SISTEMI OPERATIVI E LAB.
(A.A. 17-18) – 13 FEBBRAIO 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 **N+1** di parametri: il primo deve essere il **nome assoluto di una directory** che identifica una gerarchia (**G**) all'interno del file system, mentre gli altri **N** parametri (con **N** maggiore o uguale a 2) devono essere nomi relativi semplici di file (**F1, F2, … FN**). Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono almeno la metà degli **N** file **F1, F2, … FN (F1, F2, … FM con N/2 <= M <= N)**: si riporti il **nome assoluto** di tali direttori sullo standard output. Per ogni direttorio trovato, si deve invocare la parte in C passando come parametri i nomi relativi semplici degli **M** file trovati **F1, F2, … FM**.

La parte in C accetta un numero variabile M di parametri (con M maggiore o uguale a 1, da controllare) che rappresentano nomi di file (**F1, F2, ... FM**).
Il processo padre deve generare **M** processi figli (**P0, P1, … PM-1**): i processi figli **Pi** (con i che varia da 0 a M-1) sono associati agli **M** file Fk (con k= i+1). Ogni processo figlio **Pi** deve calcolare (in termini di *long int*) la lunghezza in linee del file associato Fk. Quindi, 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 **PM-1** che comunica con il padre. Questo schema a pipeline deve prevedere l'invio in avanti di una singola struttura dati, che deve contenere due campi: 1) *c1*, di tipo *long int*, che deve contenere il valore massimo di linee calcolato dal processo **Pi**; 2) *c2*, di tipo *int*, che deve contenere l'indice d'ordine (**i**) del processo che ha calcolato il massimo. Quindi la comunicazione deve avvenire in particolare in questo modo: il figlio **P0**, dopo aver calcolato il numero di linee tot0 del file F1, passa in avanti (cioè comunica) (con una singola *write*) al processo **P1** una struttura S0, con *c1* uguale a tot0 e *c2* uguale a 0; il figlio seguente **P1**, dopo aver calcolato il numero di linee tot1 del file F2, deve leggere (con una singola *read*) la struttura S0 inviata da **P0** e quindi deve confezionare la struttura S1 con *c1* uguale al massimo fra tot0 e tot1, *c2* uguale all'indice del processo che ha calcolato il valore di *c1* e la passa (con una singola *write*) al figlio seguente **P2**, etc. fino al figlio **PM-1**, che si comporta in modo analogo, ma passa al padre. Quindi, al processo padre deve arrivare la struttura **SM-1**. Il padre deve riportare i dati di tale struttura su standard output insieme al **pid** del processo che ha calcolato il massimo numero di linee e al nome del file per cui sono stati calcolati (inserendo opportune spiegazioni per l'utente).
Al termine, ogni processo figlio **Pi** deve ritornare al padre il valore intero corrisponde al proprio indice d'ordine (i); 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_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 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) **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!