

Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

11 febbraio 2011

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 %ecx pushl %esi movl \$0, %esi movl 8(%ebp), %ebx movl 12(%ebp), %ecx l1: cmpb \$0, (%ebx)</pre>	<pre>je 12 cmpb (%ebx), %cl jne 13 incl %esi 13: incl %ebx jmp 11 12: movl %esi, %eax popl %esi popl %ecx popl %ebx leave ret</pre>
<pre>#include <stdio.h> int f1(char* s, char c); int f2(char* v[], int n, char c) { int i; int j = 0; for(i=0; i<n; i++) { int q = f1(v[i], c); if(q > j) j = q; } }</pre>	<pre>return j; } int main(int argc, char* argv[]){ char** pp = argv; pp = pp + 2; char c = argv[1][0]; int k = f2(pp, argc-2, c); printf("%d\n", k); 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 `match` con argomenti `dir` e `schema` da riga di comando. Il programma deve mostrare sull'uscita standard tutti i nomi dei file contenuti nella directory `dir` e che rispettano lo `schema` dato. Nel confrontare un nome di file con lo `schema`, i seguenti caratteri eventualmente contenuti nello `schema` vanno interpretati in modo speciale:

- “?”: può corrispondere a qualunque altro carattere;
- “@”: può corrispondere soltanto a lettere, sia maiuscole che minuscole;
- “%”: può corrispondere soltanto alle cifre da “0” a “9”.

Tutti gli altri caratteri devono corrispondere esattamente. Per esempio: lo schema “`ci@o%?`” corrisponde ai nomi “`ciao0B`” e “`ciTo23`”, ma non ai nomi “`ci2o22`” (“@” non corrisponde a “2”), “`ciao0Bz`” (troppi caratteri) o “`ciao0`” (troppo pochi).

- (b) Un programma `list` con argomenti `dir` e `schema` da riga di comando. Il programma deve eseguire `ls -l` su tutti i file della directory `dir` il cui nome rispetti lo schema `schema`. Per svolgere il proprio compito il programma deve utilizzare il programma `match` per ottenere la lista dei nomi di file, che deve essere passata in ingresso al comando di sistema `xargs` con argomenti opportuni.

Nota: Il programma di sistema `xargs` legge dal suo ingresso parole separate da spazi (anche a capo) e le aggiunge in coda alla lista dei proprio argomenti, ottenendo così un comando che viene poi eseguito. Per esempio, scrivendo il comando

```
xargs rm -rf
```

e dandogli in ingresso le stringhe “`dir1`” e “`dir2`”, verranno cancellate le directory con quei nomi.