Prova scritta di Sistemi di Elaborazione
Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

19 settembre 2008

1. Supponiamo di avere il seguente programma scritto in parte in Assembler e in parte in C++:

```
.text
.global f1
f1:     pushl %ebp
        movl  %esp, %ebp
        pushl %ebx
        pushl %ecx
        pushl %edx
        movl  8(%ebp), %ebx
        movl  12(%ebp), %ecx
uno:    movb  (%ebx), %dl
        subb  (%ecx), %dl
        cmpb  $0, %dl
        jne   l1
        cmpb  $0, (%ebx)
        je    l1
        incl  %ebx
        incl  %ecx
        jmp   uno
l1:     cmpb  $0, %dl
        jg    l2
        jl    l3
        movl  $0, %eax
        jmp   fine
l2:     movl  $1, %eax
        jmp   fine
l3:     movl  $-1, %eax
        jmp   fine
fine:   popl  %edx
        popl  %ecx
        popl  %ebx
        leave
        ret
```

```
#include <stdio.h>

const int MAXB = 100;

int f1(char *s1, char *s2);

void f2(FILE * pf, char *s)
{
    int x;
    char buf[MAXB];
    while (fscanf(pf, "%s", buf) == 1) {
        x = f1(buf, s);
        if (x < 0)
            printf("%s\n", buf);
    }
}

int main(int argc, char *argv[])
{
    FILE *pf;
    pf = fopen(argv[2], "r");
    f2(pf, argv[1]);
    return 0;
}
```

(a) Dire cosa viene calcolato dal programma complessivo.

(b) Tradurre la funzione `f2` in Assembler.

2. Scrivere i seguenti programmi in C++, utilizzando le primitive di Unix e la libreria standard del C.

   (a) Un programma `tee`, con zero o più argomenti da riga di comando. Ogni argomento è un nome di file (che si suppone non cominci con il carattere "`-`"), oppure una stringa della forma "`-`n", dove n è un numero naturale maggiore di 2. Il programma deve copiare il suo ingresso standard sull'uscita standard e su ognuno dei file ricevuti come argomento (se non esistono, vanno creati). Inoltre, per ogni argomento di tipo "`-`n", il programma deve copiare il suo ingresso standard anche sul descrittore di file numero n (che si suppone sia già stato aperto dal processo che esegue il programma).

   (b) Un programma `ptee` con uno o più argomenti da riga di comando. Ogni argomento è il nome di un comando. Per ogni argomento, il programma deve creare un processo che esegue il corrispondente comando. Il programma deve inoltre creare un ulteriore processo, che esegue il programma `tee` con argomenti opportuni. Questo processo deve comunicare con tutti i processi precedenti tramite una pipe (distinta per ogni processo), mentre la sua uscita standard deve essere rediretta sul file speciale "`/dev/null`". Il programma principale deve quindi attendere che tutti i processi figli terminino, prima di terminare esso stesso.