

Laboratorio de Ficheros

Andrés Loreto Quiros



Pontificia Universidad
JAVERIANA
Colombia

Sistemas Operativos

Prof. John Corredor

Lab01

The screenshot shows a terminal window with the nano 6.2 editor open. The file is named lab01.c. The code is a C program that reads a directory name from the user, opens it, and prints the names and sizes of all files it contains. It uses standard library functions like opendir, readdir, and stat.

```
GNU nano 6.2 lab01.c
/*
 * Autor: Andres Loreto Quiros
 * Fecha: 11 de Noviembre 2025
 * Tema: Ficheros
 * =====
 *
 * Resumen
 * Programa que lee de teclado el nombre de un directorio
 * y muestra en pantalla el nombre y el tamaño de los
 * ficheros que contiene.
 */
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>

int main() {
    DIR *d;
    char nomdir[90], nomfich[90];
    struct stat datos;
    struct dirent *direc;

    printf("Introduzca el nombre de un directorio: ");
    fgets(nomdir, sizeof(nomdir), stdin);
    nomdir[strlen(nomdir) - 1] = '\0'; // Eliminar salto de línea

    if ((d = opendir(nomdir)) == NULL) {
        printf("El directorio no existe\n");
        return -1;
    }

    while ((direc = readdir(d)) != NULL) {
        strcpy(nomfich, nomdir);
        strcat(nomfich, "/");
        strcat(nomfich, direc->d_name);

        if (stat(nomfich, &datos) == 0 && S_ISREG(datos.st_mode)) {
            printf("Nombre: %s Tamaño: %ld bytes\n", direc->d_name, datos.st_size);
        }
    }

    closedir(d);
    return 0;
}

^G Help      ^O Write Out    ^W Where Is     ^X Cut          ^T Execute      ^J Justify      [ Read 51 lines ]
^X Exit      ^R Read File    ^A Replace      ^P Paste         ^C Location     M-U Undo      M-A Set Mark   M-Q To Bracket M-O Previous  ^B Back
^M Write      ^I Insert       ^V Paste        ^N New          ^G Go To Line   M-B Redo      M-C Copy      M-Q Where Was   M-W Next     ^F Forward
```

Lab01 le pide al usuario el nombre de un directorio y muestra por pantalla los nombres y tamaños de todos los archivos que contiene. Usa funciones del sistema como opendir, readdir y stat para acceder a la información de cada fichero y determinar su tamaño en bytes, recorriendo todo el contenido del directorio indicado.

Resultados Lab01:

```
[estudiante@NGEN265:~/Ficheros$ make run1
./lab01
[Introduzca el nombre de un directorio: Folder
Nombre: ejemplo.c      Tamaño: 162 bytes
[estudiante@NGEN265:~/Ficheros$ make run1
./lab01
[Introduzca el nombre de un directorio: Folder2
Nombre: ejemplo2.c     Tamaño: 162 bytes
estudiante@NGEN265:~/Ficheros$ ]
```

El programa ejecuta correctamente la lectura de los directorios y muestra los nombres de los archivos junto con su tamaño en bytes, comprobando que ambos directorios (Folder y Folder2) contienen un solo archivo de igual tamaño.

