

Sistemi Operativi – a.a. 2015/2016

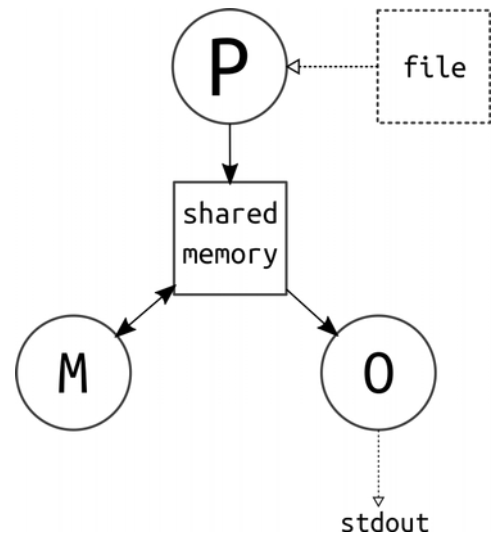
prova di laboratorio
– 27 aprile 2016 –

Creare un programma `numbers-mod.c` in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

`numbers-mod <input file> <modulus number>`

Il programma dovrà leggere dal file di input specificato una serie di numeri (uno per riga) e dovrà dare in output gli stessi una volta applicata la funzione $f(x) = x \bmod n$, dove n è il modulo specificato sulla riga di comando.

Il programma al suo avvio creerà due processi figli `Mod` e `Out`. I tre processi comunicheranno unicamente tramite un segmento di memoria condiviso e dei semafori che il padre dovrà creare e poi distruggere. Il segmento dovrà essere grande a sufficienza e organizzato per ospitare esattamente 10 record del tipo `[numero, tipo]`. I semafori dovranno essere in numero ottimale a gestire il coordinamento dei processi coinvolti ed essere utilizzati in modo opportuno.



I ruoli dei tre processi saranno i seguenti:

- il processo padre `P` dovrà leggere il file specificato inserendo, per ogni numero x letto, il record `[x, 0]` nel buffer condiviso;
- il processo figlio `Mod` dovrà estrarre dal buffer condiviso, non appena disponibili, record del tipo `[x, 0]` e reinserirne altrettanti del tipo `[f(x), 1]`;
- il processo figlio `Out` dovrà estrarre dal buffer condiviso, non appena disponibili, record del tipo `[y, 1]` e dare in output il numero y .

Il programma dovrà funzionare con file con serie arbitrariamente grandi di numeri; in particolare dovrà funzionare con i 2 file creati con i seguenti comandi shell:

- `seq 100 > list-1.txt`
- `for i in `seq 500`; do echo "$RANDOM" >> list-2.txt; done`

Alla fine i processi dovranno terminare spontaneamente, rilasciando qualunque struttura dati persistente.

Suggerimenti:

- il padre `P` dovrebbe bloccarsi se il buffer dovesse risultare pieno;
- anche il figlio `Mod` dovrebbe bloccarsi se, nel tentativo di inserire il suo record, il buffer dovesse risultare pieno;
- trovare un opportuno meccanismo di segnalazione per la fine dei dati affinché i consumatori non si blocchino all'infinito.

Tempo: 2 ore

Ricordarsi di inserire i propri dati (nome, cognome, matricola) nei commenti preliminari del codice sorgente.

Verrà valutata anche l'efficienza computazionale delle soluzioni algoritmiche utilizzate.

Per inviare il proprio elaborato sul server è necessario utilizzare il comando **exam-box-sync**. Verrà richiesta la password associata al proprio account e verrà data una conferma all'avvenuto caricamento. E' possibile, e fortemente consigliato, inviare il proprio elaborato più volte e periodicamente come copia di riserva (l'ambiente di lavoro degli esami risiede in memoria RAM e è pertanto di tipo non-persistente).