

### Esercizio (somma di matrici)

Realizzare un programma C utilizzando le primitive di sistema del sistema operativo UNIX, che preveda un'interfaccia del tipo:

*esame N M*

dove:

N ed M rappresentano numeri interi

Il programma ha come obiettivo quello di effettuare la somma di due matrici di dimensione (N\*M).

In particolare, si vuole realizzare il calcolo in modo parallelo. A questo scopo, il processo iniziale P0, dopo aver letto dallo standard input tutti gli elementi delle matrici da sommare, affiderà il calcolo relativo ad ogni colonna ad un processo figlio distinto.

Al termine della computazione, i risultati verranno raccolti da P0 e visualizzati sullo standard output.

```
#include <stdio.h>
#include <errno.h>
#include <fcntl.h>
#define MAX_RIGA 10
#define MAX_COL 10
```

```
extern int errno;
```

```
main(argc, argv)
int argc;
char **argv;
{ int mat_a[MAX_RIGA][MAX_COL], ris[MAX_COL+1];
  int mat_b[MAX_RIGA][MAX_COL], dim, i,j,R,C, pid;
  int p[2], nread, stato, uscita;

  if (argc != 3) {
    printf("Errore. Uso: %s arg1 arg2\n",argv[0]);
    exit(1);}
  R = atoi(argv[1]);
  C = atoi(argv[2]);
  for(i=0;i<R;i++) /* Esempio: riempio matrici di 1*/
    for(j=0;j<C;j++)
      mat_a[i][j]=mat_b[i][j]=1;
  pipe(p);
  dim=(C+1)*sizeof(int);
  for(i=0;i<R;i++) {
    if ((pid=fork())==0) { /*FIGLIO*/
      close(p[0]);
      for (j=0;j<C;j++)
        ris[j] = mat_a[i][j]+mat_b[i][j];
      ris[j]=i;
      write(p[1],ris,dim);
      sleep(2); /* solo per far leggere il padre
                  su pipe vuota */
      exit(0);
    }
  }
  close(p[1]);
```

```

/* Soluzione con read bloccante
while((nread=read(p[0],ris,dim))==dim) {
    i=ris[C];
    for(j=0;j<C;j++) mat_b[i][j]=ris[j];
}*/

/* Soluzione con read non bloccante
Si poteva usare O_NDELAY ?
*/

if(fcntl(p[0],F_SETFL,O_NONBLOCK)<0) exit(1);

uscita=0;
while(uscita==0) {
    switch(nread=read(p[0], ris, dim)) {
        case -1: /* ho settato O_NONBLOCK */
            perror("read");
            if (errno!=EAGAIN) exit(1);
            sleep(1);
            break;
        case 0:
            printf("pipe chiusa in scrittura\n");
            uscita=1;
            break;
        default:
            i=ris[C];
            for(j=0;j<C;j++) mat_b[i][j]=ris[j];
    }
}

/* Visualizzazione */
for(i=0;i<R;i++) {
    for(j=0;j<C;j++)
        printf("R[%d][%d]:%d  ",i,j,mat_b[i][j]);
    printf("\n");
}
for(i=0;i<R;i++) wait(&stato);
}

```