

Esercizi di Verifica Aprile 2004

Classe 2

Università di Salerno

Nome e Cognome:

Matricola:

1	2	3	4	5	6	7	tot
/12	/18	/16	/12	/18	/14	/10	/100

Spazio riservato alla correzione

1. 12 punti

Scrivere un programma C che dato in input un file di nome `topolino`, esegua le seguenti operazioni:

- 1) crei un altro file di nome `minni` con i seguenti permessi di accesso: il “proprietario” abbia gli stessi permessi del proprietario di `topolino`, il “gruppo” e gli “altri” non abbiano invece alcun permesso, lo sticky bit il set-user-it ed il set-group-id siano settati come quelli di `topolino`.
- 2) scriva in `minni` i primi 5 byte di `topolino`, lasci un buco di 10 byte ed infine scriva la parola `ciao`.

N.B. Si utilizzino esclusivamente system calls.

2. 18 punti

Si consideri il seguente programma

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

int main()
{
    int fdin, dupin, inr, c;

(1)  if((fdin=open("FILE1",O_RDONLY))<0)
(2)    printf("infile open error\n");

(3)  if((inr=dup(STDIN_FILENO))<0)
(4)    printf("inrestore dup error\n");

(5)  if((dupin=dup2(fdin,STDIN_FILENO))<0)
(6)    printf("dupin error\n");

(7)  while ( read(STDIN_FILENO, &c, 1) > 0)
(8)      write(STDOUT_FILENO, &c, 1);

(9)  if(dup2(inrestore,STDIN_FILENO)<0)
(10)    printf("standard input failed\n");

    exit(0);
}
```

Si assuma di compilarlo e di eseguirlo. Si disegni la process table evidenziando per ciascun file descriptor il file cui esso punta

- a) dopo le istruzioni da (1) a (6);
- b) dopo le istruzioni da (7) a (10).

3. 16 punti

Sia `lso.out` un file eseguibile di proprietà dell'utente `rescigno` che possa essere eseguito oltre che dal proprietario anche dagli appartenenti al gruppo ed agli altri.

Si assuma che `lso.out` faccia la cosa seguente: scrive la user name dell'utente che lo esegue in un file `lista` avente le seguenti caratteristiche

```
-rw----- 1 rescigno 1891 Jun 4 09:45 lista
```

Se cioè un utente `studente` esegue `lso.out` nel file `lista` viene scritta la parola `studente`.

Spiegare dettagliatamente perchè ciò é possibile.

4. 12 punti

Si supponga di avere un file **PLUTO**. Si consideri il seguente programma

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

int main(void)
{
    int fd1, fd2;
    char *s;

    symlink("PLUTO","PIPPO");
    fd1=open("PIPPO",O_RDONLY);
    read(fd1,s,10);
    write(1,s,10);

    link("PLUTO","TOPOLINO");
    fd2=open("TOPOLINO",O_RDONLY);
    read(fd2,s,10);
    write(1,s,10);
}
```

Assumendo di compilare tale programma, dire, giustificando la risposta, quale é l'effetto ottenuto dando **a.out**.

5. 18 punti

Si scriva il codice di una funzione `my_system` che simula il comportamento della system call `system`.

Si assuma che `my_system` prenda in input un vettore contenente la lista delle stringhe di caratteri che compongono il comando shell da eseguire.

Per esempio, sia `vet[4]={"echo","laboratorio","di","sistemi"}` definito in un programma C, allora inserendo nel programma l'istruzione `my_system(vet)`, compilandolo e mandandolo in esecuzione si otterrà sullo standard output la scritta `laboratorio di sistemi`.

6. 14 punti

Si dica, giustificando la risposta, che cosa accade quando si compila e si manda in esecuzione il seguente programma.

```
#include <stdio.h>
int main(void)
{
    pid_t  pid;
    int    *s;

    pid = fork();
    if (!pid) {
        atexit(ex-1);
        while getppid()!=1 do ;
        return(0);
    }
    else {
        atexit(ex-2);
        _exit(0);
    }
}
static void ex-1(void)
{
    printf("primo");
}

static void ex-2(void)
{
    printf("secondo");
}
```

7. 10 punti

Scrivere il comando Bash per gestire le seguenti situazioni:

Descrizione	Comando
visualizzare i nomi dei file nella cwd che finiscono per .c	
lanciare un file eseguibile a.out presente nella directory padre in background	
stampare il contenuto di un file di nome aaa dopo averlo ordinato	
creare un nuovo comando di nome home che faccia coincidere la directory corrente con la home directory	
cambiare il prompt così che esso sia nome utente @ cwd \$	

Descrivere quale è l'effetto dei seguenti comandi:

Comando	Descrizione
ls -l /*	
cat a?a > bbb	
cd /usr/sys echo \$PWD	
UN0="Dani" echo "\${UN0}ele"	
UN0="Dani" echo '\${UN0}ele'	