

Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

19 febbraio 2010

1. (a) Il programma legge da ingresso standard una sequenza di numeri interi terminata da un numero negativo. Successivamente il programma ordina in modo decrescente i valori letti (con esclusione del numero negativo) e li stampa su uscita standard.
- (b) Una possibile traduzione è la seguente:

```
.text                                j1    e4
.global f1
f1:                                  movl  (%ecx, %ebx, 4), %eax
    pushl %ebp                       cmpl  %eax, (%ecx, %edi, 4)
    movl  %esp, %ebp                 jge   avanti
                                      movl  %ebx, %edi
    pushal                            avanti:
                                      incl  %ebx
                                      jmp   e3
    # esi -> i                          e4:
    # edi -> ind                        pushl %esi
    # ebx -> j                          pushl %edi
    # ecx -> v                          pushl %ecx
                                      call  f2
    movl  8(%ebp), %ecx               addl  $12, %esp
    decl  12(%ebp)
    movl  $0, %esi
e1:                                  incl  %esi
    cmpl  %esi, 12(%ebp)              jmp   e1
    jle   e2                          e2:
    movl  %esi, %edi
    movl  %esi, %ebx
    incl  %ebx
    popal
e3:                                  leave
    cmpl  %ebx, 12(%ebp)              ret
```

2. (a)

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

#define MAXLINE 1024

void testa(FILE *f, int n)
{
    int i;
    char buf[MAXLINE];

    for (i = 0; i < n; i++)
        if (fgets(buf, MAXLINE, f))
```

```

                fputs(buf, stdout);
            else
                break;
    }

int main(int argc, char* argv[])
{
    int nlines = 10, i;
    FILE *f;

    if (argc > 1) {
        if (sscanf(argv[1], "%d", &nlines) == 1) {
            if (nlines <= 0) {
                fprintf(stderr, "argomento erratto: %d", nlines);
                exit(1);
            }
            argv++;
            argc--;
        }
    }
    if (argc == 1) {
        testa(stdin, nlines);
    } else {
        for (i = 1; i < argc; i++) {
            if ( !(f = fopen(argv[i], "r")) ) {
                perror(argv[i]);
                continue;
            }
            testa(f, nlines);
            fclose(f);
        }
    }
    return 0;
}

```

```

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

#define MAXARGS 20

int main(int argc, char* argv[])
{
    char* argv2[MAXARGS];
    int i, fd[2];

    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]);
    argv2[0] = "testa";
    argv2[1] = "20";
    for (i = 1; i < MAXARGS - 2; i++)
        argv2[i + 1] = argv[i];
    argv2[i] = NULL;
    execv("testa", argv2);
    perror("testa");
    exit(1);
default:
    break;
}

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

close(fd[0]);
close(fd[1]);

wait(NULL);
wait(NULL);

return 0;
}

```