SISTEMI OPERATIVI e
LABORATORIO DI SISTEMI OPERATIVI
(A.A. 09-10) – 4 GIUGNO 2010

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 **ESAME04Giu10-1-1.** 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 **75 minuti**.
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**.

Parte in Linguaggio C

La parte in C accetta un numero variabile **N** di parametri (maggiore o uguale a 1) che rappresentano **N** nomi di file (**F1, F2, ... FN**). Il processo padre deve generare **N+1 processi figli**: il primo figlio generato (**P0**) verrà indicato con il nome di **MASTER**, mentre gli altri **N** figli verranno indicati con il nome di **SLAVE** (**P1** … **PN**): ognuno dei processi SLAVE è associato ad uno dei file **Fi**. Ogni processo figlio esegue in modo concorrente: il MASTER deve per prima cosa creare un file di nome **UGUALE,** quindi si deve mettere in attesa delle informazioni comunicate dagli SLAVE che devono essere scritte sul file creato, secondo le indicazioni seguenti (vedi *****); ogni SLAVE deve leggere le singole linee del file associato **Fi sempre** fino alla fine. Gli SLAVE devono attenersi a questo schema di comunicazione con il MASTER: dopo ogni lettura di una linea (detta linea corrente), se il primo carattere della linea corrente è uguale all'ultimo carattere della linea (terminatore escluso), lo SLAVE comunica con un unico invio (una sola write!) al MASTER la linea corrente letta con il terminatore (*si supponga che la lunghezza massima della linea, compreso il terminatore, sia di 120 caratteri*). Il MASTER deve ricevere ogni linea con un'unica ricezione (una sola read!) via via, fino a che gli SLAVE non hanno terminato di leggere le linee dei file associati F1, F2, ... FN, e deve scrivere sul file **UGUALE** le linee comunicate dagli SLAVE seguendo questo ordine (*****): la prima linea inviata dal primo SLAVE, poi la prima linea inviata dal secondo SLAVE, etc. fino alla prima linea inviata dall'ultimo SLAVE, quindi deve re-iniziare il giro e cioè la seconda linea inviata dal primo SLAVE, poi la seconda linea inviata dal secondo SLAVE, etc. fino alla seconda linea inviata dall'ultimo SLAVE, e così via fino a che non sono state scritte sul file tutte le linee inviate da tutti i figli. Al termine, il figlio MASTER (**P0)** deve ritornare al padre il numero intero corrispondente al numero di linee scritte sul file **UGUALE**, mentre i figli SLAVE (**P1** … **PN)** devono ritornare al padre il numero intero corrispondente al numero di linee comunicate al MASTER (*si suppongano tutti i valori di ritorno minori 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.