

**Sistemi Operativi L-A**  
**Compito di Venerdì 18 luglio 2008**  
CdS in Ingegneria Informatica - Prof. Paolo Bellavista

**Compito B - 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:

`unicaPipeB num fileIn fileOut`

dove **num** è un numero naturale, **fileIn** è un nome relativo di file esistente nel file system e **fileOut** è il nome assoluto 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** deve generare un processo figlio **P1**, il quale a sua volta dovrà generare due processi nipote **P2** e **P3**.

**fileIn** contiene una sequenza di caratteri organizzati in righe (numero di righe non noto a priori, lunghezza massima di una riga pari a 80 caratteri), con ciascuna riga che comincia con il carattere '2' oppure con il carattere '3'. Ciascuno dei due processi nipote deve leggere concorrentemente l'intero **fileIn**, **P2** processando le sole righe che cominciano con '2', **P3** occupandosi delle righe che cominciano per '3'. In particolare, per ciascuna riga di pertinenza, ciascuno dei nipoti dovrà inviare un messaggio (**P2** a **P0**, **P3** a **P1**) contenente "sì" se il numero di caratteri minuscoli trovati nella riga ha superato **num**, "no" altrimenti. I processi nipote devono comunicare con **P0** e **P1** facendo uso di una unica pipe condivisa, utilizzata quindi da **P2** e **P3** per scrivere e da **P0** e **P1** per leggere messaggi. I processi lettori devono semplicemente scrivere il contenuto dei messaggi letti su **fileOut**.

In ogni istante, deve essere possibile per l'utente forzare la modifica del comportamento del programma concorrente premendo la combinazione di tasti <CTRL-C>. In particolare, se <CTRL-C> viene premuto una sola volta, allora **P1** e **P3** dovranno essere terminati, mentre **P2** dovrà occuparsi di leggere tutta la parte del file che gli era rimasta, indipendentemente dal carattere iniziale delle righe (non costituisce problema il fatto che alcune righe possano non essere lette da alcun processo), sempre trasmettendo i messaggi a **P0**. Nel caso di seconda pressione di <CTRL-C> da parte dell'utente, tutti i processi dovranno invece essere terminati.

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.