

# Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

30 giugno 2008

1. (a) La funzione `f2` riceve due argomenti: un puntatore `a` a una matrice di interi e un intero `c`, e restituisce la somma degli elementi della colonna `c`-esima della matrice `a`. Complessivamente, il programma: (i) legge da standard input una matrice di interi  $N \times N$ ; (ii) per ogni colonna calcola la somma dei valori in essa contenuti; (iii) stampa il più grande dei valori ottenuti al punto precedente.
- (b) Una possibile traduzione è la seguente:

```
.text                                cmpl N, %esi
.global f1                            jge fine
f1: pushl %ebp                         pushl %esi
    movl %esp, %ebp                    pushl 8(%ebp)
    subl $12, %esp                      call f2
    pushl %esi                          addl $8, %esp
    # -4(%ebp) -> b                      cmpl -4(%ebp), %eax
    # -8(%ebp) -> x                      jle avanti
    pushl $0                             movl %eax, -4(%ebp)
    pushl 8(%ebp)                        avanti: incl -8(%ebp)
    call f2                               jmp for
    addl $8, %esp                         fine: popl %esi
    movl %eax, -4(%ebp)                  movl -4(%ebp), %eax
    movl $1, -8(%ebp)                    leave
for: movl -8(%ebp), %esi                 ret
```

2. (a) `#include <sys/types.h>`  
`#include <sys/stat.h>`  
`#include <fcntl.h>`  
`#include <unistd.h>`  
`#include <stdlib.h>`  
`#include <stdio.h>`  
`#include <string.h>`  
`#include <dirent.h>`
- ```
int main(int argc, char* argv[])
{
    int num, i, fd;
    char c, C, dummy;
    DIR *dir;
    struct dirent *ent;
    struct stat buf;

    if (argc != 3) {
        fprintf(stderr, "Uso: %s <num> <car>\n", argv[0]);
        exit(1);
    }
}
```

```

}

if (sscanf(argv[1], "%d", &num) != 1 || num < 0) {
    fprintf(stderr, "%s: parametro errato '%s' "
        "(deve essere un numero >= 0)\n",
        argv[0], argv[1]);
    exit(1);
}

c = argv[2][0];
if (strlen(argv[2]) != 1 || c < 'a' || c > 'z') {
    fprintf(stderr, "%s: parametro errato '%s' "
        "(deve essere un car. compreso tra 'a' e 'z')\n",
        argv[0], argv[2]);
    exit(1);
}
C = c - 'a' + 'A';

if ( !(dir = opendir(".")) ) {
    perror(".");
    exit(1);
}

i = -1;
while ( i < num && (ent = readdir(dir)) ) {
    if ( (stat(ent->d_name, &buf) < 0) || S_ISDIR(buf.st_mode))
        continue;
    i++;
}
closedir(dir);
if (i < num) {
    fprintf(stderr, "%s: file numero '%d' non trovato\n",
        argv[0], num);
    exit(1);
}
if ((fd = open(ent->d_name, O_RDONLY)) < 0) {
    perror(ent->d_name);
    exit(1);
}
while ( read(fd, &dummy, 1) == 1 )
    write(1, &c, 1);
write(1, &C, 1);
return 0;
}

```

```

(b) #include <stdlib.h>
#include <unistd.h>
#include <stdio.h>

const int MAX = 10;
const int MAXNUM = 4;

```

```

int main(int argc, char* argv[])
{
    int conta[MAX], quanti, i, fd[2], nproc;
    char buf1[MAXNUM], buf2[2], c;

    if (argc != 2) {
        fprintf(stderr, "Uso: %s <quanti>\n", argv[0]);
        exit(1);
    }

    if (sscanf(argv[1], "%d", &quanti) != 1 || quanti < 1 || quanti > MAX) {
        fprintf(stderr, "%s: parametro errato '%s' "
            "(deve essere un num. compreso tra 1 e %d)\n",
            argv[0], argv[1], MAX);
        exit(1);
    }

    if (pipe(fd) < 0) {
        perror(argv[0]);
        exit(1);
    }

    for (i = 0; i < quanti; i++)
        conta[i] = 0;

    nproc = 0;
    for (i = 0; i < quanti; i++) {
        switch (fork()) {
            case -1:
                perror(argv[0]);
                continue;
            case 0:
                close(1);
                dup(fd[1]);
                close(fd[0]);
                close(fd[1]);
                sprintf(buf1, "%d", i);
                buf2[0] = i + 'a';
                buf2[1] = '\0';
                execl("corridore", "corridore", buf1, buf2, NULL);
                perror("corridore");
                exit(1);
            default:
                nproc++;
        }
    }
    close(fd[1]);
    if (nproc == 0) {
        fprintf(stderr, "%s: nessun corridore creato\n", argv[0]);
        exit(1);
    }

    while (read(fd[0], &c, 1) > 0) {

```

```

        if (c >= 'a' && c < 'a' + MAX) {
            conta[c - 'a']++;
            continue;
        }
        if (c >= 'A' && c < 'A' + MAX) {
            printf("Vincitore: %d\n", c - 'A');
            printf("Tabella:\n");
            for (i = 0; i < quanti; i++)
                printf("%d: %d\n", i, conta[i]);
            break;
        }
        fprintf(stderr, "%s: errore nell'output di corridore\n");
        exit(1);
    }
    close(fd[0]);
    while (nproc > 0) {
        wait(0);
        nproc--;
    }
    return 0;
}

```