

# SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 09-10) – 10 SETTEMBRE 2010

## 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 **ESAME10Set10-1-1**. 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, il secondo deve essere considerato un numero intero strettamente positivo **dispari (K1)** e il terzo deve essere considerato un numero intero strettamente positivo **pari (K2)**. Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono *esattamente* due file che abbiano dimensione in byte rispettivamente uguale a **K1** e uguale a **K2**: in altre parole, si devono cercare tutti i direttori che contengono uno e un solo file di dimensione **K1** e uno e un solo file di dimensione **K2**. Si riporti il nome assoluto di tali direttori sullo standard output. Al termine dell'intera esplorazione ricorsiva di **G**, si deve invocare la parte in C, passando come parametri i nomi assoluti dei file trovati (**F0, F1, ... FN-1**).

La parte in C accetta un numero variabile **N** di parametri che rappresentano nomi assoluti di file (**F0, F1, ... FN-1**). Il processo padre deve generare **N processi figli (P0 ... PN-1)** ognuno dei quali è associato ad uno dei file **Fi**: i processi figli devono essere distinti, in base al loro numero d'ordine, in *processi pari* e in *processi dispari*. Ogni processo figlio deve leggere i caratteri del proprio file **Fi** e deve effettuare **due conteggi**, usando due variabili, entrambe di tipo *long int*:

- a) ogni processo figlio **pari** deve contare quanti caratteri di posizione pari nel file associato hanno codice ASCII pari (*primo conteggio*) e quanti hanno codice ASCII dispari (*secondo conteggio*);
- b) ogni processo figlio **dispari** deve contare quanti caratteri di posizione dispari nel file associato hanno codice ASCII dispari (*primo conteggio*) e quanti hanno codice ASCII pari (*secondo conteggio*);

Completati i due conteggi, ogni processo figlio deve comunicare al processo padre una **struttura** che deve contenere due campi, *c1* e *c2*, con *c1* uguale al primo conteggio ottenuto e con *c2* uguale al secondo conteggio ottenuto. Il processo padre deve ricevere, **rispettando l'ordine dei file**, *prima* tutte le strutture inviate dai figli pari e *poi* tutte le strutture inviate dai figli dispari e deve stampare su standard output sia il numero d'ordine dei processi figli, che i due campi *c1* e *c2* delle strutture ricevute.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il **valore 0** (zero) se il primo conteggio è maggiore del secondo, altrimenti il **valore 1** (uno); il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.