 soELab users 72 Oct 10 17:27 prova20.txt
/home/soELab/6Nov24/ger1/d3
4199344 -rw-r--r-- 1 soELab users 44 Oct 10 17:27 F5
4199343 -rw-r--r-- 1 soELab users 0 Oct 10 17:27 f6
```