

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 03-04) – 2 APRILE 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 **ESAME2Apr-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 tre parametri: il primo deve essere il **nome assoluto di un direttorio** che identifica una gerarchia (**G**) all'interno del file system; il secondo parametro deve essere un numero **M** minore di 6; mentre il terzo parametro deve essere considerato un singolo carattere **C**. Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono un numero di file esattamente uguale a 6 aventi al loro interno almeno una occorrenza del carattere **C**. Si riporti il nome assoluto di tali direttori sullo standard output. In ogni direttorio trovato, si deve invocare la parte in C, passando come parametri i **nomi dei primi M file trovati e il carattere C**.

La parte in C accetta un numero variabile di parametri che rappresentano **M** nomi di file **F1..FM** e un carattere **C**. Il processo padre deve generare **M processi figli (P0 ... PM-1)**: ogni processo figlio è associato ad uno dei file **Fj**. Ognuno di tali processi figli esegue concorrentemente, legge e calcola il numero di occorrenze **Occj** del carattere **C** nel file associato. Ogni figlio **Pj** deve comunicare una informazione al figlio seguente **Pj+1**, a parte l'ultimo figlio che la comunica al padre: l'informazione da comunicare è costituita da un **PID** e un numero **Occ**. In particolare il primo processo figlio **P0** dopo aver calcolato il numero di occorrenze **Occ0** nel proprio file, comunica il proprio **PID** e **Occ0** al processo **P1**; il processo **P1**, dopo aver calcolato **Occ1**, riceve da **P0** l'informazione comunicata e passa al processo **P2** il proprio **PID** e **Occ1** se il valore di **Occ** ricevuto è maggiore di quello calcolato oppure l'informazione ricevuta da **P0** e così via fino a che l'ultimo processo **PM-1**, dopo aver calcolato **OccM-1**, riceve da **PM-2** l'informazione comunicata e passa al processo **padre** il proprio **PID** e **OccM-1** se il valore di **Occ** ricevuto è maggiore di quello calcolato oppure l'informazione ricevuta da **PM-2**. Il padre ha il compito di stampare su standard output l'informazione ricevuta dal processo **PM-1**. Al termine, ogni processo figlio **Pj** deve ritornare al padre il proprio **Occj** e il padre deve stampare su standard output i **PID** e i valori ritornati dai figli.