

Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

14 Luglio 2006

1. (a) La funzione `f1` riceve tre argomenti: un vettore di interi `v`, un intero `n` (che rappresenta il numero di elementi del vettore `v`) e un intero `i`, che può valere `-1`, `0` o `1`. La funzione restituisce il numero di elementi del vettore `v` che sono negativi (se `i` vale `-1`), nulli (se `i` vale `0`) o positivi (se `i` vale `1`). La funzione `f2` chiama la funzione `f1` per tre volte, passandole sempre lo stesso vettore `v`, e ogni volta, un valore diverso per `i`. Quindi, memorizza i risultati così ottenuti nel vettore `u`. Infine, la funzione `main` costruisce un vettore `vv` con gli interi letti dal file, il cui nome le viene passato da riga di comando, e lo passa alla funzione `f2`. Quindi, stampa i risultati. Il programma stampa il numero di interi negativi, nulli e positivi contenuti nel file il cui nome viene passato come argomento.

- (b) Una possibile traduzione è la seguente:

```
.text                                pushl 12(%ebp)
.global f2                           pushl 8(%ebp)
                                       call f1
f2:                                   addl $12, %esp
    pushl %ebp                       movl %esi, %edi
    movl %esp, %ebp                  incl %edi
                                       movl %eax, (%ebx, %edi, 4)
    pushl %esi                       incl %esi
    pushl %edi                       jmp for
    pushl %ebx
    movl 16(%ebp), %ebx              fine: popl %ebx
    movl $-1, %esi                  popl %edi
                                       popl %ebx
for:  cmpl $1, %esi
    jg fine                          leave
    pushl %esi                       ret
```

2. (a) `#include <sys/types.h>`
`#include <sys/stat.h>`
`#include <unistd.h>`
`#include <dirent.h>`
`#include <stdio.h>`
- ```
int main(int argc, char* argv[])
{
 int primo, ultimo, conta = 0, stampati = 0;
 DIR* cwd;
 struct dirent* dir;
 struct stat st;

 if (argc != 3) {
 fprintf(stderr, "Uso: %s primo ultimo\n", argv[0]);
```

```

 exit(1);
 }
 if (sscanf(argv[1], "%d", &primo) != 1 ||
 sscanf(argv[2], "%d", &ultimo) != 1 ||
 primo < 0 || ultimo < 0 || ultimo < primo) {
 fprintf(stderr, "parametri errati\n");
 exit(1);
 }
 if ((cwd = opendir(".")) == NULL) {
 perror(argv[1]);
 exit(1);
 }
 while ((dir = readdir(cwd)) != NULL && conta < ultimo) {
 if (stat(dir->d_name, &st) < 0) {
 perror(dir->d_name);
 continue;
 }
 if (!S_ISREG(st.st_mode))
 continue;
 if (conta++ >= primo) {
 if (stampati++ > 0)
 printf(" ");
 printf("%s", dir->d_name);
 }
 }
 printf("\n");
 return 0;
}

```

```

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

#define MAX_ARGS 100
#define BUF_SIZE 1024

int main(int argc, char* argv[])
{
 char* args[MAX_ARGS + 3];
 char buf[BUF_SIZE];
 int fd[2];
 int conta, num, n, i;

 if (argc != 3) {
 fprintf(stderr, "Uso: %s num dir\n", argv[0]);
 exit(1);
 }
 if (sscanf(argv[1], "%d", &num) != 1 || num < 0 || num > MAX_ARGS) {
 fprintf(stderr, "parametro num errato\n");
 exit(1);
 }

 if (pipe(fd) < 0) {

```

```

 perror(argv[0]);
 exit(1);
 }
 switch (fork()) {
 case -1:
 perror(argv[0]);
 exit(1);
 case 0: // figlio
 close(1);
 dup(fd[1]);
 close(fd[0]);
 close(fd[1]);
 execl("lista", "lista", "0", argv[1], NULL);
 perror("lista");
 exit(1);
 }
 close(fd[1]);
 if ((n = read(fd[0], buf, BUF_SIZE)) < 0) {
 perror(argv[0]);
 exit(1);
 }
 close(fd[0]);
 wait(NULL);
 if (n == 0 || buf[n - 1] != '\n') {
 fprintf(stderr, "errore nell'esecuzione di lista\n");
 exit(1);
 }
 buf[n - 1] = '\0';
 conta = 1;
 args[1] = buf;
 for (i = 0; i < n; i++) {
 if (buf[i] == ' ' && conta < MAX_ARGS) {
 buf[i] = '\0';
 args[conta + 1] = &buf[i + 1];
 conta++;
 }
 }

 if (conta > 0) {
 args[0] = "cp";
 args[conta + 1] = argv[2];
 args[conta + 2] = NULL;
 execvp("cp", args);
 perror("cp");
 exit(1);
 }
 return 0;
}

```

