

# Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

31 Gennaio 2008

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

<pre>.text .global f1 f1:  pushl %ebp       movl %esp, %ebp       pushl %ebx       pushl %esi       movl \$0, %esi       movl 8(%ebp), %ebx indietro:       pushl (%ebx)       call strlen       cmpl %esi, %eax</pre>	<pre>      jle  avanti       movl %eax, %esi avanti:       addl \$4, %ebx       decl 12(%ebp)       cmpl \$0, 12(%ebp)       jne indietro       movl %esi, %eax       popl %esi       popl %ebx       leave       ret</pre>
<pre>#include &lt;string.h&gt; #include &lt;stdio.h&gt;  #define MAXP 100 #define MAXL 1000  char *vv[MAXP];  int f1(char *v[], int n);  int f2(char *s) {     int i = 0;     int c = 0;      vv[i++] = &amp;s[c];     while (s[c] != '\0') {         if (s[c] == ' ') {</pre>	<pre>            s[c] = '\0';             vv[i] = &amp;s[c + 1];             i++;         }         c++;     }     return f1(vv, i); }  int main() {     char buf[MAXL];     int n = read(0, buf, MAXL);     buf[n - 1] = '\0';     int r = f2(buf);     printf("%d\n", r);     return 0; }</pre>

- (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 `contalinee` senza argomenti. Il programma deve leggere nomi di file dal suo ingresso standard (uno per linea) e mostrare sull'uscita standard il numero di linee di ogni file corrispondente. Se il file corrispondente non esiste o non può essere aperto, il programma deve stampare un messaggio di errore e passare al prossimo nome di file. Il programma termina quando il suo ingresso standard termina.

Un programma `calc` con un argomento da riga di comando, che può assumere il valore “+” o “x”. Il programma legge numeri interi dal suo ingresso standard e li somma (caso “+”) o li moltiplica (caso “x”). Quando l'ingresso standard termina, il programma mostra il totale sull'uscita standard e termina.

- (b) Un programma `totale` con argomenti  $file_1, file_2, \dots$  (almeno un nome di file deve essere presente). Il programma mostra sull'uscita standard il numero di linee totali di tutti i file  $file_1, file_2, \dots$ . Per svolgere il suo compito, il programma deve creare tre processi collegati tramite pipe. Il primo processo invia al secondo i nomi di file, il secondo esegue il programma `contalinee` e invia il risultato al terzo, che esegue opportunamente il programma `calc`. Infine, il programma attende la terminazione dei tre processi e termina esso stesso.