

Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

18 giugno 2009

1. (a) Il programma prende in ingresso tre valori da riga di comando: a , b e c . Quindi stampa una matrice avente a righe e b colonne. La matrice è composta da zeri con l'esclusione degli elementi in cui l'indice della riga è uguale all'indice della colonna. In quest'ultimo caso il valore dell'elemento è un numero casuale compreso tra zero e $c - 1$.
- (b) Una possibile traduzione è la seguente:

<pre>.data formato: .asciz " %d" formato2: .asciz "\n" .text .global f1 f1: pushl %ebp movl %esp, %ebp subl \$12, %esp pushal movl \$0, -4(%ebp) for1: movl -4(%ebp), %ebx cmpl %ebx, 8(%ebp) jle finefor1 movl \$0, -8(%ebp) for2: movl -8(%ebp), %ebx cmpl %ebx, 12(%ebp) jle finefor2 movl -4(%ebp), %edx cmpl %edx, -8(%ebp) jne else</pre>	<pre> pushl 16(%ebp) call f2 movl %eax, -12(%ebp) addl \$4, %esp jmp dopoelse else: movl \$0, -12(%ebp) dopoelse: pushl -12(%ebp) pushl \$formato call printf addl \$8, %esp incl -8(%ebp) jmp for2 finefor2: pushl \$formato2 call printf addl \$4, %esp incl -4(%ebp) jmp for1 finefor1: popal leave ret</pre>
--	--

2. (a)

```
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <unistd.h>
```

```
int main(int argc, char* argv[])
```

```

{
    struct stat st;
    DIR *d;
    struct dirent *e;
    int dim;

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

    if (sscanf(argv[1], "%d", &dim) != 1 || dim < 0) {
        fprintf(stderr, "parametro non valido: '%s'\n", argv[1]);
        exit(1);
    }

    if ( !(d = opendir(".")) ) {
        perror(argv[0]);
        exit(1);
    }

    while ( e = readdir(d) ) {
        if ( stat(e->d_name, &st) < 0) {
            perror(e->d_name);
            continue;
        }
        if (!S_ISREG(st.st_mode))
            continue;
        if (st.st_size > dim)
            printf("%s\n", e->d_name);
    }
    closedir(d);
    return 0;
}

(b) #include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <string.h>

#define MAXNAME 256
#define MAXFILES 100

int main(int argc, char* argv[])
{
    char *c_argv[MAXFILES + 3];
    char names[MAXFILES][MAXNAME];
    int fd[2], n;
    struct stat st;

```

```

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

if ( stat(argv[2], &st) < 0) {
    perror(argv[2]);
    exit(1);
}

if (!S_ISDIR(st.st_mode)) {
    fprintf(stderr, "%s non e' una directory\n", argv[2]);
    exit(1);
}

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

switch ( fork() ) {
case -1:
    perror(argv[0]);
    exit(1);
case 0:
    close(1);
    dup(fd[1]);
    close(fd[0]);
    close(fd[1]);
    execl("bigger", "bigger", argv[1], NULL);
    perror("bigger");
    exit(1);
default:
    break;
}
close(0);
dup(fd[0]);
close(fd[0]);
close(fd[1]);

c_argv[0] = "mv";
n = 0;
while ( n < MAXFILES && fgets(names[n], MAXNAME, stdin) ) {
    c_argv[n + 1] = names[n];
    c_argv[n + 1][strlen(names[n]) - 1] = '\0';
    n++;
}
c_argv[n + 1] = argv[2];
c_argv[n + 2] = NULL;

wait(0);

switch ( fork() ) {

```

```
case -1:
    perror(argv[0]);
    exit(1);
case 0:
    execvp("mv", c_argv);
    perror("mv");
    exit(1);
default:
    break;
}

wait(0);

return 0;
}
```