Calcolatori Elettronici II
19 Aprile 1997

Esercizio di Unix

Si scriva il codice di un programma UNIX (BSD o System V a scelta) che stampi tutti i file contenuti nel direttorio /var/spool/stampa/

Il processo deve essere lanciato dall'utente (da shell), ma deve poter continuare l'esecuzione anche all'uscita dell'utente stesso dal sistema.

Nel progettare la soluzione si tengano presenti i seguenti punti:

- altri processi dello stesso tipo potrebbero essere presenti nel sistema; è quindi necessario garantire la sincronizzazione corretta dei processi;
- si supponga di avere sempre tutti i diritti necessari all'apertura dei file;
- si ricordi l'omogeneità tra file e dispositivi in UNIX e si supponga di stampare utilizzando il file /dev/stampante;

Evidenziare e discutere i punti più importanti incontrati nella soluzione del problema.

Schema della soluzione (Solaris 2.5)
(trasformazione in demone e sincronizzazione tra processi)

```
#include <stdio.h>
#include <signal.h>
#include <fcntl.h>

main(argc, argv)
int argc;
char *argv[];
{
pid_t pid;
pid_t session;
int fdlock, fdstampante, fdfile;
int nread, nwrite;
char buff[BUFSIZ];

pid=fork();
if(pid<0){perror("Errore in fork"); exit(1);}
if(pid>0){  /* Padre*/
   printf("Termine del padre, con pid %d", getpid());
   exit(0);
```

```
} else {  /* codice eseguito dal figlio in background*/
   printf("figlio: pid=%d\n",getpid());
   printf("figlio: pgid=%d\n", getpgid(0));
   session = setsid(); /* sgancio dal terminale */
   printf("figlio: pgid=%d, sid=%d\n", getpgid(0), session);

   /* polling */
   while((fdlock=open("/tmp/lock",O_WRONLY|O_CREAT|O_EXCL,0644))<0){
      perror("Open Lock"); sleep(1);
   }
   if((fdstampante=open("/dev/stampante", O_WRONLY))<0) {
      perror("Open stampante"); exit(1);
   }
   /* stampo solo il file fileprova */
   if((fdfile=open("fileprova", O_RDONLY))<0) {
      perror("Open pippo"); exit(1);
   }
   while((nread=read(fdfile,buff,BUFSIZ))>0) {
      nwrite=write(fdstampante,buff,nread);
   }
   close(fdfile); close(fdstampante); close(fdlock);
   unlink("fileprova");  unlink("lock");
   exit(0);}}
```

Errori più frequenti (19 Aprile 1997)

File

- usare opendir e readdir
- open("nomefile",...) nomefile deve essere nel direttorio corrente
- /dev/stampante è il file device stampante:
  NON è un file eseguibile nè un comando che stampa i file
  NON bisogna crearlo
  NON bisogna cancellarlo dopo l'uso
- è consigliabile cancellare i file stampati (unlink("nomefile"))

Processi

per mettere un processo in background è sufficiente generare un figlio e terminare il padre

```
pid=fork();
if(pid>0)    exit(0);
if(pid==0) ....figlio...
```

Segnali

- sganciare il processo dal terminale
  ```
  session = setsid();
  ```
  o almeno cercare di ignorare qualche segnale
  ```
  signal(SIGINT, SIG_IGN); .....
  ```
- ogni processo ha una propria signal process mask

Sincronizzazione

- NON ci sono variabili condivise tra processi diversi
- Nel caso si usi un lock, attenzione alla possibile non atomicità della sequenza di azioni
- `open("/dev/printer", ..O_EXCL..)` prevedere il caso che la stampante sia occupata e sia necessario attendere
- Usare fcntl() per eseguire il lock va ovviamente bene
  ```
  fcntl(fd, F_SETLKW, &lock)
  ```