

Quinta esercitazione

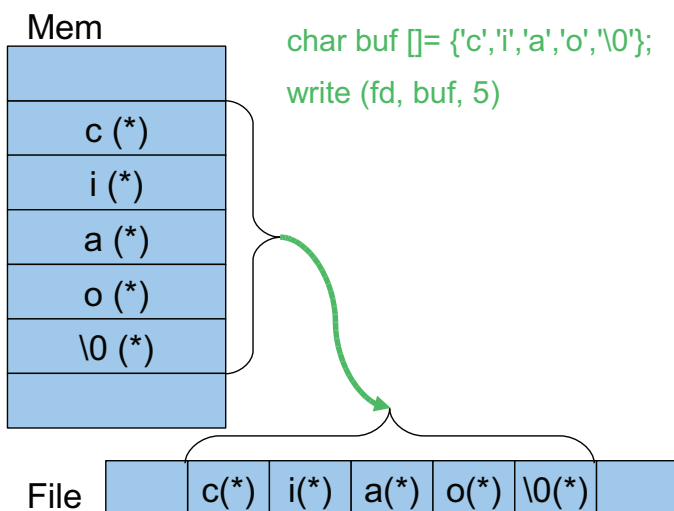
Gestione FileSystem Unix

Stefano Monti
stefano.monti6@unibo.it

Un breve richiamo...

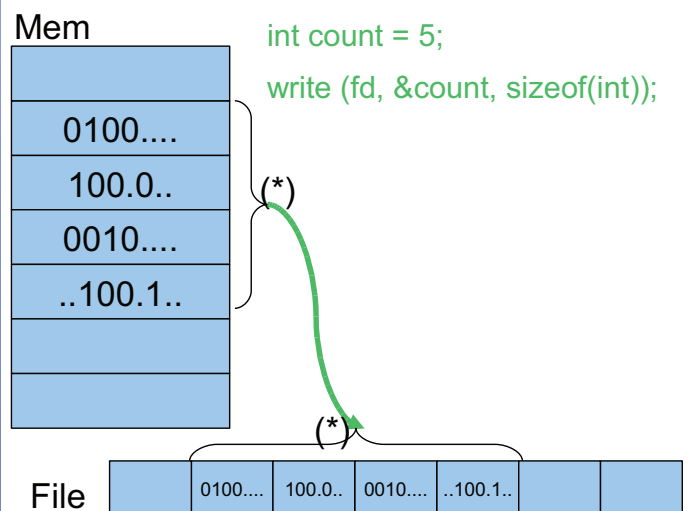
```
int read (int fd, char *buf, int n)  
int write (int fd, char *buf, int n)
```

Lettura/scrittura di file testuali



(*) ciascun byte (in memoria o su file) contiene la rappresentazione binaria del carattere corrispondente (cioè `sizeof(char) == 1`)

Lettura/scrittura di file binari



(*) i 4 byte (in memoria o su file) contengono la rappresentazione binaria dell'intero 5 (cioè `sizeof(int) == 4`)

Esercizio 1

Si realizzi un programma C usando le opportune system call Unix che abbia la seguente interfaccia

contaCaratteri <*c1*> <*c2*> <*N*> <*fileIn*> <*fileOut*>

dove

- *c1* e *c2* sono 2 caratteri
- *N* numero intero
- *fileIn*, *fileOut* due nomi (assoluti) di file

In particolare

- *fileIn* : file **di testo** presente su file system, organizzato in righe di lunghezza non nota a priori
- *fileOut* : nome di file **non presente** su file system

3

Specifiche

Il processo padre P0 deve generare due figli P1 e P2 (fratelli), ognuno dei quali deve

