

Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

31 gennaio 2008

1. (a) La funzione `f1` riceve in ingresso un vettore di stringhe e il numero di stringhe contenute nel vettore stesso. Calcola il massimo della lunghezza delle stringhe e lo restituisce. La funzione `f2` riceve in ingresso una stringa e la suddivide in parole (separate da un singolo spazio); riempie il vettore `vv` in modo che l'elemento i -esimo punti alla parola i -esima, quindi passa `vv` alla funzione `f1`. La funzione `main` legge un vettore di `MAXL` caratteri dall'ingresso standard (se l'ingresso standard è un terminale, legge una linea), lo trasforma in stringa, lo passa a `f2`, quindi stampa il valore restituito. Complessivamente, il programma mostra in uscita la lunghezza massima delle parole inserite in ingresso.

- (b) Una possibile traduzione è la seguente:

```
.extern vv                                cmpb  '$' ',(%ebx, %edi)
.text                                     jne  avanti2
.global f2                                movb  $0, (%ebx, %edi)
f2:                                       movl  %edi, %ecx
    pushl %ebp                               incl  %ecx
    movl  %esp, %ebp                         leal  (%ebx, %ecx), %edx
    # uso esi come i e edi come c           movl  %edx, (%eax, %esi, 4)
    # 8(%ebp) e' s                          incl  %esi
    pushl %ebx                               avanti2:
    pushl %ecx                               incl  %edi
    pushl %edx                               jmp   ciclo
    pushl %edi
    pushl %esi                               fine:
                                           pushl %esi
    movl  $0, %esi                           pushl $vv
    movl  $0, %edi                           call  f1
    movl  $vv, %eax                          addl  $8, %esp
                                           popl  %esi
    movl  8(%ebp), %ebx                       popl  %edi
    leal  (%ebx, %edi), %edx                   popl  %edx
    movl  %edx, (%eax, %esi, 4)               popl  %ecx
    incl  %esi                               popl  %ebx
                                           leave
ciclo:                                     je    fine
    cmpb  $0, (%ebx, %edi)
    je    fine
                                           ret
```

2. (a) `#include <stdio.h>`
`#include <string.h>`

`#define MAXLINE 1024`

```

int main()
{
    char buf[MAXLINE];
    FILE *in;
    int dim, conta;

    while ( fgets(buf, MAXLINE, stdin) ) {
        dim = strlen(buf);
        if (dim <= 1)
            continue;
        buf[dim - 1] = '\0';
        if ( !(in = fopen(buf, "r")) ) {
            perror(buf);
            continue;
        }
        conta = 0;
        while ( fgets(buf, MAXLINE, in) )
            conta++;
        printf("%d\n", conta);
    }
}

#include <stdio.h>

int main(int argc, char *argv[])
{
    int tot, num;
    if (argc != 2 || strlen(argv[1]) != 1 || argv[1][0] != '+' && argv[1][0] != 'x') {
        fprintf(stderr, "Uso: %s +|x\n", argv[0]);
        exit(1);
    }
    tot = (argv[1][0] == '+') ? 0 : 1;
    while (scanf("%d", &num) != EOF) {
        switch (argv[1][0]) {
            case '+':
                tot += num;
                break;
            case 'x':
                tot *= num;
                break;
        }
    }
    printf("%d\n", tot);
}

(b) #include <stdio.h>

int main(int argc, char* argv[])
{
    int pf1[2], pf2[2], i;

    if (argc < 2) {
        fprintf(stderr, "Uso: %s file1 file2...\n", argv[0]);
        exit(1);
    }
}

```

```

}

if (pipe(pf1) < 0) {
    perror(argv[0]);
    exit(1);
}
switch (fork()) {
case -1:
    perror(argv[0]);
    exit(1);
case 0:
    close(1);
    dup(pf1[1]);
    close(pf1[0]);
    close(pf1[1]);
    for (i = 1; i < argc; i++)
        printf("%s\n", argv[i]);
    exit(0);
default:
    break;
}
if (pipe(pf2) < 0) {
    perror(argv[0]);
    exit(1);
}
switch (fork()) {
case -1:
    perror(argv[0]);
    exit(1);
case 0:
    close(0);
    dup(pf1[0]);
    close(1);
    dup(pf2[1]);
    close(pf1[0]);
    close(pf1[1]);
    close(pf2[0]);
    close(pf2[1]);
    execl("contalinee", "contalinee", NULL);
    perror("contalinee"),
    exit(1);
default:
    break;
}
close(pf1[0]);
close(pf1[1]);
switch (fork()) {
case -1:
    perror(argv[0]);
    exit(1);
case 0:
    close(0);
    dup(pf2[0]);

```

```
        close(pf2[0]);
        close(pf2[1]);
        execl("calc", "calc", "+", NULL);
        perror("calc");
        exit(1);
default:
        break;
}
close(pf2[0]);
close(pf2[1]);
for (i = 0; i < 3; i++)
        wait(NULL);
return 0;
}
```