

Lezione 12

- Si scriva un programma che conti le modifiche ad un file (specificato come primo argomento sulla riga di comando) nell'arco di un intervallo di tempo (specificato in secondi come secondo argomento sulla linea di comando). Alla fine il programma deve produrre sullo schermo del terminale un istogramma che mostri il numero di modifiche (si utilizzi ad esempio il carattere *).

```
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>

main(int argc, char **argv) {
    int sec, i, mod=0;
    struct stat buf;
    time_t last_mtime;

    if(argc!=3) {
        fprintf(stderr, "Uso: mod <file> <secondi>\n");
        exit(1);
    }

    if(stat(argv[1], &buf) == -1) {
        perror("Errore nella chiamata a stat");
        exit(2);
    }

    last_mtime = buf.st_mtime;
    sec = atoi(argv[2]);

    for(i=0; i<sec; i++) {

        if(stat(argv[1], &buf) == -1) {
            perror("Errore nella chiamata a stat");
            exit(2);
        }

        if(buf.st_mtime != last_mtime) {
            mod++;
            last_mtime = buf.st_mtime;
        }

        sleep(1);
    }

    if(mod == 0)
        printf("\nNegli ultimi %s secondi il file '%s' "
              "non e' stato modificato.\n", argv[2], argv[1]);
    else
```

```

    printf("\nModifiche effettuate al file '%s' "
           "negli ultimi %s secondi:\n",argv[1],argv[2]);

    for(i=0;i<mod;i++)
        putchar('*');
        putchar('\n');

    exit(0);
}

```

- Si scriva un programma C che realizza una versione semplificata del comando unix `find`. Il programma dovrà ricevere sulla linea di comando il nome di una directory *dir* ed una stringa *str* e dovrà visitare l'intero albero di directory e file che ha come radice *dir*, stampando su std output tutti i file i cui nomi hanno come suffisso la stringa *str*, segnalando se si tratta di directory o file ordinari.

```

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

char *src_string;

int strrindex(char s[],char t[]);
int src(const char *name,const struct stat *sptr,int type);

main(int argc,char **argv) {

    if(argc!=3) {
        fprintf(stderr,"Utilizzo: myfind <dir> <stringa>\n");
        exit(1);
    }

    src_string=argv[2];

    if(ftw(argv[1],src,5)==-1) {
        perror("Errore nell'esecuzione di ftw");
        exit(2);
    }

}

/* strrindex restituisce l'indice dell'occorrenza piu'
 * a destra di t in s, -1 se invece t non occorre in s
 */
int strrindex(char s[],char t[]) {
    int i=0,j,index=-1;

    if(strlen(s)<strlen(t))
        return -1;

```

```

while(s[i]!='\0') {

    while(s[i]!=t[0] && s[i]!='\0')
        i++;

    j=1;

    while(s[i+j]==t[j] && s[i+j]!='\0' && t[j]!='\0')
        j++;

    if(t[j]=='\0')
        index=i;

    i++;
}

return index;
}

int src(const char *name,const struct stat *sptr,int type) {
    int index;

    if(type==FTW_F || type==FTW_D) {
        index=strrindex(name,src_string);

        /* con il comando if seguente si controlla che l'occorrenza
        * piu' a destra di t in s si trovi effettivamente alla fine
        * della stringa s (i.e., sia un suffisso)
        */
        if(index!=-1 && index+strlen(src_string)==strlen(name)) {
            printf("%s ",name);

            if(type==FTW_F)
                printf("(file ordinario)\n");
            else
                printf("(directory)\n");

        }

    }

    return 0;
}

```