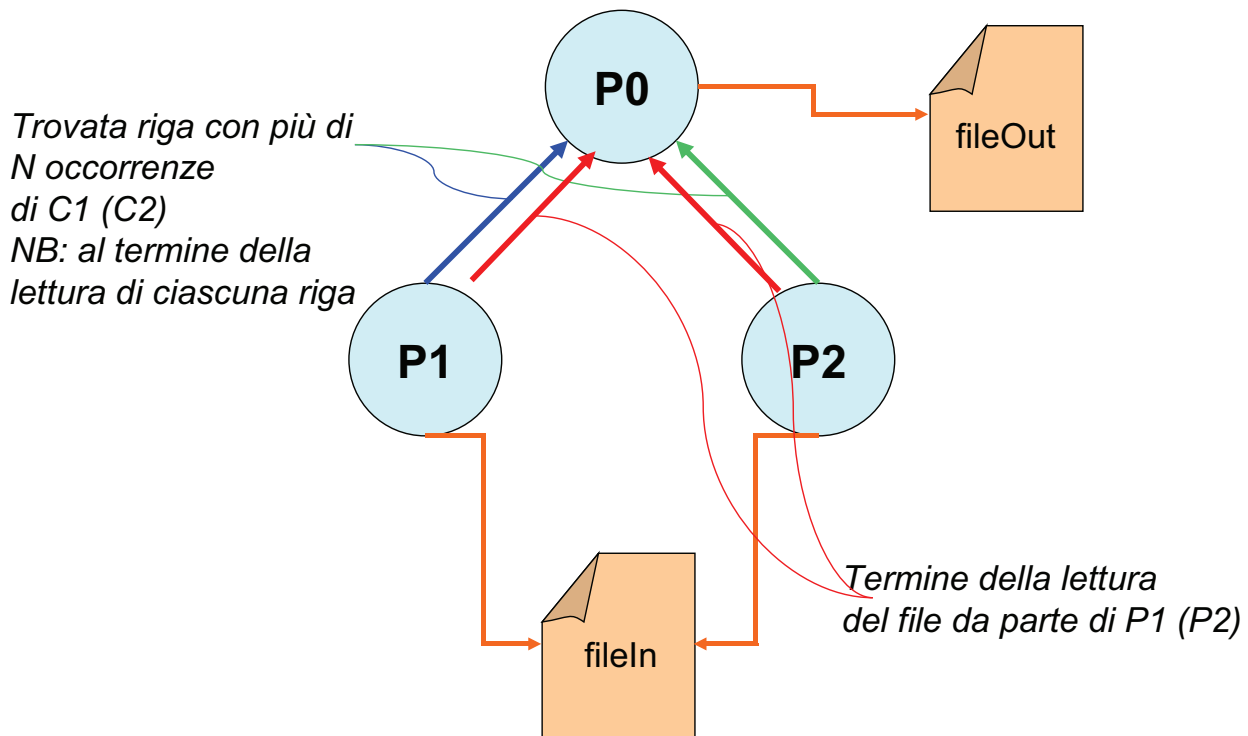
- **leggere** *fileIn* e **contare** le occorrenze del carattere cercato (rispettivamente *c1* per P1 e *c2* per P2) **nella riga** appena letta
- **avvertire** il padre, al termine della lettura di ciascuna riga di *fileIn*, se le occorrenze del carattere cercato sono **più di N**
- **avvertire** il padre al termine della lettura del file

Il padre deve

- tenere traccia, **separatamente** per ciascun figlio, del numero di righe con più di N occorrenze dei caratteri cercati
- al termine delle letture dei figli, scrivere l'informazione al punto precedente su **fileOut**

