

SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 12-13) – 11 SETTEMBRE 2013

IMPORTANTE:

- 1) Fare il login sui sistemi in modalità Linux usando il proprio **username** e **password**, attivare syncexam.sh e passare in modalità testuale.
- 2) I file prodotti devono essere collocati in un **sottodirettorio** (che deve essere nella directory studente_XXX) che deve essere creato e avere nome **ESAME11Set13_1_01**. 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 **75 MINUTI** per lo svolgimento della sola parte C e 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 **due** parametri: il primo deve essere il nome assoluto di un direttorio che identifica una gerarchia (**G**) all'interno del file system e il secondo deve essere considerato un numero intero strettamente positivo (**K**). Il programma deve cercare nella gerarchia **G** specificata tutti i direttori che contengono almeno **un** file la cui lunghezza in byte sia esattamente uguale a **K**: si riporti il nome assoluto di tali direttori sullo standard output. Al termine dell'intera esplorazione ricorsiva di G, si deve verificare che il numero globale di file trovati sia **pari** e, solo in tale caso, si deve invocare la parte in C passando come parametri i nomi assoluti dei file trovati **F0, F1, ... FN-1 (con N appunto pari)**.

La parte in C accetta un numero variabile pari **N** di parametri maggiore o uguale a 2 (*da controllare che N sia pari e sia ≥ 2*) che rappresentano i nomi assoluti di file **F0, F1, ... FN-1 (tutti con uguale lunghezza, che non deve essere controllata)**. Il processo padre deve generare **N/2 processi figli (P0 ... PN/2-1)** e ognuno dei processi figli deve generare un processo nipote (**PP0 ... PPN/2-1**): i processi figli **Pj** sono associati ai file **Fj** mentre i processi nipoti **PPj** ai file **Fj+N/2 (con j che, in entrambi i casi, varia da 0 a N/2)**. Ogni processo figlio **Pj** deve, prima di creare il proprio nipote, creare un file **FOut** il cui nome deve risultare dalla concatenazione della stringa "*merge*" e della stringa corrispondente a **j** (numero d'ordine di creazione del processo figlio). Una volta creato il processo nipote, ogni figlio e ogni nipote eseguono concorrentemente; in particolare, ognuno dei due 'tipi' di processi deve leggere, dal suo file associato, un carattere alla volta e quindi lo deve scrivere sul file **FOut**: la scrittura deve avvenire in modo strettamente alternato, iniziando dal figlio **Pj**. In altre parole, il figlio **Pj** legge il primo carattere dal file **Fj** e lo scrive sul file **FOut** e quindi deve comunicare l'avvenuta scrittura sul file al nipote **PPj**, quindi il processo **PPj** che ha concorrentemente letto il primo carattere dal file **Fj+N/2** lo può scrivere sul file **FOut** e può comunicare al figlio l'avvenuta scrittura; tale schema di comunicazione/sincronizzazione¹ deve continuare per tutti i caratteri dei due file associati.

Al termine, ogni processo nipote **PPj** deve ritornare al figlio il valore dell'ultimo carattere scritto nel file **FOut** e, a sua volta, il processo figlio lo deve ritornare al padre. Il padre, dopo che i figli sono terminati, deve stampare, su standard output, i PID di ogni figlio con il corrispondente valore ritornato.

¹ Se si vuole per la sincronizzazione si possono usare i segnali.