



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y
CIENCIAS SOCIALES Y ADMINISTRATIVAS



Práctica 9

Manejo de archivos

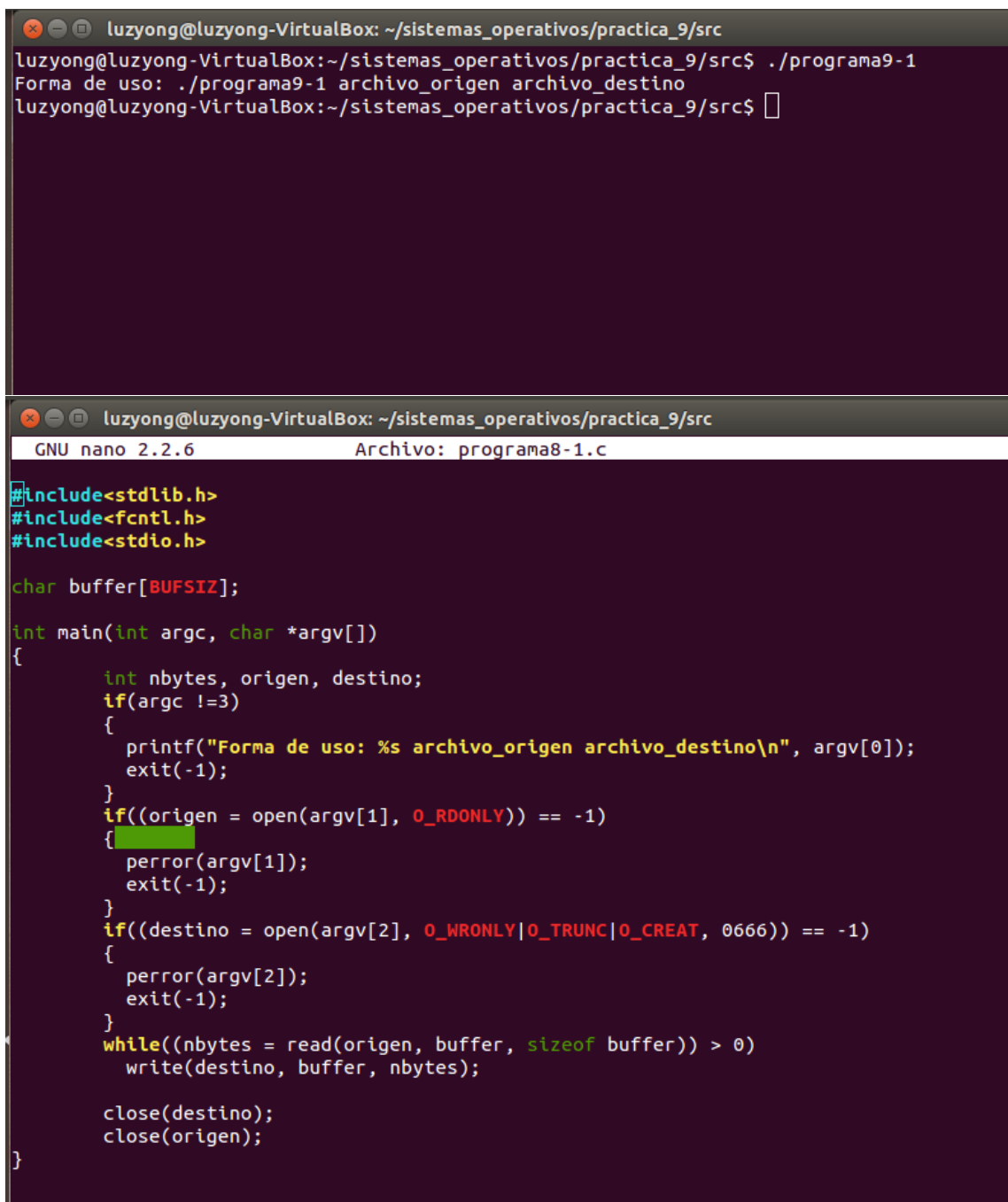
Yong Rodríguez Luz María

El manejo de archivos de Linux incluye la reservación de una dirección de memoria que guarda la información parcial o total que contiene un documento. Esto permite acceder a dicho documento de forma más fácil que buscándolo dentro del disco duro.

A esta característica se le conoce como proyección de archivos.

