

Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

1 Febbraio 2007

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 %eax pushl %ecx movl \$0, %esi for: cmpl 16(%ebp), %esi jge avan movl %esi, %eax movl MAXL, %ecx imul %ecx</pre>	<pre> addl 12(%ebp), %eax pushl %eax pushl 8(%ebp) call f2 addl \$8, %esp incl %esi jmp for avan: popl %ecx popl %eax popl %esi leave ret</pre>
<pre>#include <unistd.h> #include <stdio.h> #include <stdlib.h> #include <sys/types.h> #include <sys/wait.h> void f2(char* a, char* b){ if(fork() == 0) execlp(a, a, b, NULL); else wait(0); }</pre>	<pre>const int MAXN = 1000; const int MAXL = 100; void f1(char a[], char b[MAXN][MAXL], int c); int main(int argc, char* argv[]) { int q = 0; char aa[MAXN][MAXL]; FILE* ff = fopen(argv[2], "r"); while(fscanf(ff, "%s", aa[q]) == 1) q++; fclose(ff); f1(argv[1], aa, q); 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 `split`, con due argomenti (da riga di comando) *linee* e *file*. Il programma deve controllare che *linee* sia un numero intero maggiore di zero e che *file* sia il nome di un file regolare esistente (altrimenti deve stampare un messaggio di errore ed uscire), quindi deve suddividere il contenuto del file *file* in tanti file più piccoli, ciascuno contenente al massimo *linee* linee. Il nome dei nuovi file deve avere la forma *file-numero*, dove *numero* rappresenta un numero progressivo a partire da 0. La suddivisione deve essere eseguita in modo tale che la concatenazione dei nuovi file nell'ordine dato da *numero* coincida con il contenuto del file di partenza. Si supponga, per semplicità, che esistano dei valori massimi, noti a-priori, per la lunghezza della stringa *file* e per la lunghezza massima di una linea in un file.
 - (b) Un programma `multisplit`, con un primo argomento *log*, seguito da un numero qualunque (maggiore di uno) di argomenti *file1*, *file2*, etc. Per ogni argomento *fileX* il programma deve lanciare un nuovo processo che esegua il programma `split` con quel nome di file come secondo argomento, e il numero 10 come primo. I processi devono poter essere eseguiti concorrentemente e il programma deve attendere la terminazione di tutti i suoi processi figli, prima di terminare esso stesso. Il primo argomento, *log*, deve essere usato nel seguente modo: se il programma riesce ad aprire correttamente il file di nome *log* (o, eventualmente, a crearlo se non esiste), allora deve fare in modo che tutti i processi figli accodino i loro messaggi di errore in quel file; altrimenti, i messaggi di errore devono poter essere visibili sullo *standard error*.