Lab02

```
GNU nano 6.2                                     lab02.c
/*
 * Autor: Andres Loreto Quiros
 * Fecha: 11 de Noviembre 2025
 * Tema: Ficheros
 * =====
 * Resumen:
 * El programa recibe el nombre de un directorio y muestra los nombres de los archivos y subdirectorios que contiene, indicando sus permisos, si el propietario tiene permiso de lectura
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <dirent.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>

#define TEN_DAYS (10*24*60*60)
#define BUF 2048

static void chop(char *s) {
    size_t n = strlen(s);
    if (n && (s[n-1] == '\n' || s[n-1] == '\r')) s[n-1] = '\0';
}

int main(void) {
    char nomdir[BUF];
    DIR *d;
    struct dirent *ent;

    printf("Nombre de directorio: ");
    if (!fgets(nomdir, sizeof(nomdir), stdin)) {
        fprintf(stderr, "Error leyendo entrada\n");
        return 1;
    }
    chop(nomdir);

    d = opendir(nomdir);
    if (!d) {
        perror("No existe ese directorio / no se puede abrir");
        return 1;
    }

    time_t ahora = time(NULL);
    while ((ent = readdir(d)) != NULL) {
        /* omitir . y .. */
        if (strcmp(ent->d_name, ".") == 0 || strcmp(ent->d_name, "..") == 0)
            continue;

        char ruta[BUF];
        sprintf(ruta, "%s/%s", nomdir, ent->d_name);

        struct stat st;
        if (lstat(ruta, &st) != 0) { /* lstat para no seguir symlinks */
            perror("stat");
            continue;
        }

        /* modo en octal y lectura para propietario */
        unsigned modo = st.st_mode & 07777;
        int owner_read = (st.st_mode & S_IRUSR) ? 1 : 0;
        int es_dir = S_ISDIR(st.st_mode);

        printf("Entrada: %s | modo: %04o | R propietario: %s | %s\n",
               ent->d_name,
               modo,
               owner_read ? "SI" : "NO",
               es_dir ? "Directorio" : "Fichero");

        /* si es fichero y fue modificado en los últimos 10 días, imprimir fecha de acceso */
        if (S_ISREG(st.st_mode) && (difftime(ahora, st.st_atime) <= TEN_DAYS)) {
            char *fa = ctime(&st.st_atime); /* fecha de último acceso */
            if (fa) {
                /* ctime incluye salto de línea al final */
                printf(" -> Modificado <10d | Último acceso: %s", fa);
            }
        }
    }
    closedir(d);
    return 0;
}

^G Help      ^O Write Out   ^W Where Is   ^X Cut          ^T Execute   ^C Location   N-U Undo   N-A Set Mark   N-B To Bracket   N-Q Previous   ^B Back
^X Exit     ^R Read File   ^A Replace   ^U Paste         ^J Justify   ^V Go To Line  N-E Redo   N-C Copy       N-Q Where Was   N-W Next    ^F Forward
```

```
GNU nano 6.2                                     lab02.c
if (!fgets(nomdir, sizeof(nomdir), stdin)) {
    fprintf(stderr, "Error leyendo entrada\n");
    return 1;
}
chop(nomdir);

d = opendir(nomdir);
if (!d) {
    perror("No existe ese directorio / no se puede abrir");
    return 1;
}

time_t ahora = time(NULL);
while ((ent = readdir(d)) != NULL) {
    /* omitir . y .. */
    if (strcmp(ent->d_name, ".") == 0 || strcmp(ent->d_name, "..") == 0)
        continue;

    char ruta[BUF];
    snprintf(ruta, sizeof(ruta), "%s/%s", nomdir, ent->d_name);

    struct stat st;
    if (lstat(ruta, &st) != 0) { /* lstat para no seguir symlinks */
        perror("stat");
        continue;
    }

    /* modo en octal y lectura para propietario */
    unsigned modo = st.st_mode & 07777;
    int owner_read = (st.st_mode & S_IRUSR) ? 1 : 0;
    int es_dir = S_ISDIR(st.st_mode);

    printf("Entrada: %s | modo: %04o | R propietario: %s | %s\n",
           ent->d_name,
           modo,
           owner_read ? "SI" : "NO",
           es_dir ? "Directorio" : "Fichero");

    /* si es fichero y fue modificado en los últimos 10 días, imprimir fecha de acceso */
    if (S_ISREG(st.st_mode) && (difftime(ahora, st.st_atime) <= TEN_DAYS)) {
        char *fa = ctime(&st.st_atime); /* fecha de último acceso */
        if (fa) {
            /* ctime incluye salto de línea al final */
            printf(" -> Modificado <10d | Último acceso: %s", fa);
        }
    }
}
closedir(d);
return 0;
}

^G Help      ^O Write Out   ^W Where Is   ^X Cut          ^T Execute   ^C Location   N-U Undo   N-A Set Mark   N-B To Bracket   N-Q Previous   ^B Back
^X Exit     ^R Read File   ^A Replace   ^U Paste         ^J Justify   ^V Go To Line  N-E Redo   N-C Copy       N-Q Where Was   N-W Next    ^F Forward
```

El programa solicita el nombre de un directorio y muestra para cada archivo o subdirectorio su modo (permisos), si el propietario tiene permiso de lectura y si es un directorio o un fichero. Además, identifica los archivos modificados en los últimos 10 días y muestra su

