



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martínez Quintana

Asignatura: Laboratorio de Fundamentos de Programación

Grupo: 3

No de Práctica(s): 13

Integrante(s): Cecilia Torres Bravo

*No. de Equipo de
cómputo empleado:* N/A

No. de Lista o Brigada: 54

Semestre: 1º

Fecha de entrega: 20 de enero del 2021

Observaciones:

CALIFICACIÓN: _____

Lectura y escritura de datos

Objetivo

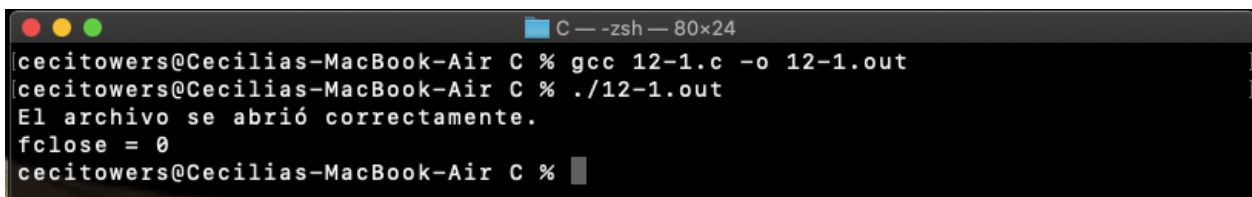
Elaborar programas en lenguaje C que requieran el uso de archivos de texto plano en la resolución de problemas, entendiendo a los archivos como un elemento de almacenamiento secundario.

Introducción

Todos los programas realizados cuentan con datos estructurados dentro de registros, almacenados sistemáticamente. En el lenguaje C, los apuntadores de archivo se manejan como variables apuntador de tipo FILE. Por otro lado, también podemos abrir y cerrar archivos con las funciones `fopen()` y `fclose()`. Sin embargo, hay que tomar en cuenta que un error en el cierre puede conllevar varios problemas.

Actividades

- A través de programas en C, emplear las funciones para crear, leer, escribir y sobrescribir archivos de texto plano.

A screenshot of a macOS terminal window titled "C — zsh — 80x24". The terminal shows the following commands and output:

```
cecitowers@Ceciliass-MacBook-Air C % gcc 12-1.c -o 12-1.out
cecitowers@Ceciliass-MacBook-Air C % ./12-1.out
El archivo se abrió correctamente.
fclose = 0
cecitowers@Ceciliass-MacBook-Air C %
```

```

1  #include<stdio.h>
2
3  /*
4  Este programa permite abrir un archivo en modo de lectura, de ser posible.
5  */
6
7  int main()
8  {
9      FILE *archivo;
10     archivo = fopen("fp_p13.pdf", "r");
11     if (archivo != NULL)
12     {
13         printf("El archivo se abrió correctamente.\n");
14         int res = fclose(archivo);
15         printf("fclose = %d\n", res);
16     } else
17     {
18         printf("Error al abrir el archivo.\n");
19         printf("El archivo no existe o no se tienen permisos de lectura.\n");
20     }
21     return 0;
22 }

```

```

cecitowers@Ceciliias-MacBook-Air C % gcc 12-1.c -o 12-1.out
cecitowers@Ceciliias-MacBook-Air C % ./12-1.out
El archivo se abrió correctamente.
Contenido del archivo:
Hola, este es el archivo de prueba%
cecitowers@Ceciliias-MacBook-Air C %

```

```

1  #include<stdio.h>
2
3  /*
4  Este programa permite lee el contenido de un archivo, de ser posible, a
5  través de la función fgets.
6  */
7
8  int main()
9  {
10     FILE *archivo;
11     char caracteres[50];
12     archivo = fopen("prueba.txt", "r");
13     if (archivo != NULL)
14     {
15         printf("El archivo se abrió correctamente.");
16         printf("\nContenido del archivo:\n");
17         while (feof(archivo) == 0)
18         {
19             fgets (caracteres, 50, archivo);
20             printf("%s", caracteres);
21         }
22         fclose(archivo);
23     }
24     return 0;
25 }
26

```

```
C — zsh — 80x24
cecitowers@Cecilias-MacBook-Air C % gcc 12-1.c -o 12-1.out
cecitowers@Cecilias-MacBook-Air C % ./12-1.out
El archivo se abrió correctamente.
cecitowers@Cecilias-MacBook-Air C %
```

```
prueba.txt
1 |
2 | Escribir cadena en archivo mediante fputs.
3 | Facultad de Ingeniería.
4 |
```

```
1 #include<stdio.h>
2
3 /*
4 Este programa permite escribir una cadena dentro de un archivo, de ser
5 posible, a través de la función fputs.
6 */
7
8 int main()
9 {
10 FILE *archivo;
11 char escribir[] = "\nEscribir cadena en archivo mediante fputs. \n\tFacultad de Ingeniería.\n";
12 archivo = fopen("prueba.txt", "r+");
13 if (archivo != NULL)
14 {
15     printf("El archivo se abrió correctamente.\n");
16     fputs (escribir, archivo);
17     fclose(archivo);
18 } else
19 {
20     printf("Error al abrir el archivo.\n");
21     printf("El archivo no existe o no se tienen permisos de lectura.\n");
22 }
23 return 0;
24 }
25
```

```
C — zsh — 80x24
cecitowers@Cecilias-MacBook-Air C % gcc 12-1.c -o 12-1.out
cecitowers@Cecilias-MacBook-Air C % ./12-1.out
Escribir
cadena
en
archivo
mediante
fputs.
Facultad
de
Ingeniería.
Ingeniería.
cecitowers@Cecilias-MacBook-Air C %
```

```
1  #include<stdio.h>
2
3  /*
4   Este programa permite leer el contenido de un archivo, de ser posible,
5   a través de la función fscanf.
6  */
7
8  int main()
9  {
10     FILE *archivo;
11     char caracteres[50];
12     archivo = fopen("prueba.txt", "r");
13     if (archivo != NULL)
14     {
15         while (feof(archivo)==0)
16         {
17             fscanf(archivo, "%s", caracteres);
18             printf("%s\n", caracteres);
19         }
20         fclose(archivo);
21     }
22     else
23     {
24         printf("El archivo no existe.\n");
25     }
26     return 0;
27 }
```

