SISTEMI OPERATIVI e
LABORATORIO DI SISTEMI OPERATIVI
(A.A. 16-17) – 26 MAGGIO 2017

IMPORTANTE:

LEGGERE LE INFORMAZIONI SUL RETRO DEL FOGLIO!!!

Esercizio

Si realizzi un programma **concorrente** per UNIX che deve avere una parte in **Bourne Shell** (già svolta) e una parte in **C**.

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 (**Cx**) (da controllare).
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 i caratteri del file associato **Fk** calcolando (in termini di *long int*) le occorrenze del carattere **Cx**. 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 **PN-1** che comunica con il **padre**. Questo schema a pipeline deve prevedere l'invio in avanti di una singola **struttura** dati, che deve contenere tre campi: 1) *c1*, di tipo *long int*, che deve contenere il valore massimo di occorrenze calcolato dal processo **Pi**; 2) *c2*, di tipo *int*, che deve contenere l'indice d'ordine (i) del processo che ha calcolato il massimo; 3) c3, di tipo *long int*, che deve contenere la somma di tutte le occorrenze calcolate dai processi. Quindi la comunicazione deve avvenire in particolare in questo modo: il figlio **P0**, dopo aver calcolato il numero di occorrenze *occ0* del carattere **Cx** trovate nel file **F1**, passa in avanti (cioè comunica) (*con una singola write*) al processo **P1** una struttura **S0**, con *c1* uguale a *occ0*, *c2* uguale a 0 e c3 uguale *occ0*; il figlio seguente **P1**, dopo aver calcolato il numero di occorrenze *occ1* del carattere **Cx** trovate nel 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 *occ0* e *occ1*, *c2* uguale all'indice del processo che ha calcolato il valore di *c1* e *c3* uguale alla somma di *occ0* e *occ1* e la passa (*con una singola write*) al figlio seguente **P2**, etc. fino al figlio **PN-1**, che si comporta in modo analogo, ma passa al **padre**. Quindi, al processo padre deve arrivare la struttura **SN-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 occorrenze e al nome del file in cui sono state trovate (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_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 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 **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!