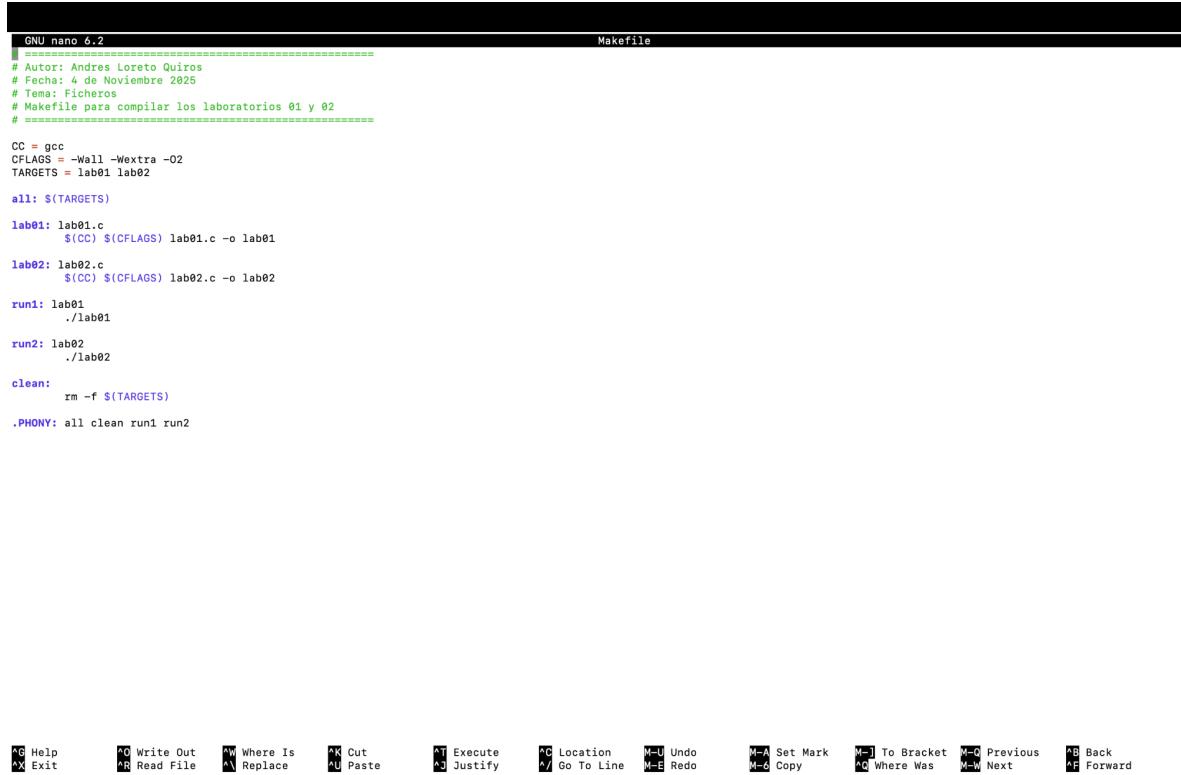
fecha de acceso. Utiliza las estructuras stat y dirent junto con funciones como time y difftime para comparar fechas y verificar permisos.

Resultados Lab02:

```
estudiante@NGEN265:~/Ficheros$ make run2
./lab02
Nombre de directorio: Folder
entrada: ejemplo.c | modo: 0664 | R propietario: SI | Fichero
-> Modificado <10d | Último acceso: Tue Nov 11 11:04:09 2025
estudiante@NGEN265:~/Ficheros$ make run2
./lab02
Nombre de directorio: Folder2
entrada: ejemplo2.c | modo: 0664 | R propietario: SI | Fichero
-> Modificado <10d | Último acceso: Tue Nov 11 11:12:27 2025
estudiante@NGEN265:~/Ficheros$
```

El programa amplía la funcionalidad mostrando los permisos, si el propietario tiene permiso de lectura, y la fecha de último acceso de los archivos modificados recientemente. Los resultados demuestran que los archivos tienen permisos 0664, el propietario cuenta con lectura habilitada y ambos fueron modificados dentro de los últimos 10 días.

Makefile



```
GNU nano 6.2
=====
# Autor: Andres Loreto Quiros
# Fecha: 4 de Noviembre 2025
# Tema: Ficheros
# Makefile para compilar los laboratorios 01 y 02
# =====

CC = gcc
CFLAGS = -Wall -Wextra -O2
TARGETS = lab01 lab02

all: $(TARGETS)

lab01: lab01.c
    $(CC) $(CFLAGS) lab01.c -o lab01

lab02: lab02.c
    $(CC) $(CFLAGS) lab02.c -o lab02

run1: lab01
    ./lab01

run2: lab02
    ./lab02

clean:
    rm -f $(TARGETS)

.PHONY: all clean run1 run2
```

El Makefile automatiza la compilación de los programas lab01.c y lab02.c. Define las reglas para compilar cada uno con gcc, optimizando el proceso con banderas de advertencia (-Wall -Wextra). También incluye comandos para ejecutar los programas, que son Run1 y Run2 que son para Lab01 y Lab02 respectivamente.

```
[estudiante@NGEN265:~/Ficheros$ ls
Folder Folder2 lab01 lab01.c lab02 lab02.c Makefile
[estudiante@NGEN265:~/Ficheros$ nano lab01.c
[estudiante@NGEN265:~/Ficheros$ nano lab02.c
[estudiante@NGEN265:~/Ficheros$ make lab01
make: 'lab01' is up to date.
[estudiante@NGEN265:~/Ficheros$ make lab02
make: 'lab02' is up to date.
estudiante@NGEN265:~/Ficheros$ ]
```

```
[estudiante@NGEN265:~/Ficheros$ make run1
./lab01
[Introduzca el nombre de un directorio: Folder
Nombre: ejemplo.c Tamaño: 162 bytes
[estudiante@NGEN265:~/Ficheros$ make run2
./lab02
[Nombre de directorio: Folder2
entrada: ejemplo2.c | modo: 0664 | R propietario: SI | Fichero
-> Modificado <10d | Último acceso: Tue Nov 11 11:12:27 2025
estudiante@NGEN265:~/Ficheros$ ]
```

Link Repositorio:

<https://github.com/Andresloretoq/Labs-Ficheros.git>