

Sistemi Operativi L-A
Compito di Venerdì 20 Giugno 2008
CdS in Ingegneria Informatica - Prof. Paolo Bellavista

Compito A - Parte di Programmazione di Sistema (16 punti)

Si scriva un programma C che, utilizzando le System Call del sistema operativo UNIX, abbia un'interfaccia del tipo:

`codaAllePoste fileIn fileOut`

dove **fileIn** è un nome assoluto di file esistente nel file system mentre **fileOut** è il nome relativo del file che deve essere creato nel direttorio corrente dall'esecuzione del programma concorrente. Dopo aver effettuato gli opportuni controlli sui parametri di invocazione, il processo iniziale **P0** (impiegato delle poste) deve generare altri 2 processi **P1** (cliente1) e **P2** (cliente2), entrambi figli di **P0**.

Il file **fileIn** contiene un numero **N** di righe (non noto a priori); ogni riga descrive un'operazione da effettuare, con in testa il carattere 'n' in caso di operazione normale, 'p' in caso di operazione prioritaria. Ad esempio, alcune righe di **fileIn** potrebbero essere:

```
n versamento cc 1234 importo 100 \n
p spedizione raccomandata destinatario Pippo \n
n acquisto 25 francobolli posta prioritaria \n
```

I processi cliente devono dedicarsi a leggere concorrentemente il file delle operazioni e a comunicare una operazione alla volta all'impiegato, dedicandosi **P1** alle prime **N/2** righe del file, **P2** alle rimanenti. Nel caso venga incontrata una operazione prioritaria, questa deve "scavalcare" le operazioni normali ed essere ricevuta con precedenza dall'impiegato. Nel caso di operazione normale, l'impiegato dovrà semplicemente scrivere il contenuto della stringa operazione sul file **fileOut**; nel caso di operazione prioritaria, la scrittura su **fileOut** dovrà essere effettuata sia dall'impiegato che dal cliente corrispondente, in modo tale che il file di uscita contenga due righe identiche, che devono essere consecutive.

In ogni istante, deve essere possibile per l'utente forzare la modifica del comportamento dell'ufficio postale premendo la combinazione di tasti <CTRL-C>. In particolare, se <CTRL-C> viene premuto una sola volta, allora è **P1** che si dovrà occupare di trasmettere tutte le operazioni di **fileIn** ancora non lette e trasmesse, mentre **P2** sarà immediatamente terminato. Invece, nel caso di seconda pressione di <CTRL-C>, **P2** si occuperà di completare la trasmissione delle operazioni ancora non lette e trasmesse, mentre **P1** sarà immediatamente terminato.

Si facciano le ipotesi semplificative desiderate in termini di modello affidabile dei segnali e di innestamento dell'esecuzione dei gestori associati. Come sempre, si facciano le scelte di sincronizzazione dei processi ritenute più opportune, cercando di sequenzializzare il meno possibile le varie operazioni richieste.