

# Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

16 Febbraio 2007

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
      subl $4, %esp
      pushl %esi
      pushl %ebx
      movl $0, -4(%ebp)
      movl $0, %esi
      movl 8(%ebp), %ebx
L1:  movb (%ebx, %esi), %al
      cmpb $0x00, %al
      jz L3
      cmpb %al, 12(%ebp)
      jne L2
      incl -4(%ebp)
L2:  incl %esi
      jmp L1
L3:  movl -4(%ebp), %eax
      popl %ebx
      popl %esi
      leave
      ret
```

```
#include <stdio.h>
const int MAXN = 1000;
const int MAXL = 100;
int f1(char* p, char c);
int f2(char* a, char* b){
    int i = 0;
    int r = 0;
    while(b[i] != '\0') {
        r += f1(a, b[i]);
        i++;
    }
    return r;
}

int main(int argc, char* argv[]) {
    int i, q = 0;
    char m[MAXN][MAXL];
    FILE* ff = fopen("input.txt", "r");
    while(fscanf(ff, "%s", m[q]) == 1)
        q++;
    fclose(ff);
    for(i=0; i<q; i++)
        printf("%d\n", f2(m[i], 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 `lines` con un argomento *file* da riga di comando. Il programma deve leggere numeri interi maggiori o uguali a 0 dal suo ingresso standard (fino a quando tale ingresso non viene chiuso). Per ogni numero *n* letto, il programma deve mostrare sulla sua uscita standard la linea *n*-esima (contando a partire da 0) del file *file*. **Suggerimento:** si usi `fseek` per riportarsi ogni volta all'inizio del file.
  - (b) Un programma `reverse` con due argomenti *file* e *linee* da riga di comando. Il programma deve creare due processi figli, collegati tramite una pipe, di cui il secondo esegua il programma `lines` con argomento *file*, e il primo scriva opportunamente sulla propria uscita standard, in modo che il secondo mostri in uscita le prime *linee* linee del file *file*, in ordine inverso. Il programma principale deve attendere la terminazione di entrambi i suoi figli.