Programa 9.1

Este programa copia un archivo a otro destino con las funciones `open()`, `read()`, `write()`. Es equivalente a `cp`.



```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_9/src
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_9/src$ ./programa9-1
Forma de uso: ./programa9-1 archivo_origen archivo_destino
luzyong@luzyong-VirtualBox:~/sistemas_operativos/practica_9/src$

GNU nano 2.2.6 Archivo: programa8-1.c
#include<stdlib.h>
#include<fcntl.h>
#include<stdio.h>

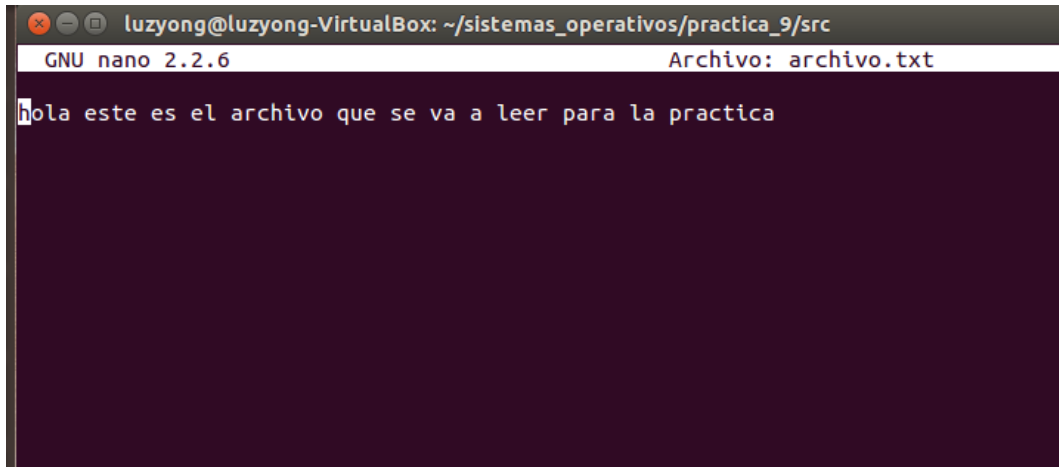
char buffer[BUFSIZ];

int main(int argc, char *argv[])
{
    int nbytes, origen, destino;
    if(argc !=3)
    {
        printf("Forma de uso: %s archivo_origen archivo_destino\n", argv[0]);
        exit(-1);
    }
    if((origen = open(argv[1], O_RDONLY)) == -1)
    {
        perror(argv[1]);
        exit(-1);
    }
    if((destino = open(argv[2], O_WRONLY|O_TRUNC|O_CREAT, 0666)) == -1)
    {
        perror(argv[2]);
        exit(-1);
    }
    while((nbytes = read(origen, buffer, sizeof buffer)) > 0)
        write(destino, buffer, nbytes);

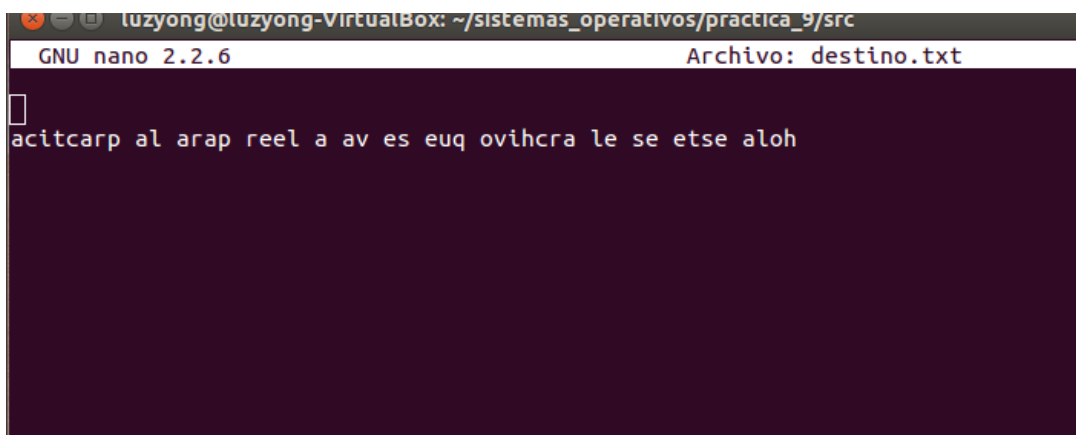
    close(destino);
    close(origen);
}
```

