

SISTEMI OPERATIVI e

LABORATORIO DI SISTEMI OPERATIVI

(A.A. 03-04) – 20 SETTEMBRE 2004

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 **ESAME20Set-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.
- 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 devono essere delle stringhe **S1, S2, ... SN** ognuna di 2 caratteri.

Il programma deve cercare nella gerarchia **G** specificata tutti i file che contengono (nel contenuto) *almeno una* delle stringhe **Si**. Si riporti il nome assoluto di tali file sullo standard output. Per ogni file trovato, si deve invocare la parte in C, passando come parametri il nome del file trovato e le stringhe **Si**.

La parte in C accetta un numero variabile **K+1** di parametri che rappresentano un nome di file **F** e **K** stringhe di 2 caratteri ognuna. Il processo padre deve generare **K processi figli (P0 ... PK-1)**: ogni processo figlio è associato ad una stringa **Sj**. Ognuno di tali processi figli deve creare a sua volta un processo figlio (**PP0 ... PPK-1**): ogni processo figlio **Pj/nipote PPj** esegue concorrentemente. Ogni **coppia** di processi figlio **Pj/nipote PPj** deve cercare nel file **F** i caratteri costituenti la

stringa **Sj** associata: in particolare, il **figlio Pj** deve cercare il primo carattere di Sj, mentre il **nipote PPj** il secondo carattere. Ogni figlio **Pj**/nipote **PPj** alla prima occorrenza del carattere cercato deve comunicare al padre una struttura che contiene **esclusivamente** la posizione nel file dell'occorrenza trovata e il valore 0 o 1 a seconda che il mittente sia il figlio o il nipote.

Per ogni coppia e quindi per ogni stringa, il padre ha il compito di stampare su standard output **rispettando l'ordine di creazione dei figli** se la stringa corrispondente è stata trovata o meno, usando in modo opportuno le informazioni ricevute da ogni coppia (e cioè se la prima occorrenza sia del primo che del secondo carattere della stringa sono state trovate in posizioni adiacenti).

Al termine, ogni processo figlio **Pj** deve ritornare al padre il numero di occorrenze trovate del primo carattere della stringa **Sj** associata (supposto minore di 255) e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.