

# SISTEMI OPERATIVI e LAB.

## (A.A. 18-19) – 10 LUGLIO 2019

**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 cercare tutti i direttori che contengono almeno un file leggibile che contenga (nel suo contenuto) almeno in due linee differenti il carattere **Cz**: si riporti il nome assoluto di tali direttori sullo standard output. Al termine di tutte le W fasi, si deve invocare la parte in C, passando come parametri i **nomi assoluti** dei file trovati (**F1, F2, ... FN**) e il carattere **Cz**.

La parte in C accetta un numero variabile **N+1** di parametri (con **N** maggiore o uguale a 2, da controllare) che rappresentano i primi **N** nomi di file (**F1, F2. ... FN**), mentre l'ultimo rappresenta un singolo carattere **Cz** (*da controllare*).

Il processo padre deve generare **2 \* N processi figli (P0, P1 .. PN-1, PN, PN+1 ... P2\*N-1)**: i processi figli vanno considerati a **coppie**, ognuna delle quali è associata ad uno dei file **Ff** (con  $f = i+1$ , con  $i$  che varia da 0 a ... N-1 e che corrisponde al file  $F_x$  con  $x = 2*N-i$  per  $i$  da N a  $2*N-1$ ). In particolare, la prima coppia è costituita dal processo **P0** e dal processo **P2\*N-1**, la seconda dal processo **P1** e dal processo **P2\*N-2** e così via fino alla coppia costituita dal processo **PN-1** e dal processo **PN**. **Entrambi i processi della coppia** (nel seguito chiamati *primo* e *secondo* processo) **devono cercare il carattere Cz nel file associato Fi sempre** fino alla fine attuando una sorta di staffetta così come illustrato nel seguito. Il *primo* processo della coppia **Pi** (con  $i$  che varia da 0 a ... N-1) deve cominciare a leggere dal file associato **Ff** cercando la prima occorrenza del carattere **Cz**; appena trovata deve comunicare al *secondo* processo della coppia **Pi** (con  $i$  da N a  $2*N-1$ ) **la posizione** del carattere trovato all'interno del file (in termini di *long int*); quindi il secondo processo della coppia **deve partire nello stesso file associato Fx (che corrisponde a Ff) con la sua ricerca del carattere Cz dalla posizione seguente a quella ricevuta**; appena trovata una nuova occorrenza di **Cz**, deve comunicare al primo processo della coppia la posizione del carattere trovato all'interno del file (in termini di *long int*) che quindi riparte dalla posizione seguente a cercare; tale staffetta deve avere termine quando il file è finito\*.

Al termine, ogni processo figlio deve ritornare al padre il numero di occorrenze del carattere **Cz** trovate dal singolo processo della coppia (*supposto minore o uguale a 255*) e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

---

\* Si precisa che è garantito dalla parte Shell che nel file vengano trovate almeno DUE occorrenze del carattere **Cz** e quindi la staffetta viene eseguita almeno una volta dal primo processo al secondo e almeno una volta dal secondo processo al primo. SI FACCIA ATTENZIONE ALL'INDICE CHE DEVE ESSERE USATO DAL SECONDO PROCESSO PER INDIVIDUARE IL FILE SU CUI OPERARE COSÌ ALL'INDICE CHE DEVE ESSERE USATO PER LA PIPE DA USARE IN LETTURA E PER LA PIPE DA USARE IN SCRITTURA!

## 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) 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) **ATTENZIONE: IN PARTICOLARE PER LA PARTE C NON VERRANNO CORRETTE SOLUZIONI CHE PRESENTANO ERRORI RIPORTATI DAL COMANDO gcc ALL'INVOCAZIONE DEL COMANDO make!!!!**
- 5) NON devono essere presenti altri file con nome che termina con `.sh` o con `.c` nella directory **studente\_1\_1\_USERNAME**.
- 6) 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.
- 7) 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.**
- 8) L'assenza di commenti significativi verrà penalizzata, così come la mancanza del makefile!
- 9) **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.**
- 10) **SI RICORDA CHE IN CASO DI ESITO INSUFFICIENTE è necessario visionare il compito prima di potersi iscrivere a qualunque appello successivo!**