

# SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 16-17) – 17 GENNAIO 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 **4 parametri**: il primo deve essere il nome assoluto di un direttorio che identifica una gerarchia (**G**) all'interno del file system, il secondo e il terzo devono essere considerati numeri interi strettamente positivi (**N** e **H**), mentre il quarto deve essere considerato un singolo carattere (**Cx**). Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono **esattamente N** file che abbiano un numero di linee uguale a **H** e che contengano (nel contenuto) almeno una occorrenza del carattere **Cx**. Si riporti il nome assoluto di tali direttori sullo standard output. In ognuno di tali direttori trovati, si deve invocare la parte in C, passando come parametri i nomi degli **N** file trovati (**F1, ... FN**) che soddisfano la condizione precedente, il carattere **Cx** e il numero intero **H**.

La parte in C accetta un numero variabile **N+2** di parametri (con **N** maggiore o uguale a **1**) che rappresentano i primi **N** nomi di file (**F1, ... FN**), mentre il penultimo rappresenta un singolo carattere (**Cx**) (da controllare) e l'ultimo rappresenta un numero intero (**H**) strettamente positivo (da controllare) che indica la lunghezza in linee dei file: infatti, la lunghezza in linee dei file è la stessa (questo viene garantito dalla parte shell e NON deve essere controllato).

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 agli **N** file **Fk** (con  $k=i+1$ ). Ogni processo figlio **Pi** deve leggere TUTTE le linee del file associato **Fk** fino alla fine. I processi figli e il processo padre devono attenersi a questo **schema di comunicazione a pipeline**: il figlio **P0** comunica con il figlio **P1** che comunica con il figlio **P2** etc. fino al figlio **PN-1** che comunica con il **padre**: Questo schema a pipeline deve essere ripetuto **H** volte e cioè per ogni linea letta dai file associati **Fk** e deve prevedere l'invio in avanti, per ogni linea letta, di una **struttura** che deve contenere due campi, **c1** e **c2**, con **c1** uguale all'indice d'ordine di un processo e con **c2** uguale al numero di occorrenze del carattere **Cx** nella linea corrente. In particolare, il figlio **P0** passa in avanti (cioè comunica) per ogni linea letta una struttura **S0**, con **c1** uguale a 0 e con **c2** uguale al numero di occorrenze del carattere **Cx** nella linea corrente; il figlio seguente **P1**, dopo la lettura della propria linea corrente, confronta il numero di occorrenze del carattere **Cx** nella propria linea corrente con il valore ricevuto da **P0** e se il proprio conteggio è strettamente maggiore confeziona la struttura **S1** con i propri dati e la passa al figlio seguente **P2** altrimenti passa avanti la struttura **S0** ricevuta, etc. fino al figlio **PN-1**, che si comporta in modo analogo, ma passa al **padre**. Quindi, al processo padre devono arrivare H strutture, una per ogni linea letta dai processi P0 ... PN-1. Il padre per ogni linea riceve, quindi, l'informazione di quale figlio (in termini di indice) ha letto la linea con il numero di occorrenze maggiore per la linea corrente: il padre, per ogni informazione ricevuta, deve indicare\* al figlio il cui indice risulta nella struttura ricevuta di stampare la sua linea corrente su standard output, insieme con l'indice del processo, il suo pid e il numero d'ordine della linea, mentre agli altri figli deve indicare di non stampare nulla.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il numero di linee stampate su standard output e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

---

\* Volendo per questo tipo di interazione si possono usare i segnali.

## 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\_USERNAME** 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\_USERNAME** 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!