

Soluzioni della Prova Scritta di Sistemi di Elaborazione Ingegneria delle Telecomunicazioni

Ing. G. Lettieri, Ing. A. Vecchio

20 Luglio 2007

1. (a) La funzione `f1` riceve come argomento un puntatore a caratteri. Se il carattere puntato è una lettera minuscola, restituisce il numero d'ordine della lettera ('a' ha il numero d'ordine 0, 'b' il numero d'ordine 1 e così via), altrimenti restituisce `-1`. La funzione `f2` incrementa il valore del vettore `vv` corrispondente al carattere iniziale della stringa `p`, se questo è una lettera minuscola, altrimenti non fa niente. La funzione `f3` stampa sull'uscita standard il contenuto del vettore `ww`, usando la funzione `f4` per la stampa vera e propria. Complessivamente, il programma riceve un numero qualunque di argomenti da riga di comando e mostra sull'uscita standard una colonna in cui la riga `i`-esima corrisponde alla lettera minuscola `i`-esima e mostra il numero di argomenti che iniziano con quella lettera.

- (b) Una possibile traduzione è la seguente:

```
.text                                f3:    pushl %ebp
.global f2                            movl   %esp, %ebp
.global f3                            pushl  %ebx
f2:    pushl %ebp                        pushl  %esi
      movl  %esp, %ebp                movl   8(%ebp), %ebx
      pushl %eax                       movl   $0, %esi
      pushl %ebx                       ciclo_for:
      pushl 8(%ebp)                    cmpl  $26, %esi
      call  f1                          jge   fine2
      addl $4, %esp                     pushl (%ebx, %esi, 4)
      cmpl $-1, %eax                   call  f4
      je   fine                         addl $4, %esp
      movl 12(%ebp), %ebx               incl  %esi
      incl (%ebx, %eax, 4)              jmp   ciclo_for
fine:  popl  %ebx                       fine2: popl  %esi
      popl  %eax                       popl  %ebx
      leave                                  leave
      ret                                  ret
```

2. (a) `#include <sys/types.h>`
`#include <sys/stat.h>`
`#include <fcntl.h>`
`#include <stdio.h>`
`#include <errno.h>`
`#include <dirent.h>`

`#define MAXPOSTI 10`
`#define BUFSIZE 1024`
`#define MAXTEMPO 10`

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

```

{
    DIR *d;
    int posto = 0;
    char buf[BUFSIZE];
    int fd;
    int tempo;

    if (argc != 4) {
        fprintf(stderr, "Uso: %s <dir> <nome> <tempo>\n", argv[0]);
        exit(1);
    }

    if (sscanf(argv[3], "%d", &tempo) != 1 || tempo < 0) {
        fprintf(stderr, "argomento non valido\n");
        exit(1);
    }

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

    while (posto < MAXPOSTI) {
        sprintf(buf, "%d", posto);
        fd = open(buf, O_WRONLY|O_CREAT|O_EXCL, 0666);
        if (fd >= 0) {
            write(fd, argv[2], strlen(argv[2]));
            sleep(tempo);
            if (unlink(buf) < 0) {
                perror(buf);
                exit(1);
            }
            exit(0);
        }
        if (errno != EEXIST) {
            perror(buf);
            exit(1);
        }
        posto++;
    }

    fprintf(stderr, "ND\t%s\n", argv[2]);
    return 2;
}

```

```

(b) #include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

```

```

#define BUFSIZE 100
#define MAXTEMPO 10

int main(int argc, char* argv[])
{
    int fd, i, quanti;
    char buf[BUFSIZE];
    int tempo;

    if (argc < 2) {
        fprintf(stderr, "Uso: %s <dir> <nome1>...\n", argv[0]);
        exit(1);
    }

    if ( (fd = open("log.txt", O_WRONLY|O_CREAT|O_APPEND, 0666)) < 0) {
        perror("log.txt");
        exit(1);
    }

    srand(time(0));
    quanti = 0;
    for (i = 2; i < argc; i++) {
        tempo = rand() % MAXTEMPO;
        switch (fork()) {
            case -1:
                perror(argv[0]);
                break;
            case 0:
                close(2);
                dup(fd);
                close(fd);
                sprintf(buf, "%d", tempo);
                execl("occupa", "occupa", argv[1], argv[i], buf, NULL);
                perror("occupa");
                exit(1);
            default:
                quanti++;
                sleep(1);
                break;
        }
    }
    while (quanti > 0) {
        wait(0);
        quanti--;
    }
    return 0;
}

```