

Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

3 luglio 2009

1. (a) Il programma prende in ingresso, da riga di comando, due valori interi a e b . Quindi stampa tutti gli interi minori di b che hanno nella loro rappresentazione binaria un numero di bit ad 1 esattamente pari ad a .
- (b) Una possibile traduzione è la seguente:

```
.data                                addl $12, %esp
frmt1: .asciz "%d"                   ciclo:
frmt2: .asciz "%d\n"                 cml %esi, -8(%ebp)
.text                                  jle ff
.global f1                             pushl %esi
f1:                                    call f2
    pushl %ebp                        addl $4, %esp
    movl %esp, %ebp                   cml %eax, -4(%ebp)
    subl $8, %esp                     jne av
                                      pushl %esi
    pushl %ebx                        pushl $frmt2
    pushl %esi                         call printf
                                      addl $8, %esp
    movl 12(%ebp), %ebx               av:
    movl $0, %esi                     incl %esi
    cml $3, 8(%ebp)                  jmp ciclo
    jne ret1                           ff:
                                      movl $0, %eax
                                      jmp fine
    leal -4(%ebp), %eax               ret1:
    pushl %eax                         movl $1, %eax
    pushl $frmt1
    pushl 4(%ebx)
    call sscanf
    addl $12, %esp
    leal -8(%ebp), %eax
    pushl %eax
    pushl $frmt1
    pushl 8(%ebx)
    call sscanf
    fine:
    popl %esi
    popl %ebx
    leave
    ret
```

2. (a) #include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

```

#define MAX_NAME 256
#define BUF_SIZE 1024

int main(int argc, char* argv[])
{
    char name[MAX_NAME];
    char buf[BUF_SIZE];
    int nbytes = BUF_SIZE;
    int fin, fout, n, tocopy, nfile;
    struct stat st;

    if (argc < 3 || argc > 4) {
        fprintf(stderr, "Uso: %s <orig> <new> [<byte>]\n", argv[0]);
        exit(1);
    }

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

    if (!S_ISREG(st.st_mode)) {
        fprintf(stderr, "%s: non e' un file regolare\n", argv[1]);
        exit(1);
    }

    if (argc == 4) {
        if (sscanf(argv[3], "%d", &nbytes) != 1 || nbytes <= 0 || nbytes > BUF_SIZE)
            fprintf(stderr, "Il terzo argomento deve essere un intero compreso tra 1 e %d\n", BUF_SIZE);
        exit(1);
    }

    if ( (fin = open(argv[1], O_RDONLY)) < 0) {
        perror(argv[1]);
        exit(1);
    }

    nfile = 0;
    for (;;) {
        sprintf(name, "%s-%03d", argv[2], nfile);
        if ( (fout = open(name, O_WRONLY | O_CREAT | O_TRUNC, 0666)) < 0) {
            perror(name);
            exit(1);
        }
        n = 0;
        tocopy = nbytes;
        while ( tocopy > 0 && (n = read(fin, buf, tocopy)) > 0) {
            write(fout, buf, n);
            tocopy -= n;
        }
        close(fout);
        if (n == 0)

```

```

        break;
        nfile++;
    }
    return 0;
}

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

#define MAX_NAME 256

int main(int argc, char* argv[])
{
    char buf[MAX_NAME];
    int fd;

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

    if ( (fd = open(argv[1], O_WRONLY|O_CREAT|O_APPEND, 0666)) < 0) {
        perror(argv[1]);
        exit(1);
    }

    while (fgets(buf, MAX_NAME, stdin) != NULL) {
        buf[strlen(buf) - 1] = '\0';
        switch (fork()) {
            case -1:
                perror(argv[0]);
                break;
            case 0:
                close(2);
                dup(fd);
                close(fd);
                execl("splitb", "splitb", buf, buf, argv[2], NULL);
                perror("splitb");
                exit(1);
            default:
                wait(0);
                break;
        }
    }

    return 0;
}

```

