

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 07-08) – 17 GIUGNO 2008

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 **ESAME17Giu08-1-1**. 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 2 **parametri**: il primo deve essere il **nome assoluto di un direttorio** che identifica una gerarchia (**G**) all'interno del file system, mentre il secondo parametro deve essere considerato un numero intero strettamente positivo (**K**).

Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono file che abbiano numero di linee uguale a **K**. Si riporti il nome assoluto di tali direttori sullo standard output. In ognuno di tali direttori trovati, per ogni file F_i trovato (**F1, F2, ... FN**) che soddisfa la condizione precedente si deve produrre un **ordinamento delle linee** secondo il normale ordine alfabetico ridirigendo il risultato su un file di nome **$F_i.sort$** . Quindi, sempre in ogni direttorio trovato, si deve invocare la parte in C, passando come parametri i nomi dei file generati (**F1.sort, F2.sort, ... FN.sort**).

La parte in C accetta un numero variabile **N** di parametri che rappresentano **N** nomi di file (**FS1, FS2, ... FSN**): tutti i file hanno lo stesso numero di linee (questo viene garantito dalla parte shell e NON deve essere controllato). Il processo padre deve creare un file di nome **MERGE** e quindi generare **N processi figli (P0 ... PN-1)** ognuno dei quali è associato ad uno dei file **FSi**. Ogni processo figlio deve leggere le singole linee del file associato **FSi sempre** fino alla fine del file. I processi figli devono attenersi a questo **schema di comunicazione**: ogni figlio P_i comunica al padre, **per ogni linea** del file **FSi** associato, via via **due informazioni separate**:

- ? la lunghezza in byte della linea corrente (compreso il terminatore di linea);
- ? i byte che costituiscono la linea corrente (compreso il terminatore di linea).

Il padre deve ricevere, rispettando l'ordine dei figli, via via le informazioni inviate da ogni figlio e deve scrivere ogni linea ricevuta sul file **MERGE** (si usi opportunamente la prima delle due informazioni via via inviate).

Al termine, ogni processo figlio P_i deve ritornare al padre il numero di linee inviate con successo al padre (*supposto minore di 255*) e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.