

Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

22 Settembre 2006

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

```
.text
.global f1
f1:   push  %ebp
      movl %esp, %ebp
      pushl %ebx
      pushl %ecx
      pushl %edx
      cmpl $0, 8(%ebp)
      jne  avanti
      movl $0, %edx
      jmp  fine
avanti: movl $0, %ebx
        movl $1, %ecx
        movl $1, %edx
L1:    cmpl $1, 8(%ebp)

      jle  fine
      movl %ecx, %edx
      addl %ebx, %edx
      movl %ecx, %ebx
      movl %edx, %ecx
      decl 8(%ebp)
      jmp  L1
fine:  movl %edx, %eax
      popl %edx
      popl %ecx
      popl %ebx
      leave
      ret

#include <stdio.h>
const int MAXL = 100;

void f2(int v[], int n)
{
    int i;
    for (i = 0; i < n; i++)
        if (v[i] >= 0)
            v[i] = f1(v[i]);
}

int main(int argc, char *argv[])
{
    int i;
    int vett[MAXL];
    for (i = 1; i < argc; i++)
        sscanf(argv[i], "%d",
                &vett[i - 1]);
    f2(vett, argc - 1);
    for (i = 0; i < argc - 1; i++)
        printf("%d\n", vett[i]);
}
```

- (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 **parole**, con due argomenti (da riga di comando) *file* e *num*. Il primo argomento deve essere un nome di file, mentre il secondo un numero intero maggiore o uguale a zero. Il programma deve stampare (sulla sua uscita standard) tutte le parole contenute nel file *file*, in modo che ogni linea (tranne, eventualmente, l'ultima) contenga esattamente *num* parole. Per "parole", si intende una qualunque sequenza di caratteri, delimitata da spazi. Si supponga che ci sia una dimensione massima, nota a-priori, per la dimensione delle parole contenute nei file.
- (b) Un programma **elenco**, con un argomento (da riga di comando) *file*. Il programma deve mostrare sulla sua uscita standard tutte le parole contenute nel file *file*, una per riga, in ordine lessicografico e senza ripetizioni. Per svolgere il suo compito, il programma **elenco** deve creare tre processi, collegati in serie tramite due pipe. Il primo processo deve eseguire il programma **parole**, con argomenti opportuni, e il secondo processo deve eseguire il comando di sistema **sort**. Il programma **sort**, invocato senza argomenti, ordina lessicograficamente le linee che riceve sul suo ingresso standard e le mostra in uscita. Il terzo processo deve eliminare le ripetizioni tra le parole.