

# Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

18 luglio 2008

1. (a) La funzione `f2` restituisce il valore  $x + y$ ,  $x - y$ ,  $x \times y$  o  $x \div y$  (tra numeri naturali), a seconda che il valore di `c` sia "s", "d", "x" o "v", rispettivamente. Se `c` ha un valore diverso da questi, `f2` restituisce  $x$ . La funzione `f1` applica l'operazione specificata da `c` ai due vettori `a` e `b`, elemento per elemento, e ne stampa il risultato, purché i due vettori abbiano la stessa dimensione, altrimenti non fa niente.
- (b) Una possibile traduzione è la seguente:

```
.data                                pushl 24(%ebp)
formato: .asciz "%d\n"                pushl (%edi, %esi, 4)
.text                                  pushl (%ebx, %esi, 4)
.global f1                             call f2
f1:  pushl %ebp                         addl $12, %esp
      movl %esp, %ebp                  pushl %eax
      pushl %eax                       pushl $formato
      pushl %ebx                       call printf
      pushl %edi                       addl $8, %esp
      pushl %esi                       incl %esi
      movl 20(%ebp), %ebx              jmp for
      cmpl 12(%ebp), %ebx              fine: popl %esi
      jne fine                         popl %edi
      movl 8(%ebp), %ebx               popl %ebx
      movl 16(%ebp), %edi             popl %eax
      movl $0, %esi                   leave
for:  cmpl 20(%ebp), %esi              ret
      jge fine
```

2. (a) 

```
#include <stdio.h>
#include <string.h>

struct intervallo {
    char primo;
    char ultimo;
    int conta;
};

#define MAX_STATS 10

struct intervallo stats[MAX_STATS];

int main(int argc, char* argv[])
{
    int i;
```

```

char c;

if (argc < 2) {
    fprintf(stderr, "Uso: %s <intervallo> ...\n", argv[0]);
    exit(1);
}
if (argc - 1 > MAX_STATS) {
    fprintf(stderr, "troppi argomenti (max = %d)\n", MAX_STATS);
    exit(1);
}

for (i = 1; i < argc; i++) {
    if (strlen(argv[i]) != 3 ||
        sscanf(argv[i], "%c-%c", &stats[i-1].primo, &stats[i-1].ultimo) != 2)
    {
        fprintf(stderr, "l'argomento '%s' non ha la forma c-c\n", argv[i]);
        exit(1);
    }
    stats[i].conta = 0;
}

while ( (c = getchar()) != EOF ) {
    for (i = 0; i < argc - 1; i++)
        if (c >= stats[i].primo && c <= stats[i].ultimo)
            stats[i].conta++;
}

for (i = 0; i < argc - 1; i++)
    printf("%s: %d\n", argv[i+1], stats[i].conta);

return 0;
}
(b) #include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>

int main(int argc, char* argv[])
{
    int nproc, i, quanti;
    int fin, fd[2];
    char c;

    if (argc < 5) {
        fprintf(stderr, "Uso: %s <nproc> <intervallo> <file_in> <file_out\n", argv[0]);
        exit(1);
    }

    if (sscanf(argv[1], "%d", &nproc) != 1 || nproc <= 0) {
        fprintf(stderr, "il primo argomento deve essere un numero maggiore di zero\n");
        exit(1);
    }
}

```

```

if ( (fin = open(argv[3], O_RDONLY)) < 0) {
    perror(argv[3]);
    exit(1);
}

if (pipe(fd) < 0) {
    perror(argv[0]);
    exit(1);
}

quanti = 0;
for (i = 0; i < nproc; i++) {
    switch (fork()) {
        case -1:
            perror(argv[0]);
            break;
        case 0:
            close(0);
            dup(fd[0]);
            close(fd[0]);
            close(fd[1]);
            close(1);
            if (open(argv[4], O_WRONLY | O_CREAT | O_APPEND, 0666) < 0) {
                perror(argv[3]);
                exit(1);
            }
            execl("classifica", "classifica", argv[2], NULL);
            perror("classifica");
            exit(1);
        default:
            quanti++;
            break;
    }
}
close(fd[0]);
if (quanti == 0) {
    fprintf(stderr, "impossibile creare processi\n");
    exit(1);
}

while (read(fin, &c, 1) == 1) {
    write(fd[1], &c, 1);
}
close(fd[1]);

while (quanti) {
    wait(NULL);
    quanti--;
}

return 0;
}

```