

Prova scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

9 giugno 2008

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

```
.text                                decl %ebx
.global f1                           pushl %ebx
f1:  pushl %ebp                        call f1
     movl %esp, %ebp                 addl $4, %esp
     pushl %ebx                      movl 8(%ebp), %ebx
     pushl %edx                      imul %ebx
     cmpl $1, 8(%ebp)                12:  popl %edx
     jne  11                          popl %ebx
     movl $1, %eax                   leave
     jmp  12                          ret
11:  movl 8(%ebp), %ebx
```

```
#include <stdio.h>
#define DIM 1000

int f1(int n);

void f2(int v[], int n)
{
    int i;
    int r;
    for (i = n - 1; i >= 0; i--) {
        r = f1(v[i]);
        printf("%d\n", r);
    }
}

int main()
{
    int x;
    int i = 0;
    int vv[DIM];
    while (1) {
        scanf("%d", &x);
        if (x == 0)
            break;
        vv[i++] = x;
    }
    f2(vv, i);
    return 0;
}
```

- (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 `conta` $file_1 file_2 \dots$, con uno o più nomi di file come argomento. Si suppone che ogni file sia un file di testo contenente numeri reali. Per ogni file $file_i$, il programma `conta` produce in uscita una linea

$file_i \quad num \quad somma$

contenente il nome del file ($file_i$), il numero totale di numeri reali letti nel file (num), e la somma di tali numeri ($somma$). I tre campi devono essere separati dal carattere di tabulazione (“\t”). Se un file non esiste, non è leggibile, o è una directory, il programma deve stampare un messaggio di errore e passare al file successivo (se ve ne sono).

Un programma `media`, senza argomenti. Il programma legge dal suo ingresso linee che devono avere il formato

$num \quad somma$

Dove num è un numero intero, e $somma$ è un numero reale. Il programma somma tutti i num letti (sia $numtot$ la somma) e tutti i valori $somma$ letti (sia $sommatot$ la somma), quindi stampa in uscita il valore $sommatot/numtot$.

- (b) Un programma `mediatot` $file_1 file_2 \dots$, con uno o più nomi di file come argomento. Si suppone che ogni file sia un file di testo contenente numeri reali. Il programma deve mostrare in uscita il valore medio di tutti i numeri letti. Per svolgere il suo compito, il programma deve creare tre processi figli, collegati tramite due pipe. Il primo figlio esegue il programma `conta`, con argomenti $file_1 file_2 \dots$, e invia i risultati al secondo figlio. Il secondo figlio esegue opportunamente il programma di sistema `cut` e invia i risultati al terzo. Il terzo esegue il programma `media`.

Nota: il programma `cut` invia in uscita una parte di ciascuna linea che legge in ingresso. Più in dettaglio, legge linee dal suo ingresso e le separa in parole in base al carattere di tabulazione. Quindi, stampa solo le parole che gli sono state richieste tramite gli argomenti da riga di comando. Per esempio, supponiamo di invocare il comando `cut` nel seguente modo:

`cut -f 1,3`

e di inviare in ingresso al comando le seguenti linee (si intende che le parole siano separate da “\t”):

```
aaa      bbbb      cccc
dddddd   eee       fff
gg       hhhhhh    ii
```

Allora, il programma produrrà in uscita solo la prima e la terza parola di ogni riga:

```
aaa      cccc
dddddd   fff
gg       ii
```