

# Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

28 Giugno 2006

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

```
.text
.global f2
f2:   pushl %ebp
      movl  %esp, %ebp
      pushl %esi
      pushl %edi
      pushl %ecx
      movl  8(%ebp), %esi
      movl  %esi, %edi
L1:   cmpb  $0, 1(%edi)
      je   L2
      incl %edi
      jmp  L1
L2:   cmpl  %esi, %edi
      jle  L3
      movb (%esi), %cl
      movb (%edi), %ch
      movb %cl, (%edi)
      movb %ch, (%esi)
      incl %esi
      decl %edi
      jmp  L2
L3:   popl  %ecx
      popl %edi
      popl %esi
      leave
      ret
```

```
#include <stdio.h>
const int MAXL = 50;
void f1(char *aa, char *bb)
{
    int i = 0;
    while (aa[i] != 0 && aa[i] == bb[i]) i++;
    if (aa[i] == bb[i]) f2(aa);
}
int main(int argc, char *argv[])
{
    char a[MAXL], b[MAXL];
    b[0] = '\0';
    FILE *ff = fopen(argv[1], "r");
    while (fscanf(ff, "%s", a) == 1) {
        f1(a, b);
        printf("%s\n", a);
        strcpy(b, a);
    }
}
```

- (a) Dire cosa viene calcolato dal programma complessivo.
- (b) Tradurre la funzione f1 in Assembler.

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

- (a) Un programma **pensa**, che accetta un argomento da riga di comando. L'argomento deve essere un numero naturale  $m$ , maggiore di 0. Il programma sceglie un numero casuale compreso tra 1 e  $m$  (estremi inclusi), quindi legge ciclicamente il suo ingresso standard, fino a quando non viene inserito un numero uguale a quello scelto. Il programma termina, stampando un messaggio opportuno, se il numero viene indovinato, oppure se viene interrotto dalla signal **SIGINT** (generabile da terminale tramite la combinazione di tasti **Ctrl** e **c**). Nel secondo caso, il programma, prima di terminare, stampa il numero che aveva scelto.

**Suggerimenti:** per generare un numero (pseudo-)casuale compreso tra 1 e  $a$  si può usare la seguente espressione: `rand() % a + 1`.

- (b) Un programma **indovina**, che deve accettare due argomenti da riga di comando. Entrambi gli argomenti sono numeri interi maggiori di 0,  $n_1$  e  $n_2$ , con  $n_2 < n_1$ . Il programma deve creare un processo figlio e fargli eseguire il programma **pensa**, con argomento  $n_1$ . Quindi, comunicando con il processo figlio tramite una pipe, deve cercare di indovinare il numero pensato, inviando un nuovo numero ogni secondo. Se non è riuscito ad indovinare il numero dopo  $n_2$  tentativi, deve inviare la signal **SIGINT** al processo figlio, aspettare la terminazione del figlio e terminare lui stesso.

**Suggerimenti:** conviene che il padre sostituisca la sua uscita standard con l'ingresso della pipe, in modo da poter usare `printf` per inviare i numeri al figlio. Inoltre, conviene eseguire una `fflush` dopo ogni invio. Infine, per sapere se un processo è ancora vivo, si può controllare che `kill(pid, 0)` restituisca 0.