

## Esercizi di Verifica Aprile 2006

Università di Salerno

Nome e Cognome:

Matricola:

Classe 1  
p-pClasse 2  
p-d

1	2	3	4	5	6	7	tot
/10	/22	/12	/12	/18	/12	/14	/100

Spazio riservato alla correzione

## 1. 10 punti

Scrivere il comando Bash per gestire le seguenti situazioni:

Descrizione	Comando
definire un comando che permetta di stampare i nomi dei file presenti nella cwd	
mandare in esecuzione solo la prima parola l'ultimo dato mandando in esecuzione	
visualizzare la lista dei file presenti in /bin essendo posizionati nella directory /local/bin	
accodare in un file di nome FF il contenuto di un file EE	
stampare i nomi di tutti i file presenti nella directory padre	

Descrivere quale é l'effetto dei seguenti comandi Bash:

Comando	Descrizione
ls [a-f]?[0-9]	
set -o nounset echo \$nome	
pwd	
alias readc='cat *.c' readc	
PATH=\$PATH:.	

## 2. 22 punti

Simulare il comportamento di `cat FILE1 >> FILE2` ricevendo da linea di comando i nomi dei file `FILE1` e `FILE2`, dove

- (14 punti) se `FILE2` esiste allora viene accodato a `FILE2` il contenuto di `FILE1`, se invece `FILE2` non esiste allora `FILE2` viene prima creato con gli stessi permessi di `FILE1` e dopo avviene l'accodamento; e
- (8 punti) si aggiunga che alla fine venga scritto sullo standard output l'avvenuto accodamento.

## 3. 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, fd3;
    char s[15];

    symlink("PLUTO", "PIPP0");
    fd1=open("PIPP0", 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);

    fd3=open("PLUTO", O_RDONLY);
    read(fd3, s, 10);
    write(1, s, 10);
}
```

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

4. 12 punti

Si disegnino la *process table*, la *file table*, e la *v-node table* nei seguenti casi:

- 1) 1 processo ha duplicato il file descriptor di un suo file;
- 2) 2 processi generici accedono ad un medesimo file;
- 3) un processo figlio accede ad un file già aperto dal processo padre.

## 5. 18 punti

Si consideri il seguente programma e si supponga di compilarlo.

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

int main(){
    if (access("prova.txt", W_OK)<0)
        err_ret("access error per prova.txt");
    else printf("access OK\n");
    if (open("prova.txt", O_WRONLY)<0)
        err_ret("open error per prova.txt");
    else printf("open OK\n");
    if (open("prova.txt", O_RDONLY)<0)
        err_ret("open error per prova.txt");
    else printf("open OK\n");
    exit(0);
}
```

Se fosse

```
-r-xr-xr-x 1 rescigno 10932 Jun 4 10:45 a.out
--w----- 1 rescigno 1891  Jun 4 09:45 prova.txt
```

- a) assumendo che si sia connesso l'utente **studente** rispondere alle domande seguenti
  - 1) può **studente** mandare in esecuzione **a.out**?
  - 2) che cosa succede dando **a.out**.
  - 3) assumendo ora di settare a 1 il set-user-id di **a.out**, dire se ci sono cambiamenti dando **a.out**.
- b) assumendo che si sia connesso l'utente **rescigno** rispondere alle domande seguenti (nell'ipotesi che il set-user-id di **a.out** sia a 0)
  - 1) può **rescigno** mandare in esecuzione **a.out**?
  - 2) che cosa succede dando **a.out**.
  - 3) assumendo ora di settare a 1 il set-user-id di **a.out**, dire se ci sono cambiamenti dando **a.out**.

*In tutti i casi la risposta va giustificata.*

## 6. 12 punti

Si consideri il seguente programma.

```
#include    <stdio.h>
int main(void)
{
    pid_t    pid1,pid2;
    int    s;
        s=0;
        pid1=fork();
        if (pid1==0) { atexit(ex-2); atexit(ex-1);
            s=s+1;
                                printf("user: %d\n",s);
                                exit(0);
                                }
        else    {
            atexit(ex-1);  atexit(ex-2);
            wait();
            printf("user: %d\n",s);
            exit(0);
        }
    }
}

static void ex-1(void)
{
    printf("student\n");
}

static void ex-2(void)
{
    printf("prof \n");
}
```

Dire quale e' l'output nei due casi sottostanti, giustificando le risposte.

- a) a.out
- b) a.out > file  
cat file

7. 14 punti

Si scrivano 2 possibili programmi C in cui un processo stampi sullo standard output il proprio pid solo dopo essere diventato figlio di init.