

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 09-10) – 11 FEBBRAIO 2011

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 **ESAME11Feb11-2-1**. 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 considerato un numero intero positivo pari (**K**), mentre il terzo parametro deve essere considerato un numero intero positivo dispari (**H**). Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono almeno **un file Fi** la cui lunghezza in byte sia uguale a **K** e sia anche un multiplo intero di **H**. Si riporti il nome assoluto di tali direttori sullo standard output. Al termine dell'intera esplorazione ricorsiva di G, si deve invocare la parte C passando come parametri **H** e tutti i **nomi assoluti** dei file **Fi** trovati.

La parte in C accetta un numero variabile di parametri **N+1**: il primo parametro rappresenta un numero intero positivo dispari (da verificare) **H**, mentre gli altri **N** parametri rappresentano nomi assoluti di file **F0...FN-1**: si può ipotizzare che la lunghezza di tutti i file sia uguale, pari e multiplo intero **H**(senza verificarlo). Il processo padre deve generare **2Nprocessi figli (P0 ... P2N-1)**; tali processi figli costituiscono **N coppie di processi**: ogni **coppia Ci** è composta dal processo **Pi** (primo processo della coppia) e dal processo **Pi+N** (secondo processo della coppia), con **i** variabile da 0 a **N-1**. Ogni coppia di processi figli **Ci** è associata ad uno dei file **Fi***. Il primo processo della coppia deve creare un file il cui nome (**FCreato**) risulti dalla concatenazione del nome del file associato alla coppia con la stringa **“.mescolato”** (ad esempio se **Fi** è **/tmp/pippo.txt** il file **FCreato** si deve chiamare **/tmp/pippo.txt.mescolato**). Tutte le **coppie** di processi figli eseguono concorrentemente leggendo il proprio file associato: in particolare, il primo processo della coppia deve leggere la prima metà del file associato, mentre il secondo processo la seconda metà del file; inoltre, per entrambi i processi della coppia la lettura deve avvenire a blocchi di dati di grandezza uguale a **H** byte. Il primo processo di ogni coppia, dopo la lettura di ogni blocco di dati **B1** della sua prima metà del file, lo scrive (con un'unica write!) sul file **FCreato**; quindi deve ricevere (con un'unica read!) dal secondo processo della coppia il suo corrispondente blocco di dati **B2** e quindi deve scriverlo (sempre con un'unica write!) sul file **FCreato**; viceversa, il secondo processo di ogni coppia, dopo la lettura di ogni blocco di dati **B2** della sua seconda metà del file, lo comunica (con un'unica write!) al primo processo della coppia. Al termine, ogni processo di ogni coppia deve ritornare al padre il numero di blocchi letti dalla propria metà del file. Il padre, dopo che i figli sono terminati, deve stampare su standard output i PID di ogni figlio con il corrispondente valore ritornato.

* Se **N** è 3 (**i** varia da 0 a 2), le coppie di processi e i file associati sono **P0-P3** con **F0**, **P1-P4** con **F1** e **P2-P5** con **F2**.