4

Modello di soluzione



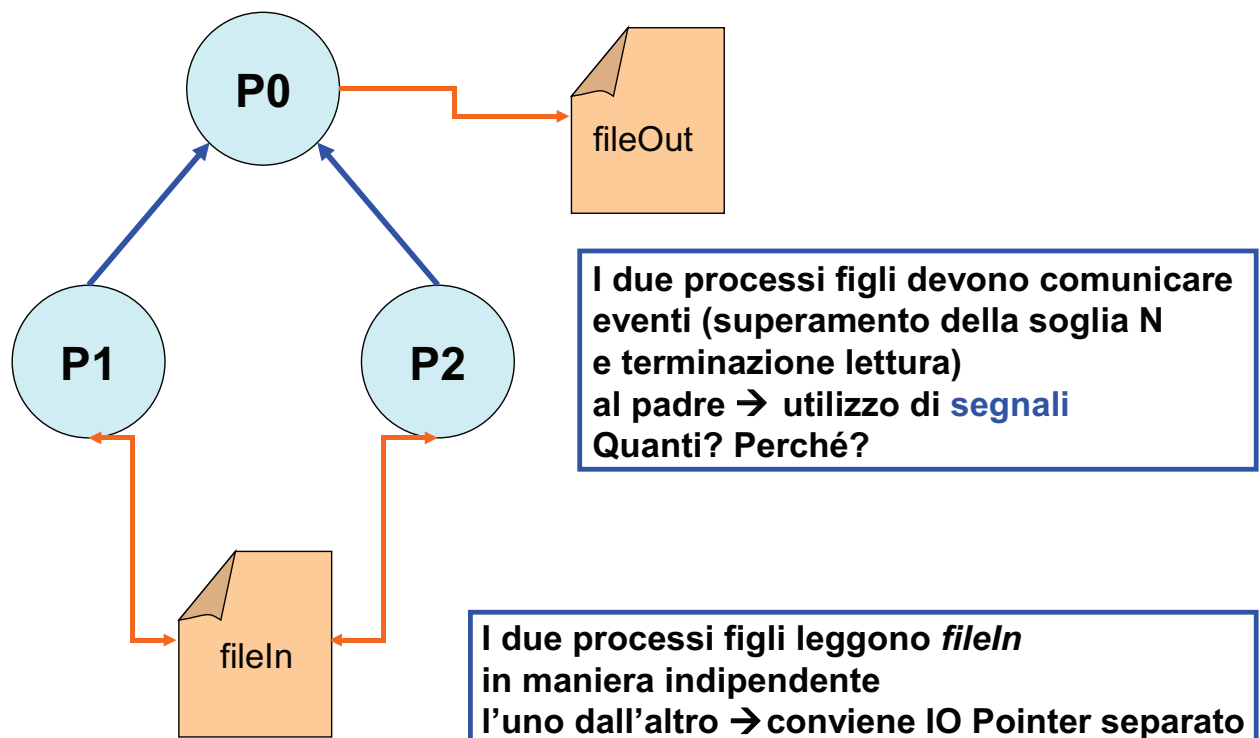
5

Problematiche

- Gestione di file
 - Apertura in lettura o scrittura?
 - Apertura con IO pointer condiviso o separato?
- Gestione della comunicazione interprocesso
 - Quali (e quanti) strumenti?
- Come gestire la sincronizzazione tra processi?
 - in particolare, quanti segnali sono necessari?
- IPOTESI: modello affidabile dei segnali!

6

Strumenti IPC e gestione IO Pointer



7

Generazione processi figli

```
void handler_riga(int sig); void handler_term_lettura(int sig);  
void wait_child();  
int pid0,pid1,pid2; int fileIn,fileOut; int righe_c1 = 0,righe_c2 = 0; int N;  
char c1, c2; int processi_in_lettura=2;
```

```
int main(int argc, char* argv[]) {  
    //--- Generazione figli-----  
    if((pid1=fork())<0) {  
        perror("Errore nella fork:"); exit(-2);  
    }  
    if( pid1 == 0) {//--- Codice figlio P1-----  
    }  
    else{  
        if((pid2=fork())<0) {perror("Errore nella fork:"); exit(-2);}  
        if( pid2 == 0) {//--- Codice figlio P2  
        }  
        else{//--- Codice Padre -----  
        }  
    }  
}
```

8

Codice P1

```
if (pid1==0){//Codice figlio P1
  if((fileIn=open(argv[4],O_RDONLY))<0) {
    perror("Error opening file:"); exit(-1);
  }
  //Ciclo di lettura
  int byteLetti; char c; int occorrenze = 0;
  while((byteLetti=read(fileIn,&c,sizeof(char)))>0){
    if (c == '\n'){//siamo giunti a fine linea
      if (occorrenze > N) {
        kill (pid0,SIGUSR1);}
      occorrenze=0;
    }
    else if (c == c1)
      occorrenze++;
  }
  kill (pid0,SIGTERM);
  //Chiusura delle risorse e uscita
  close(fileIn); exit(0);
}
```

Apertura in sola lettura
Perché IO Pointer non condiviso?