- Manipular archivos empleando los diferentes tipos de acceso a ellos.

```
≡ prueba.txt
1  |Escribir cadena en archivo mediante fprintf.
2  |Facultad de Ingeniería.
3  |UNAM
4  |
```

```
1  #include<stdio.h>
2
3  /*
4  Este programa permite escribir dentro de un archivo,
5  de ser posible, a través de la función fprintf.
6  */
7
8  int main()
9  {
10     FILE *archivo;
11     char escribir[] = "Escribir cadena en archivo mediante fprintf. \nFacultad de Ingeniería.\n";
12     archivo = fopen("prueba.txt", "r+");
13     if (archivo != NULL)
14     {
15         fprintf(archivo, "%s", escribir);
16         fprintf(archivo, "%s", "UNAM\n");
17         fclose(archivo);
18     }
19     else
20     {
21         printf("El archivo no existe o no se tiene permisos de lectura/escritura.\n");
22     }
23     return 0;
24 }
```

```
C — -zsh — 80x24
cecitowers@Ceciliias-MacBook-Air C % gcc 12-1.c -o 12-1.out
cecitowers@Ceciliias-MacBook-Air C % ./12-1.out prueba.txt
Escribir cadena en archivo mediante fprintf.
Facultad de Ingeniería.
UNAM
cecitowers@Ceciliias-MacBook-Air C %
```

```

1  #include <stdio.h>
2
3  /*
4  Este programa muestra el contenido de un archivo de texto. El
5  nombre del archivo se recibe como argumento de la función principal.
6  */
7
8  int main(int txt, char **argv)
9  {
10     FILE *ap;
11     unsigned char buffer[2048];
12     // Buffer de 2 Kbytes
13     int bytesLeidos;
14     // Si no se ejecuta el programa correctamente
15     if(txt < 2)
16     {
17         printf("Ejecutar el programa de la manera siguiente:\n\tnombre\tprograma nombre_archivo\n")
18         return 1;
19     }
20     // Se abre el archivo de entrada en modo lectura y binario
21     ap = fopen(argv[1], "rb");
22     if(!ap)
23     {
24         printf("El archivo %s no existe o no se puede abrir", argv[1]);
25         return 1;
26     }
27     while((bytesLeidos = fread(buffer, 1, 2048, ap)))
28         printf("%s", buffer);
29     fclose(ap); return 0;
30     return 0;
31 }

```

≡ prueba.txt

```

1  Escribir cadena en archivo mediante fprintf.
2  Facultad de Ingeniería.
3  UNAM
4  |

```

≡ nuevo.txt

```

1  Escribir cadena en archivo mediante fprintf.
2  Facultad de Ingeniería.
3  UNAM
4

```

```
C — zsh — 80x24
cecitowers@Cecilias-MacBook-Air C % gcc 12-1.c -o 12-1.out
cecitowers@Cecilias-MacBook-Air C % ./12-1.out prueba.txt nuevo.txt
cecitowers@Cecilias-MacBook-Air C %
```

```
1  #include <stdio.h>
2
3  /*
4  Este programa realizar una copia exacta de dos archivos. Los nombres de los archivos
5  (origen y destino) se reciben como argumentos de la función principal.
6  */
7
8  int main(int argc, char **argv)
9  {
10     FILE *archEntrada, *archivoSalida;
11     unsigned char buffer[2048];
12     // Buffer de 2 Kbytes
13     int bytesLeidos;
14     // Si no se ejecuta el programa correctamente
15     if(argc < 3)
16     {
17         printf("Ejecutar el programa de la siguiente manera:\n");
18         printf("\tnombre_programa \tarchivo_origen \tarchivo_destino\n");
19         return 1;
20     }
21     // Se abre el archivo de entrada en modo de lectura y binario
22     archEntrada = fopen(argv[1], "rb");
23     if(!archEntrada)
24     {
25         printf("El archivo %s no existe o no se puede abrir", argv[1]);
26         return 1;
27     }
28     // Se crea o sobrescribe el archivo de salida en modo binario
29     archivoSalida = fopen(argv[2], "wb");
30     if(!archivoSalida)
31     {
32         printf("El archivo %s no puede ser creado", argv[2]);
33         return 1;
34     }
35     // Copia archivos
36     while ((bytesLeidos = fread(buffer, 1, 2048, archEntrada)))
37         fwrite(buffer, 1, bytesLeidos, archivoSalida);
38     // Cerrar archivos
39     fclose(archEntrada);
40     fclose(archivoSalida);
41     return 0;
42 }
```


Conclusión

Para terminar, el realizar esta práctica me costó más trabajo que las anteriores, pues no lograba entender completamente el tema y cómo funcionaba cada función y modo de apertura de archivo, sin embargo, después de ir trabajando uno por uno logré entenderlo mucho mejor. Aun así, repasaré el tema y haré más ejercicios por mi cuenta para que todas mis dudas queden claras y no dude tanto a la hora de ver cómo funciona el código o cómo programarlo.

Fuentes de Consulta

Facultad de Ingeniería, (2018). *Guía práctica de estudio 13: Lectura y escritura de datos*. Recuperado el 20 de enero del 2021.