

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 12-13) – 17 LUGLIO 2013

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 **ESAME17Lug13_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.
- 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 un numero variabile **N+1** di parametri: il primo deve essere il **nome assoluto di un direttorio** 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 $N/2 \leq M \leq N$**): 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 maggiore o uguale a 1 (*da controllare*) che rappresentano i nomi relativi semplici di file **F1, F2, ... FM**.

Il processo padre deve generare **M processi figli (P0 ... PM-1)**: ognuno dei processi figli è associato ad uno dei file **F1, F2 ... FM**. Il padre deve quindi chiedere all'utente una serie di caratteri fino all'eof dello standard input: il padre comunica ad ogni figlio ogni carattere ricevuto dall'utente e ogni processo figlio **Pi** deve cercare tale carattere nel proprio file associato **Fi** e quindi deve comunicare il numero di occorrenze presenti (come *long int*) al processo padre. Il padre ha il compito di stampare su standard output, rispettando l'ordine dei file, il numero di occorrenze ricevute riportando anche sia il carattere che il file cui si riferiscono.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il valore 0 nel caso non sia stata trovata nessuna occorrenza di alcun carattere passato dal padre, altrimenti il valore 1.

Il padre, dopo che i figli sono terminati, deve stampare, su standard output, i PID di ogni figlio con il corrispondente valore ritornato.