ATTENZIONE: perché leggiamo
carattere per carattere?

Notifica al padre della condizione
“trovati più di N caratteri
c1 in questa riga”
Analogamente, SIGUSR2 per P2

Notifica al padre della condizione
“termine lettura”

9

Codice P0

```
else{//Codice padre
  signal(SIGUSR1, handler_riga); signal(SIGUSR2, handler_riga);
  signal (SIGTERM,handler_term_lettura);
  //Ciclo di attesa terminazione lettura
  while (processi_in_lettura > 0){ pause(); }

  if((fileOut=open(argv[5],O_WRONLY|O_CREAT,0600))<0) {
    perror("Error opening file:"); exit(-1);
  }

  char log[100];
  sprintf(log,"Numero occorrenze %c : %d\n",c1,righe_c1 );
  write(fileOut,log,strlen(log));
  sprintf(log,"Numero occorrenze %c : %d\n",c2,righe_c2 );
  write(fileOut,log,strlen(log));
  //attesa dei figli e terminazione
  close (fileOut);
  wait_child();wait_child();
}
```

Apertura in scrittura/creazione

10

Handler segnali e attesa figli

```
void handler_term_lettura(int sig){
    processi_in_lettura --;
}
void handler_riga(int sig){
    if (sig == SIGUSR1)
        righe_c1 ++;
    else if (sig == SIGUSR2)
        righe_c2 ++;
}
```

```
void wait_child() {
    int pid_terminated,status;
    pid_terminated=wait(&status);
    if(WIFEXITED(status))
        printf("\nPADRE: terminazione volontaria del figlio %d con stato
%d\n",pid_terminated,WEXITSTATUS(status));
    else if(WIFSIGNALED(status))
        printf("\nPADRE: terminazione involontaria del figlio %d a causa del segnale
%d\n",pid_terminated,WTERMSIG(status));
}
```

11

Esercizio 2

Si realizzi un programma C usando le opportune system call Unix che abbia la seguente interfaccia

correggi *<fileIn>* *<fileOut>*

dove *fileIn*, *fileOut* sono due nomi (assoluti) di file.

In particolare

- *fileOut* : nome di file non presente su file system
- *fileIn* : file **binario** presente su file system contenente un numero N non noto a priori di triplette di interi:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|-----|
| A | B | C | A | B | C | A | B | C | ... |
|---|---|---|---|---|---|---|---|---|-----|

Ad esempio:

| | | | | | | | | | |
|---|---|---|----|----|----|----|----|----|-----|
| 1 | 3 | 3 | 78 | 49 | 78 | 40 | 56 | 56 | ... |
|---|---|---|----|----|----|----|----|----|-----|

12

Specifiche

Il processo padre P0 deve generare due figli P1 e P2 (fratelli)

Il processo P2 deve

- **leggere** i primi due interi (A,B) di ogni tripletta in *fileIn*
- segnalare a P1 quale dei due interi (A o B) è il maggiore

Il processo P1 deve

- reperire dal file l'intero maggiore della tripletta (A o B) a seconda della segnalazione ricevuta da P2
- leggere il valore di C e se questo risulta differente dal massimo appena letto
 - **scrivere** il valore dell'intero maggiore **al posto del relativo elemento C** della tripletta
 - **comunicare** a P0 l'avvenuta correzione
- comunicare il termine della elaborazione

Il processo P0 deve

- tenere traccia del numero di correzioni effettuate
- al termine della elaborazione dei figli, **scrivere** tale valore su fileOut 13

Problematiche

- Gestione di file
 - Apertura in lettura o scrittura?
 - Apertura con IO pointer condiviso o separato?
- Gestione della comunicazione interprocesso
 - Quali (e quanti) strumenti?
- Come gestire la sincronizzazione tra processi?
 - in particolare, terminazione della lettura/elaborazione
- IPOTESI: modello affidabile dei segnali

Modello di soluzione

