

Esame del 9 luglio 2007

Università di Napoli "Federico II"

Nome e Cognome:

Matricola:

1	2	3	4	tot
/33	/22	/31	/14	/100

1. 33 punti

Si supponga che l'output tipico del comando "ls -al" sia:

```
-rw----- 2 utente1 staff      4096 2007-02-10 14:28 doc1.pdf
-rw-r--r-- 1 utente1 staff       102 2007-01-11 02:05 doc2.pdf
-rw-r--r-- 1 utente2 users     3145 2007-01-16 01:48 rel.tex
-rw----- 1 utente2 users       268 2007-04-26 10:37 song1.mp3
-rw-r--r-- 4 utente3 staff      4096 2006-08-28 19:29 text1.tex
-rw-r--r-- 1 utente3 staff        11 2007-05-02 12:10 text2.tex
```

Si scriva un comando di shell UNIX per ciascuno dei seguenti compiti¹:

- (5 punti) Elencare tutti i gruppi che posseggono almeno un file nella directory, eliminando i duplicati.
- (8 punti) Elencare soltanto i nomi di tutti i file regolari che risultino eseguibili da qualucuno (sia esso l'utente, il suo gruppo o tutti) e che abbiano un nome senza estensione.
- (8 punti) Contare le occorrenze del carattere "c" all'interno del file "rel.tex".

Supponendo che la directory corrente contenga esattamente i file mostrati sopra, si determini l'output dei seguenti comandi:

- (6 punti) `ls | sed 's/ex/ax/' | grep 'ta'`
- (6 punti) `ls -al | awk '{ print $7 }' | sed 's/:.../' | sort | uniq`

2. 22 punti

Si realizzi uno script che elenchi il tipo di file regolari contenuti nella directory classificandoli tramite l'estensione ed elenchi per ogni estensione i gruppi che possiedono almeno un file con tale estensione e il numero di file da essi posseduti.

Ad esempio, se la directory corrente contiene esattamente i file mostrati nell'Esercizio 1, lo script deve produrre il seguente output:

```
.tex: users 1 staff 2
.pdf: staff 2
.mp3: users 1
```

3. 31 punti

Usando soltanto le system call di I/O di basso livello e la funzione `strcmp`, si implementi un programma C chiamato "print_constants", che prende come argomento il nome di un file sorgente in C e stampa il nome di tutte le costanti simboliche (definite con `#define`) definite in quel file.

4. 14 punti

La system call `raise` manda un segnale al processo corrente. Ovvero, `raise(SIGINT)` è equivalente a `kill(getpid(), SIGINT)`.

Dato il seguente programma:

¹Per "comando", si intende qualunque istruzione impartita al prompt della shell, che non contenga strutture di controllo, uso di variabili, o l'operatore ";".

```
void foo(int s) {
    char c;
    lseek(4, -4, SEEK_CUR);
    read(4, &c, 1);
    write(STDOUT_FILENO, &c, 1);
    close(4);
    return;
}

int main(void) {
    int fd = open("prova.txt", O_RDONLY);
    char buf[10];

    dup2(fd, 4);
    signal(SIGUSR1, foo);

    read(fd, buf, 2);
    read(4, &buf[2], 2);
    if (buf[2] == 'a')
        raise(SIGUSR1);
    else
        write(STDOUT_FILENO, &buf[2], 1);

    printf("\n%d\n", read(fd, buf, 10));
    close(fd);
    return 0;
}
```

- (a) (7 punti) Determinare l'output del programma supponendo che il contenuto del file "prova.txt" sia (4 caratteri):

ciao

- (b) (7 punti) Determinare l'output del programma supponendo che il contenuto del file "prova.txt" sia:

carta