

# SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 08-09) – 3 APRILE 2009

## 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 **ESAME3Apr09-1-1**. 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 così come, per la sola valutazione di Laboratorio di Sistemi Operativi, la non compilabilità del sorgente C e l'assenza di un Makefile idoneo al progetto.**
- 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 un **Bourne Shell** e una parte in **C**.

La parte in Shell deve prevedere due parametri: il primo parametro 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 (compresa la radice) che contengono un numero di file (solo file e non direttori) compreso fra **1** e **K (estremi inclusi)**: si riporti il nome assoluto dell'ultimo file trovato (**Fi**) sullo standard output. Al termine dell'intera esplorazione ricorsiva di G, si deve invocare la parte C passando come parametri tutti i nomi assoluti dei file **Fi** trovati.

La parte in C accetta un numero variabile **N** di parametri che rappresentano **N** nomi di file (**F1, F2, ... FN**). Il processo padre deve creare un file di nome **NUMERICO** e deve generare **N processi figli (P0 ... PN-1)** ognuno dei quali è associato ad uno dei file **Fi**. Ogni processo figlio esegue in modo concorrente e deve leggere le singole linee del file associato **Fi sempre** fino alla fine. I processi figli devono attenersi a questo schema di comunicazione con il padre: dopo ogni lettura della linea corrente, se il primo carattere della linea è un carattere **numerico**, il figlio invia al padre la linea corrente letta. Il padre deve ricevere le informazioni dai figli via via, fino a che i figli non hanno terminato di leggere le linee dei file associati: il processo padre deve scrivere sul file **NUMERICO** le linee dei file **F1, F2 ... FN** comunicate dai figli mantenendo l'ordine dei file (prima le linee di **F1**, poi di **F2**, ecc.). Al termine, i processi figli **P0 ... PN-1** devono ritornare al padre il numero intero corrispondente al numero di linee comunicate al padre (*supposto minore di 255*).

Il padre, dopo che i figli sono terminati, deve stampare su standard output i **PID** di ogni figlio con il corrispondente valore ritornato.