

SISTEMI OPERATIVI e
LABORATORIO DI SISTEMI OPERATIVI
(A.A. 11-12) – 13 Febbraio 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 **ESAME13Feb13-1-1**. 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 mentre il secondo e il terzo devono essere considerati singoli caratteri (**C1** e **C2**). Il programma deve cercare nella gerarchia **G** specificata (compresa la radice) tutti i direttori che contengono almeno un file che ha nel suo contenuto istanze sia del carattere **C1** che del carattere **C2**. Si riporti il nome assoluto di tali direttori sullo standard output. In ognuno di tali direttori trovati, si deve invocare la parte in C, passando come parametri i nomi degli **N** file trovati (**F0, F1, ... FN-1**) che soddisfano la condizione precedente e i caratteri **C1** e **C2**.

La parte in C accetta un numero variabile **N+2** di parametri (maggiore o uguale a 4) che rappresentano i primi **N** nomi di file (**F0, F1, ... FN-1**), mentre gli ultimi due rappresentano singoli caratteri (**C1** e **C2**) (da controllare). Il processo padre deve generare con **2*N** processi figli concorrenti (**P0 ... PN-1** e **PN ... P2N-1**), ognuno dei quali è associato ad uno dei file **Fi**: **P0** e **PN** sono associati al file **F0, P1** e **PN+1** al file **F1** e così via fino a **PN-1** e **P2N-1** che sono associati al file **FN-1**. Ogni coppia di processi figli **Pi** e **Pi+N** (con *i* da **0** a **N-1**) deve leggere concorrentemente i caratteri del file associato **Fi**, cercando *indipendentemente* le occorrenze del carattere **Cx** (**C1** e **C2**): in particolare, i primi **N** processi devono cercare le occorrenze del carattere **C1**, mentre gli altri **N** processi le occorrenze del carattere **C2**. I processi figli devono attenersi ad uno schema di comunicazione a pipeline ad **N** compreso il processo padre: in particolare, il figlio **P0** comunica con il figlio **P1** che comunica con il figlio **P2** etc. fino al figlio **PN-1** che comunica con il padre e la stessa cosa per gli altri **N** figli e cioè il figlio **PN** comunica con il figlio **PN+1** che comunica con il figlio **PN+2** etc. fino al figlio **P2N-1** che comunica con il padre. Le strutture dati che i processi figli devono comunicare nelle due pipeline devono avere **3 campi**, *indice*, *occmín* e *occtotale*: *indice* deve essere l'indice del processo che ha trovato nel suo file associato il minimo numero di occorrenze di **Cx**, *occmín* il valore di tale minimo e *occtotale* il conteggio globale ottenuto fino a quel momento. Il padre ha il compito di stampare su standard output tutti i campi delle due strutture ricevute dalle due pipeline (prima quella dei primi **N** figli e poi l'altra) aggiungendo anche l'indicazione del carattere **Cx** cui si riferiscono.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il carattere **Cx** e il padre deve stampare su standard output il **PID** di ogni figlio e il valore ritornato.