

# SISTEMI OPERATIVI E LAB.

## (A.A. 10-11) – 14 SETTEMBRE 2011

### IMPORTANTE:

- 1) Fare il login sui sistemi in modalità Linux usando il proprio **username** e **password**.
- 2) I file prodotti devono essere collocati in un **sottodirettorio** della propria **HOME** directory che deve essere creato e avere nome **ESAME14Set11-2-1**. 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 **120 MINUTI** per lo svolgimento di tutto il compito e di **75 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.
- 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 **3 parametri**: il primo deve essere il nome assoluto di un direttorio che identifica una gerarchia (**G**) all'interno del file system, mentre il secondo e il terzo devono essere considerati numeri interi strettamente positivi **H** e **K**, con **H** strettamente minore di **K**. Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono un **numero** di file leggibili (**F0, F1, ... FN-1**) compreso fra **H** e **K**. Si riporti il nome assoluto di tali direttori sullo standard output. In ognuno di tali direttori trovati, si deve invocare la parte in C, passando come parametri i nomi dei file trovati (**F0, F1, ... FN-1**) che soddisfano la condizione precedente.

La parte in C accetta un numero variabile **N** di parametri che rappresentano nomi di file (**F0, F1, ... FN-1**).

Il processo padre deve generare **N processi figli Pi (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 PPi (PP0 ... PPN-1)** associato sempre al corrispondente file **Fi**. Ogni processo figlio **Pi** e ogni nipote **PPi** esegue concorrentemente andando a cercare nel file associato **Fi** tutte le occorrenze dei caratteri alfabetici **maiuscolo** per il figlio e tutte le occorrenze dei caratteri numerici per il nipote. Ognuno dei processi figlio e nipote deve operare una modifica del file **Fi**: in specifico, ogni *figlio* deve trasformare ogni carattere alfabetico maiuscolo nel corrispondente carattere alfabetico minuscolo, mentre ogni *nipote* deve trasformare ogni carattere numerico nel carattere spazio. Una volta terminate le trasformazioni, sia i processi figli **Pi** che i processi nipoti **PPi** devono comunicare al padre il numero (in termini di *long int*) di trasformazioni effettuate. Il padre ha il compito di stampare su standard output, rispettando l'ordine dei file, il numero di trasformazioni ricevute da ogni figlio **Pi** e da ogni nipote **PPi**, riportando opportuni commenti esplicativi: ad esempio “nel file *F0* sono stati trasformati *tot1* caratteri alfabetici maiuscolo in caratteri alfabetici minuscolo e *tot2* caratteri numerici nel carattere spazio”.

Al termine, ogni processo nipote **PPi** deve ritornare al figlio **Pi** un opportuno codice ed analogamente ogni processo figlio **Pi** deve ritornare al padre un opportuno codice; il codice che ogni nipote **PPi** e ogni figlio **Pi** deve ritornare è:

- a) **0** se il numero di trasformazioni attuate è minore di 256;
- b) **1** se il numero di trasformazioni attuate è maggiore o uguale a 256, ma minore di 512;
- c) **2** se il numero di trasformazioni attuate è maggiore o uguale a 512, ma minore di 768;
- d) etc.

Sia ogni figlio **Pi** e sia il padre devono stampare su standard output il PID di ogni nipote/figlio e il valore ritornato.