

SISTEMI OPERATIVI E LAB.

(A.A. 13-14) – 09 GIUGNO 2014

IMPORTANTE:

- 1) Fare il login sui sistemi in modalità Linux usando il proprio **username** e **password**, attivare `syncexam.sh` e passare in modalità testuale.
- 2) I file prodotti devono essere collocati in un **sottodirettorio** (che deve essere nella directory `studente_XXX`) che deve essere creato e avere nome **ESAME09Giu14-1-01**. FARE ATTENZIONE AL NOME DEL DIRETTORIO, in particolare alle maiuscole e ai trattini indicati. Verrà penalizzata l'assenza del direttorio con il nome indicato e/o l'assenza dei file nel direttorio specificato, al momento della copia automatica del direttorio e dei file. **ALLA SCADENZA DEL TEMPO A DISPOSIZIONE VERRÀ INFATTI ATTIVATA UNA PROCEDURA AUTOMATICA DI COPIA, PER OGNI STUDENTE DEL TURNO, DEI FILE CONTENUTI NEL DIRETTORIO SPECIFICATO.**
- 3) Il tempo a disposizione per la prova è di **75 MINUTI** per lo svolgimento della sola parte C e di **120 MINUTI** per lo svolgimento di tutto il compito.
- 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.**

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 N parametri (con N maggiore o uguale a 2) che devono essere **nomi assoluti di direttori** che identificano N gerarchie (**G1, G2, ... GN**) all'interno del file system. Il comportamento atteso dal programma, dopo il controllo dei parametri, è organizzato in N fasi.

Il programma, per ognuna delle N fasi, deve richiedere all'utente un numero X intero strettamente positivo e maggiore di 255 e quindi deve esplorare la gerarchia **Gi** specificata - tramite un file comandi ricorsivo, **FCR.sh** - cercando tutti i file leggibili con un numero di linee strettamente maggiore di X . Al termine di tutte le N fasi, per ognuna delle gerarchie **Gi**, si deve riportare sullo standard output il nome della gerarchia **Gi** e il **nome assoluto** di ogni *file trovato*; quindi, si deve selezionare (sempre per ogni gerarchia **Gi**) il **primo** file trovato **PFi**. Infine, si deve invocare la parte in C passando come parametri gli N nomi assoluti dei file selezionati **PF0, PF1, ..., PFN-1**.

La parte in C accetta un numero *variabile* di parametri (**maggiore o uguale a 2, da controllare**) che rappresentano N nomi assoluti di file **F1...FN**.

Il processo padre deve generare N **processi figli (P0 ... PN-1)**: ogni processo figlio è associato al corrispondente file **Fi**. Ognuno di tali processi figli deve creare a sua volta un **processo nipote (PP0 ... PPN-1)**: ogni processo nipote **PPi** esegue concorrentemente calcolando la **lunghezza in linee** del file associato **Fi** usando in modo opportuno il comando `wc` di UNIX/Linux.

Ogni processo figlio **Pi** deve **convertire** in termini di **valore intero*** (**lunghezza**) quanto scritto in termini di caratteri sullo standard output dal comando `wc` eseguito dal processo nipote **PPi**; quindi ogni figlio **Pi** deve comunicare tale **lunghezza** al padre. Il padre ha il compito di ricevere, rispettando l'ordine dei file, il valore **lunghezza** inviato da ogni figlio **Pi**, calcolando via via la somma (come **long int**) di tutti i valori ricevuti e stampando alla fine la somma totale (sempre come **long int**) su standard output.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il valore di ritorno del proprio processo nipote **PPi** e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.

* Chiaramente il testo assume che il valore di **lunghezza** di ogni possibile file passato come parametro sia rappresentabile con un intero del linguaggio C; quindi la corrispondente soluzione può fare riferimento a questa ipotesi senza bisogno di specificarlo.