

# Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

17 settembre 2010

1. (a) Il programma prende in ingresso da riga di comando un intero ( $n$ ) e un nome di file. Quindi stampa a video tutte le righe del file specificato che hanno una lunghezza pari ad  $n$  caratteri.  
(b) Una possibile traduzione è la seguente:

<pre>.text .global f2 f2:     pushl %ebp     movl  %esp, %ebp      pushl %eax     pushl %ebx      subl  MAXL, %esp     movl  %esp, %ebx      cmpl  \$0, 12(%ebp)     je    fine ciclo:     pushl 12(%ebp)     pushl MAXL     pushl %ebx     call  fgets</pre>	<pre>addl  \$12, %esp cmpl  \$0, %eax je    fine pushl %ebx call  f1 addl  \$4, %esp cmpl  %eax, 8(%ebp) jne   avanti pushl stdout pushl %ebx call  fputs addl  \$8, %esp avanti:     jmp  ciclo fine:     popl %ebx     popl %eax     leave     ret</pre>
---	--

2. (a) 

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

#define MAXLINE 80

int reverse1(FILE* f) {
    char buf[MAXLINE + 1];
    char *ptr;

    if (!fgets(buf, MAXLINE + 1, f))
        return 0;

    if (! (ptr = strchr(buf, '\n')) ) {
        reverse1(f);
        ptr = buf + strlen(buf) - 1;
    }

    while (--ptr >= buf)
```

```

        putchar(*ptr);

    return 1;
}

void reverse(FILE *f) {
    int cont = 1;
    do {
        if (cont = reversel(f))
            putchar('\n');
    } while (cont);
}

int main(int argc, char* argv[])
{
    int i;

    if (argc == 1)
        reverse(stdin);
    else
        for (i = 1; i < argc; i++) {
            FILE *f;

            if ( !(f = fopen(argv[i], "r")) ) {
                perror(argv[i]);
                continue;
            }

            reverse(f);

            fclose(f);
        }
    return 0;
}
(b) #include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <wait.h>

int main(int argc, char* argv[])
{
    int fd[2];
    int status;

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

```

```

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]);
    execl("reverse", "reverse", argv[1], NULL);
    perror("reverse");
    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]);
    close(1);
    if (open(argv[2], O_WRONLY | O_TRUNC | O_CREAT, 0666) < 0) {
        perror(argv[2]);
        exit(1);
    }
    execl("reverse", "reverse", NULL);
    perror("reverse");
    exit(1);
default:
    break;
}

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

wait(0);
wait(0);

switch (fork()) {
case -1:
    perror(argv[0]);
    exit(1);
case 0:

```

```
        execlp("cmp", "cmp", argv[1], argv[2], NULL);
        perror("cmp");
        exit(1);
default:
        break;
}

wait(&status);

if (WEXITSTATUS(status) == 0) {
        printf("OK\n");
        return 0;
} else {
        printf("ERROR\n");
        return 1;
}
}
```