

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 14-15) – 10 GIUGNO 2015

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 **3 parametri**: il primo deve essere il nome assoluto di un direttorio che identifica una gerarchia (**G**) all'interno del file system, il secondo deve essere il nome relativo semplice di un direttorio (**D**), mentre il terzo parametro deve essere considerato un numero intero strettamente positivo (**H**). Il programma deve cercare nella gerarchia **G** specificata tutti i direttori di nome **D** che contengono almeno *un* file che abbia un numero di linee uguale a **H**. Si riporti il nome assoluto di tali direttori sullo standard output. Al termine dell'intera esplorazione ricorsiva di **G**, si deve invocare la parte in C, passando come parametri i nomi dei file trovati (**F0, F1, ... FN-1**) -che soddisfano la condizione precedente- e il numero intero **H**.

La parte in C accetta un numero variabile **N+1** di parametri (con **N** maggiore o uguale a **2**) che rappresentano i primi **N** nomi di file (**F0, F1, ... FN-1**), mentre 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, per prima cosa, chiedere all'utente un carattere **Cx** e quindi deve generare **N processi figli (P0 ... PN-1)** ognuno dei quali è associato ad uno dei file **Fi**. Ogni processo figlio **Pi** deve leggere le linee del file associato **Fi sempre** 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 **Fi** e deve prevedere l'invio in avanti, per ogni linea letta, via via 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 via via 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 minore 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 sia di quale figlio (in termini di indice) ha letto la linea con il numero di occorrenze minore per la linea corrente e sia del numero di tali occorrenze: il padre, deve riportare tale informazione su standard output insieme al numero d'ordine di linea cui si riferiscono e al pid del processo che ha trovato il numero di occorrenze minore.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il valore intero **0** se l'esecuzione è avvenuta con successo, altrimenti deve tornare un valore intero **maggiore di 0** se ci sono stati dei problemi; il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

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_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_XXX dato che tale directory viene zippata e salvata automaticament 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 DIRETTORY SPECIFICATA.**
- 3) Il tempo a disposizione per la prova è di **120 MINUTI** per lo svolgimento di tutto il compito e di **90 MINUTI** per lo svolgimento della sola parte C.
- 4) Non è ammesso **nessun tipo di scambio di informazioni** né verbale né elettronico, pena la invalidazione della verifica.
- 5) L'assenza di commenti significativi verrà penalizzata, così come la mancanza del makefile!
- 6) **AL TERMINE DELLA PROVA È INDISPENSABILE CONSEGNARE IL TESTO DEL COMPITO (ANCHE IN CASO CHE UNO STUDENTE SI RITIRI): IN CASO CONTRARIO, NON POTRÀ ESSERE EFFETTUATA LA CORREZIONE DEL COMPITO MANCANDO IL TESTO DI RIFERIMENTO.**