

Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

5 Giugno 2006

1. (a) La funzione `f2` riceve, come unico argomento, un intero n . La funzione conta il numero di bit pari a 1 che compaiono nella rappresentazione binaria del numero n . La funzione `f1` conta i numeri, compresi tra `v1` e `v2` (estremi inclusi), la cui rappresentazione binaria contiene un solo bit pari a 1 (potenze di 2, nel caso di numeri positivi). Infine, la funzione `main` chiama la funzione `f1` passandole, come estremi della ricerca, i numeri ricevuti da riga di comando. Il programma (se il primo estremo dell'intervallo è positivo o nullo) stampa il numero di potenze di 2 contenute nell'intervallo chiuso i cui estremi sono passati come argomenti.

- (b) Una possibile traduzione è la seguente:

```
.text                                jg     fine
.global f1                           pushl  -4(%ebp)
f1:                                   call   f2
    pushl  %ebp                       addl   $4, %esp
    movl   %esp, %ebp                 movl   %eax, -12(%ebp)
    subl   $12, %esp                  cmpl   $1, -12(%ebp)
    # i vale -4(%ebp)                 jne    ava
    # j vale -8(%ebp)                 incl   -8(%ebp)
    # k vale -12(%ebp)                ava:
    # v1 vale 8(%ebp)                 incl   -4(%ebp)
    # v2 vale 12(%ebp)                jmp    for
    pushl  %ebx                       fine:
    push   %ecx                       movl   -8(%ebp), %eax

    movl   $0, -8(%ebp)                popl   %ecx
    movl   12(%ebp), %ecx              popl   %ebx
    movl   8(%ebp), %ebx               leave
    movl   %ebx, -4(%ebp)              ret

for:  cmpl   %ecx, -4(%ebp)
```

2. (a)

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

int main(int argc, char *argv[])
{
    int x, i;
    int soglia;
    char *ops[] = { "l", "le", "g", "ge" };
    int oo = -1;

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

```

        exit(1);
    }
    for (i = 0; i < 4; i++)
        if (strcmp(argv[1], ops[i]) == 0) {
            oo = i;
            break;
        }
    if (oo == -1) {
        fprintf(stderr, "Condizione non corretta (g, ge, l ,le)\n");
        exit(1);
    }
    if (sscanf(argv[2], "%d", &soglia) != 1) {
        fprintf(stderr, "Parametro soglia errato");
        exit(1);
    }
    while (scanf("%d", &x) != EOF) {
        switch (oo) {
            case 0:
                if (x < soglia)
                    printf("%d\n", x);
                break;
            case 1:
                if (x <= soglia)
                    printf("%d\n", x);
                break;
            case 2:
                if (x > soglia)
                    printf("%d\n", x);
                break;
            case 3:
                if (x >= soglia)
                    printf("%d\n", x);
        }
    }
}
}

(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) {
        // 1o figlio...
        close(0);
        if (open(argv[3], O_RDONLY) < 0)
            errore(argv[0]);

        close(1);
        dup(pp[1]);
        close(pp[0]);
        close(pp[1]);
        execl("soglia", "soglia", "ge", argv[1], NULL);
        exit(1);
    }
    // padre...
    if ((pid2 = fork()) < 0)
        errore(argv[0]);
    else if (pid2 == 0) {
        // 2o figlio...
        close(0);
        dup(pp[0]);
        close(pp[0]);
        close(pp[1]);
        execl("soglia", "soglia", "l", argv[2], NULL);
        exit(1);
    }
    // padre ...
    close(pp[0]);
    close(pp[1]);
    wait(0);
    wait(0);
}

```