

# SISTEMI OPERATIVI e LABORATORIO DI SISTEMI OPERATIVI (A.A. 13-14) – 9 FEBBRAIO 2015

## 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 **ESAME09Feb15\_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, mentre 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 leggibile con lunghezza in linee pari a **K**: si riporti il nome assoluto di tali direttori sullo standard output. Al termine dell'intera esplorazione ricorsiva di G, per ogni file trovato **Fi** si deve richiedere all'utente un numero **Xi** intero strettamente positivo e minore o uguale a **K**: quindi si deve invocare la parte in C passando come parametri i nomi assoluti dei file trovati intervallati dal numero corrispondente chiesto all'utente (perciò i parametri saranno: **F0, X0, F1, X1, ... FN-1, XN-1**).

La parte in C accetta un numero variabile pari **2N** di parametri maggiore o uguale a 2 (*da controllare*) che rappresentano **N** nomi assoluti di file **F0, F1, ... FN-1** intervallati da numeri interi strettamente positivi **X0, X1, ... XN-1** (da controllare). Il processo padre deve generare **N** processi figli: i processi figli **Pi** sono associati ai file **Fi** e al numero **Xi**. Ognuno di tali figli deve creare a sua volta un processo nipote **PPi**: ogni processo nipote **PPi** esegue concorrentemente inviando al figlio le prime **Xi** linee del file **Fi** usando in modo opportuno il comando **head** di UNIX/Linux.

Ogni processo figlio **Pi** deve ricevere tutte le prime **Xi** linee dal suo processo nipote **PPi** e deve inviare al processo padre l'ultima linea inviata dal nipote **PPi**.

Il padre ha il compito di ricevere, rispettando l'ordine dei file, la linea inviata da ognuno dei figli **Pi** che deve essere riportata sullo standard output insieme con l'indicazione del numero d'ordine del figlio, del numero di linea del file e il nome del file da cui è stata letta tale linea.

Al termine, ogni processo figlio **Pi** deve ritornare al padre il valore di ritorno del proprio processo nipote **PPi** (che a sua volta deve tornare il valore di ritorno del comando **head** oppure -1 in caso di errore nell'invocazione del comando **head**) e il padre deve stampare su standard output il PID di ogni figlio e il valore ritornato.