

Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

2 luglio 2010

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 %esi pushl %ebx movl \$0, %esi movl 8(%ebp), %ebx c1: movb (%ebx), %al</pre>	<pre> cmpb \$0, %al je l1 incl %esi incl %ebx jmp c1 l1: movl %esi, %eax popl %ebx popl %esi leave ret</pre>
<pre>#include <stdio.h> const int MAXL = 100; int f1(const char* s); int f2(){ int m = 0; char buf[MAXL]; do { scanf("%s", buf); int k = f1(buf);</pre>	<pre> if(k>m) m = k; } while(strcmp(buf, ".") != 0); return m; } int main(){ int r = f2(); printf("%d\n", r); return 0; }</pre>

- Dire cosa viene calcolato dal programma complessivo.
- 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 `contanascosti` con uno o più argomenti da riga di comando. Ciascun argomento rappresenta il nome di una directory e il programma deve contare il numero totale di file nascosti contenuti in ciascuna di esse e stamparlo sull'uscita standard. Gli argomenti che non corrispondono a directory devono essere saltati.
- Chiamiamo *file nascosti* i file il cui nome comincia con il carattere '.', ma è diverso dalla stringa "." e dalla stringa "..".
- (b) Un programma `contanascosti-R` con un argomento da riga di comando. Il programma deve stampare il numero totale di file nascosti contenuti nella directory il cui nome è passato come argomento, e nelle sue sottodirectory di qualunque livello (si assuma un numero massimo di sottodirectory possibili e una dimensione massima per il percorso di ciascuna). Per svolgere il proprio compito, il programma deve creare due processi collegati tramite una pipe. Il primo processo deve eseguire il comando di sistema `find` e il secondo deve eseguire il comando `contanascosti`.
- Nota:** Il programma `find` con i tre argomenti `dir -type d`, dove *dir* è una directory, mostra sull'uscita standard *dir* e i nomi di tutte le sue sottodirectory, a qualsiasi livello.