Ejercicio 9-1

Mediante el uso de lseek elabore un programa que invierta los bytes de un archivo, de modo que el último byte sea el primero y el primero el último.



```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_9/src
GNU nano 2.2.6 Archivo: archivo.txt
hola este es el archivo que se va a leer para la practica
```



```
luzyong@luzyong-VirtualBox: ~/sistemas_operativos/practica_9/src
GNU nano 2.2.6 Archivo: destino.txt
acitcarp al arap reel a av es euq ovihcra le se etse aloh
```

```
luzyong@luzyong-VirtualBox: ~/sisistemas_operativos/practica_9/src
:)
56se leyeron 1 bytes La palabra leida es : a :)
55se leyeron 1 bytes La palabra leida es : c :)
54se leyeron 1 bytes La palabra leida es : i :)
53se leyeron 1 bytes La palabra leida es : t :)
52se leyeron 1 bytes La palabra leida es : c :)
51se leyeron 1 bytes La palabra leida es : a :)
50se leyeron 1 bytes La palabra leida es : r :)
49se leyeron 1 bytes La palabra leida es : p :)
48se leyeron 1 bytes La palabra leida es : :)
47se leyeron 1 bytes La palabra leida es : a :)
46se leyeron 1 bytes La palabra leida es : l :)
45se leyeron 1 bytes La palabra leida es : :)
44se leyeron 1 bytes La palabra leida es : a :)
43se leyeron 1 bytes La palabra leida es : r :)
42se leyeron 1 bytes La palabra leida es : a :)
41se leyeron 1 bytes La palabra leida es : p :)
40se leyeron 1 bytes La palabra leida es : :)
39se leyeron 1 bytes La palabra leida es : r :)
38se leyeron 1 bytes La palabra leida es : e :)
37se leyeron 1 bytes La palabra leida es : e :)
36se leyeron 1 bytes La palabra leida es : l :)
35se leyeron 1 bytes La palabra leida es : :)
34se leyeron 1 bytes La palabra leida es : a :)
33se leyeron 1 bytes La palabra leida es : :)
32se leyeron 1 bytes La palabra leida es : a :)
31se leyeron 1 bytes La palabra leida es : v :)
30se leyeron 1 bytes La palabra leida es : :)
29se leyeron 1 bytes La palabra leida es : e :)
28se leyeron 1 bytes La palabra leida es : s :)
27se leyeron 1 bytes La palabra leida es : :)
26se leyeron 1 bytes La palabra leida es : e :)
25se leyeron 1 bytes La palabra leida es : u :)
24se leyeron 1 bytes La palabra leida es : q :)
23se leyeron 1 bytes La palabra leida es : :)
22se leyeron 1 bytes La palabra leida es : o :)

GNU nano 2.2.6 Archivo: ejercicio9-1.c
#include<stdlib.h>
#include<fcntl.h>
#include<stdio.h>
#include<string.h>

char buffer[BUFSIZ];

int main(int argc, char *argv[])
{
    int nbytes, origen, destino, j;
    if((origen = open("archivo.txt", O_RDONLY)) == -1)
    {
        perror("Error");
        exit(-1);
    }
    if((destino = open("destino.txt", O_RDWR|O_CREAT|O_APPEND, 0777)) == -1)
    {
        perror("Error");
        exit(-1);
    }
    lseek(origen, 0, SEEK_END); //posiciona el puntero en el byte final del archivo
    while(j!=0) //Mientras j no sea 0, va a seguir leyendo
    {
        j=lseek(origen, -1, SEEK_CUR); //j es igual a la posición del puntero, aquí siempre desde la posición actual, va a retroceder 1
        printf("%d", j);
        nbytes=read(origen, &buffer, 1); //Solo va a almacenar en el buffer 1 byte según la posición
        printf("se leyeron %d bytes La palabra leida es : %s :)\n", nbytes, buffer);
        write(destino, &buffer, nbytes);
        lseek(origen, j, SEEK_SET); //posiciona el puntero en la posición j, que es la última que se configuró, así podemos ir avanzando del 59 S
    }
}
```

Conclusiones.

En esta práctica aprendí a usar las funciones de manejo de archivos dentro de Linux. Comprendí el funcionamiento de `open()`, `write()`, `read()`, `lseek()`, `creat()` y pude conectarlo con un ejercicio práctico que me permite pensar en otras aplicaciones para las funciones.