

Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

15 febbraio 2008

1. (a) La funzione `f2` riceve come argomenti un vettore `v` e due interi `i` e `z`. Restituisce 1 se `z` è maggiore di `v[i]`, e 0 altrimenti. La funzione `f1` copia nel vettore `u` tutti gli elementi di `v` minori di `z`, quindi restituisce il numero di elementi copiati. Complessivamente, il programma accetta due parametri da riga di comando: un intero `z` e un nome di file (che si suppone contenga numeri interi). Quindi, stampa sull'uscita standard tutti gli interi contenuti nel file che sono minori di `z`.

- (b) Una possibile traduzione è la seguente:

```
.text                                pushl 8(%ebp)
.global f1                            call f2
f1:  pushl %ebp                          addl $12, %esp
      movl %esp, %ebp                  cmpl $0, %eax
      # uso esi come i                je avanti
      # e edi come j                  movl (%ebx, %esi, 4), %edx
      pushl %ebx                       movl %edx, (%ecx, %edi, 4)
      pushl %ecx                       incl %edi
      pushl %edx                       avanti: incl %esi
      pushl %edi                       jmp ciclo
      pushl %esi                       fine:  movl %edi, %eax
      movl $0, %esi                    popl %esi
      movl $0, %edi                    popl %edi
      movl 8(%ebp), %ebx               popl %edx
      movl 20(%ebp), %ecx              popl %ecx
ciclo: cmpl %esi, 12(%ebp)             popl %ebx
      jle fine                          leave
      pushl 16(%ebp)                   ret
      pushl %esi
```

2. (a)

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

#define MAXLINE 1024

int main(int argc, char* argv[])
{
    char buf[MAXLINE];
    int numwords, i;
    int bufsize;
    int first;

    if (argc != 3) {
        fprintf(stderr, "Uso: %s <pos> s|d\n", argv[0]);
```

```

        exit(1);
    }

    first = atoi(argv[1]);

    if (first < 0) {
        fprintf(stderr, "<pos> deve essere maggiore o uguale a 0\n");
        exit(1);
    }

    if (argv[2][0] != 's' && argv[2][0] != 'd') {
        fprintf(stderr, "<dir> deve essere 's' oppure 'd'\n");
        exit(1);
    }

    while (fgets(buf, MAXLINE, stdin)) {
        int contatab = 0;

        bufsize = strlen(buf);
        if (bufsize == 0)
            continue;
        bufsize--;
        for (i = 0; i < bufsize; i++) {
            if (argv[2][0] == 's' && contatab <= first ||
                argv[2][0] == 'd' && contatab >= first)
                putchar(buf[i]);
            if (buf[i] == '\t')
                contatab++;
        }
        putchar('\n');
    }
}

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

void errore(char *m)
{
    perror(m);
    exit(1);
}

int main(int argc, char *argv[])
{
    int pp[2];
    pid_t pid1, pid2;

    if (argc != 4) {
        fprintf(stderr, "Uso: %s <inf> <sup> <nomefile>\n", argv[0]);
        exit(1);
    }
}

```

```

if (pipe(pp) == -1)
    errore(argv[0]);

if ((pid1 = fork()) < 0)
    errore(argv[0]);
else if (pid1 == 0) {
    close(0);
    if (open(argv[3], O_RDONLY) < 0)
        errore(argv[0]);

    close(1);
    dup(pp[1]);
    close(pp[0]);
    close(pp[1]);
    execl("taglia", "taglia", argv[2], "s", NULL);
    exit(1);
}
if ((pid2 = fork()) < 0)
    errore(argv[0]);
else if (pid2 == 0) {
    close(0);
    dup(pp[0]);
    close(pp[0]);
    close(pp[1]);
    execl("taglia", "taglia", argv[1], "d", NULL);
    exit(1);
}
close(pp[0]);
close(pp[1]);
wait(0);
wait(0);
}

```