

# Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

20 Settembre 2007

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

<pre>.text .global f2 f2: pushl %ebp     movl %esp, %ebp     pushl 12(%ebp)     pushl 8(%ebp)     call f4     addl \$8, %esp     cmpl %eax, 8(%ebp)     jne 11     pushl 16(%ebp)</pre>	<pre>    push 8(%ebp)     call f4     addl \$8, %esp     jmp 12 11: pushl 16(%ebp)     pushl 12(%ebp)     call f4     addl \$8, %esp 12: leave     ret</pre>
<pre>#include &lt;stdio.h&gt;  int soglia = 10;  void f1(int *p1, int *p2, int *p3) {     scanf("%d%d%d", p1, p2, p3); }  void f3(int x) {     printf("%d\n", x); }  int f4(int u, int v) {     if (u &gt;= v)         return u;</pre>	<pre>    return v; }  void f5() {     int a, b, c, d;     do {         f1(&amp;a, &amp;b, &amp;c);         d = f2(a, b, c);         if (d &gt; soglia)             f3(d);     } while (d &gt;= 0); }  int main() {     f5(); }</pre>

- Dire cosa viene calcolato dal programma complessivo.
- Tradurre la funzioni f5 in Assembler.

2. Un messaggio di posta elettronica (semplificato) è un file di testo con una intestazione, seguita da una linea vuota e dal corpo del messaggio. L'intestazione è composta da varie linee della forma “*campo: valore*”. Per esempio, supponiamo di avere un file “`mail.txt`” contenente il seguente testo:

```
From: pippo@src.it
To: pluto@dst.it
```

```
Hi!
pippo
```

In questo esempio, l'intestazione è composta dalle prime due righe di testo, mentre il corpo dalle ultime due. Nell'intestazione, la prima riga (campo “From:”) contiene l'indirizzo del mittente del messaggio, mentre la seconda (campo “To:”) contiene l'indirizzo del destinatario.

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

- (a) Un programma `reply` con argomento *nome* da riga di comando. Si suppone che *nome* sia un file contenente un messaggio di posta elettronica. Il programma deve produrre sull'uscita standard un messaggio di riposta a quello contenuto in *nome*, nel seguente modo:
- L'intestazione deve avere gli stessi campi, con gli stessi valori, del messaggio contenuto in *nome*, ma con mittente e destinatario scambiati;
  - Il corpo deve contenere tutto il corpo di *nome*, con ogni linea preceduta dal carattere “>”, seguito da tutte le linee che il programma legge dal suo ingresso standard

Per esempio, eseguendo il comando “`reply mail.txt`” con ingresso standard “Bye”, otterremo il seguente output:

```
To: pippo@src.it
From: pluto@dst.it
```

```
>Hi!
>pippo
Bye
```

- (b) Un programma `autoreply` con un primo argomento *risposta* e uno o più argomenti *mail<sub>1</sub>*, *mail<sub>2</sub>*, ... Il programma deve creare un processo per ogni parametro *mail<sub>i</sub>* ricevuto. Il processo figlio *i*-esimo deve eseguire il programma `reply` con parametro *mail<sub>i</sub>*, ingresso standard rediretto dal file *risposta* (che deve esistere) e uscita standard rediretta sul file “`risp-maili`” (che, se non esiste, va creato). Infine, il programma deve attendere la terminazione di tutti i suoi figli e terminare esso stesso.