Corso di Laurea in Ingegneria Elettronica-Informatica-Telecomunicazioni
a.a. 2002-2003

CORSO DI SISTEMI OPERATIVI A
Prova del 19/6/2003

MATR. . . . . Cognome . . . . . Nome . . . . . .

Username . . . . . .

NOTE
Il presente foglio va immediatamente compilato con le proprie generalità e matricola.
Esso deve essere restituito al termine della prova. In caso di mancata restituzione, la prova dello studente non verrà presa in considerazione per la correzione.

IMPORTANTE
Tutti i file sorgenti prodotti dallo studente per l'esame devono essere memorizzati in un direttorio denominato **soa-190603-x** nella propria home, dove **x** rappresenta il carattere **i** per gli Informatici, **t** per i Telecomunicazionisti, **e** per gli Elettronici.
Soluzioni contenute in altri direttori non verranno prese in considerazione per la correzione.

Prova UNIX-A1

Si realizzi in ambiente Unix/C la seguente interazione tra processi:

- il sistema consiste di due processi: un processo P_{padre} che crea un processo P_{figlio} ;
- i due processi utilizzano una pipe p_{pf} per la comunicazione ;
- P_{padre} richiede all'utente il nome di comando e lo invia come messaggio al processo P_{figlio} ;
- il processo P_{figlio} deve mettere in esecuzione il comando ricevuto.

Prova UNIX-A2

Si realizzi in ambiente Unix/C la seguente interazione tra processi:

- il sistema consiste di due tipi di processi: un processo *server* remoto P_s e i processi clienti P_{ci};
- per la comunicazione tra P_s e i processi clienti P_{ci} vengono utilizzate socket `Stream`;
- il server P_s offre un servizio concorrente (un figlio per ogni connessione) di esecuzione comandi alla porta N (dove N è ottenuto dalle 5 cifre meno significative della vostra matricola);
- ogni processo cliente P_{ci} invia al server un valore intero casuale T(con $10 < T < 30$);

- i processi figli del server P_s, ricevuto il valore T , devono sospendersi per T secondi (utilizzando la primitiva nanosleep) e ritornare al cliente l'intervallo residuo di attesa (eventualmente pari a zero);
- i processi figli del server P_s devono aver bloccato il segnale `SIGUSR1` e predisporre un gestore per il segnale `SIGUSR2` che visualizzi un messaggio arbitrario.

Devono essere utilizzate le primitive per la gestione affidabile dei segnali.