

Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

30 gennaio 2009

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

```
.text                                idiv  %edi
.global f1                           cmpl  $0, %edx
f1:  pushl %ebp                        jne   av
      movl %esp, %ebp                 addl  %edi, %esi
      pushl %edi                       av:  incl  %edi
      pushl %esi                       jmp   l1
      movl $1, %esi                    l2:  movl %esi, %eax
      movl $2, %edi                    popl  %esi
l1:  cmpl %edi, 8(%ebp)                popl  %edi
      jle  l2                          leave
      movl 8(%ebp), %eax               ret
      movl $0, %edx

#include <stdio.h>                    return a;
                                     }
const int MAXN = 100;
int f1(int x);                        int main(int argc, char *argv[])
                                     {
int f2(int n, int v[])                int vv[MAXN];
{                                       int l, q, i;
    int i, j, a;                       sscanf(argv[1], "%d", &l);
    a = 0;                              q = f2(l, vv);
    for (i = 2; i <= n; i++)            i = 0;
        for (j = i + 1; j <= n; j++)    while (i < q) {
            if (f1(i) == j && f1(j) == i) { printf("%d %d\n", vv[i], vv[i + 1]);
                v[a++] = i;              i += 2;
                v[a++] = j;              }
            }
    }                                     return 0;
                                     }
```

- Dire cosa viene calcolato dal programma complessivo.
- Tradurre la funzione in Assembler.

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

- (a) Un programma **cerca** con un argomento $file_p$ seguito da uno o più argomenti $file_1, \dots, file_n$. Il file $file_p$ deve contenere una lista di stringhe da cercare in ciascuno dei file $file_1, \dots, file_n$. Ogni linea del file $file_p$ è una stringa da cercare. Se $file_i$, per qualche i , non è un file regolare, il programma deve saltarlo e passare ad esaminare i successivi file, se ve ne sono. Se l'argomento $file_p$ è uguale a “-” le stringhe da cercare vanno lette, una per linea, dall'ingresso standard. Per ogni stringa trovata, il programma deve stampare sulla sua uscita standard il nome del file contenente la stringa seguito da “.” e l'intera linea, all'interno del file, che contiene la stringa.
- (b) Un programma **riferimenti** con un argomento dir . L'argomento dir deve essere un nome di directory. Il programma deve cercare, all'interno della directory dir , tutti i file che contengono uno o più nomi di file all'interno della stessa directory. Per svolgere il suo compito il programma deve creare due processi figli collegati tramite una pipe. Il primo figlio deve eseguire il programma “cerca” con argomenti opportuni, mentre il secondo figlio deve inviare al primo, tramite la pipe, le stringhe da cercare (i nomi dei file regolari contenuti nella directory dir).