

# SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 17-18) – 11 LUGLIO 2018

**IMPORTANTE:** LEGGERE LE INFORMAZIONI SUL RETRO DEL FOGLIO!!!

## Esercizio

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

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

Il programma, per ognuna delle **W** fasi, deve esplorare la gerarchia **Gg** corrispondente - tramite un file comandi ricorsivo, **FCR.sh** - e deve contare **globalmente** tutti i file che saranno cercati secondo quanto di seguito specificato. Il file comandi ricorsivo **FCR.sh** deve cercare in ogni gerarchia **Gg** che esista almeno un file (**F**) che contenga (nel suo contenuto) almeno una occorrenza del carattere **CZ**: per ogni file trovato **F**, si deve riportare, contestualmente, il suo nome assoluto sullo standard output. Al termine delle **W** fasi, si deve riportare sullo standard output il numero di file (**F**) trovati e si deve invocare la parte in C, passando come parametri **CZ** e poi i nomi dei file *trovati* (**F1, F2, ... FN**).

La parte in C accetta un numero variabile **1+N** di parametri (con **N** maggiore o uguale a 2, da controllare) che rappresentano rispettivamente: il primo parametro un singolo carattere **CZ** (da controllare), mentre gli altri **N** nomi di file (**F1, F2. ... FN-1**).

Il processo padre deve generare **N processi figli (P0, P1, ... PN-1)**: i processi figli **Pi (con i che varia da 0 a N-1)** sono associati ai **N file Ff** (con  $f=i+2$ ). Ogni processo figlio **Pi** deve leggere tutti i caratteri del file associato **Ff cercando le occorrenze del carattere CZ**; appena viene trovata una occorrenza del **carattere CZ**, il processo figlio **Pi** deve comunicare al padre la **posizione** di tale carattere (in termini di *long int*) e deve ricevere dal padre l'indicazione di stampare o meno su standard output delle informazioni (*vedi dopo \**). Il padre deve ricevere, rispettando l'ordine dei file **Ff**, da ogni figlio via via i valori *long int* che rappresentano la **posizione** all'interno del file della occorrenza **corrente** trovata. Quindi, al processo padre deve arrivare **un insieme di valori long int** (che via via potrebbe diminuire in quantità in dipendenza della terminazione dei figli): sicuramente il primo insieme (questo viene garantito dalla parte Shell) è costituito dalla prima posizione inviata dal figlio **P0**, dalla prima posizione inviata dal figlio **P1, ...**, dalla prima posizione inviata dal figlio **PN-1**. Per ogni insieme ricevuto, il padre deve determinare il valore **massimo** e, SOLO AL PROCESSO FIGLIO CHE HA INVIATO TALE VALORE, deve indicare (\*) di stampare su standard output **il carattere trovato, assieme all'indice d'ordine del processo, il suo pid, la posizione e il nome del file associato**, mentre a tutti gli altri processi figli deve indicare di non stampare<sup>1</sup>.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il numero di occorrenze trovate (supposto strettamente minore di 255) del carattere **CZ**; il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

---

<sup>1</sup> Per questo tipo di interazione, volendo si possono usare i segnali. Nel caso si usino le pipe, fare attenzione che il padre deve inviare l'indicazione SOLO ai figli che non sono terminati!

## IMPORTANTE:

- 1) Fare il login sui sistemi in modalità Linux usando il proprio **username** e **password**, aprire un browser sulla pagina <ftp://lica02.lab.unimo.it/README>, copiare il comando presente in un terminale ed eseguirlo rispondendo alle domande proposte: sul Desktop, viene creata automaticamente una directory **studente\_1\_1\_XXX** al cui interno viene creato un file denominato `student_data.csv` che non va eliminato; infine, dopo avere copiato i propri file da chiavetta, passare in modalità testuale.
- 2) I file prodotti devono essere collocati nella directory **studente\_1\_1\_XXX** dato che tale directory viene zippata e salvata automaticamente sul server ad intervalli di tempo regolari. **ALLA SCADENZA DEL TEMPO A DISPOSIZIONE VERRÀ ATTIVATA UNA PROCEDURA AUTOMATICA DI ESTRAZIONE, PER OGNI STUDENTE DEL TURNO, DEI FILE CONTENUTI NELLA DIRECTORY SPECIFICATA.**
- 3) **NOVITÀ DALL'APPELLO DI LUGLIO 2016:** per facilitare le operazioni di stampa dei compiti sono imposte le seguenti regole per nominare i file da salvare nella directory **studente\_1\_1\_USERNAME**:
  - FCP.sh per il file che contiene lo script principale (quello di partenza) della parte SHELL;
  - FCR.sh per il file che contiene lo script ricorsivo della parte SHELL;
  - main.c per il file che contiene il programma della parte C;
  - makefile per il file che contiene le direttive per il comando make.

**Devono essere rispettati esattamente i nomi indicati altrimenti NON si procederà alla correzione del compito!**
- 4) NON devono essere presenti altri file con nome che termina con `.sh` o con `.c` nella directory **studente\_1\_1\_USERNAME**.
- 5) Il tempo a disposizione per la prova è di **120 MINUTI** per il compito completo e di **90 MINUTI** per lo svolgimento della sola parte C.
- 6) Non è ammesso **nessun tipo di scambio di informazioni** né verbale né elettronico, pena la invalidazione della verifica: **all'ingresso deve essere lasciato il/i cellulare/i sulla cattedra e potranno essere ripresi solo all'uscita.**
- 7) L'assenza di commenti significativi verrà penalizzata, così come la mancanza del makefile!
- 8) **AL TERMINE DELLA PROVA È INDISPENSABILE CONSEGNARE IL TESTO DEL COMPITO (ANCHE IN CASO UNO STUDENTE SI RITIRI): IN CASO CONTRARIO, NON POTRÀ ESSERE EFFETTUATA LA CORREZIONE DEL COMPITO MANCANDO IL TESTO DI RIFERIMENTO.**
- 9) **SI RICORDA CHE IN CASO DI ESITO INSUFFICIENTE** è necessario visionare il compito prima di potersi iscrivere a qualunque appello successivo!