# Programa 10

## Linux

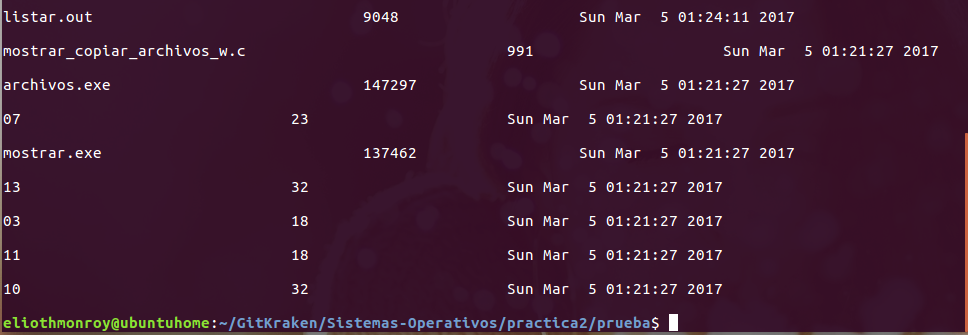
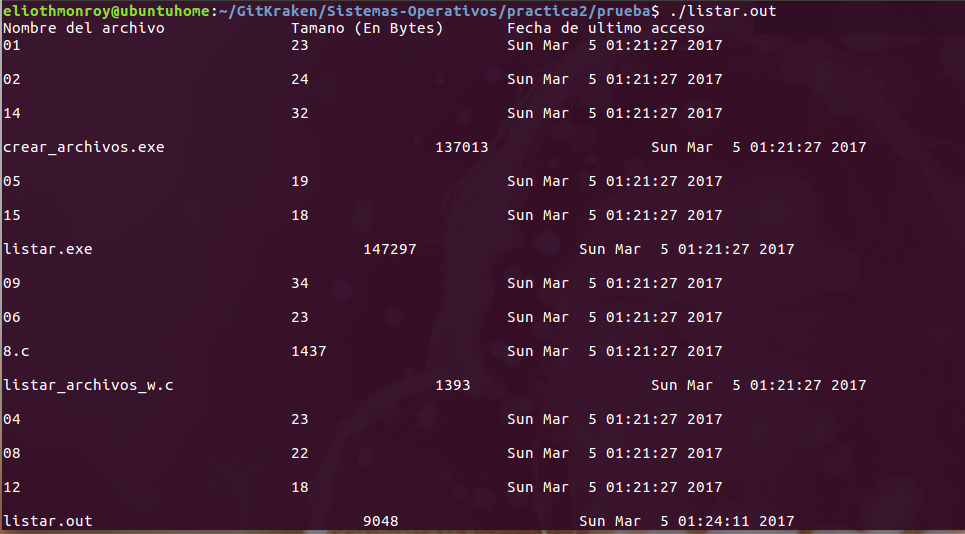
### Código

|  |
| --- |
| *//Librerías a usar*  *//Links revisados:*  *//http://pubs.opengroup.org/onlinepubs/009695399/functions/opendir.html*  *//http://pubs.opengroup.org/onlinepubs/009695399/basedefs/sys/stat.h.html*  *//http://pubs.opengroup.org/onlinepubs/009695399/functions/stat.html*  #include <sys/stat.h>  #include <dirent.h>  #include <time.h>  #include <stdio.h>  #include <string.h>  **int** main(**int** argc, **char** **const** \*argv[])  {  *//Las estructuras a usar*  **struct** stat estatus;  **struct** dirent \*archivo;  **DIR** \*directorio;  **struct** tm \*tiempo;  **char** dir[10]="./";  *//Abrimos el directorio actual*  directorio=opendir(dir);  printf("Nombre del archivo**\t\t**Tamano (En Bytes)**\t**Fecha de ultimo acceso**\n**" );  *//Ciclo que nos permite revisar cada uno de los archivos o directorios*  **while**((archivo=readdir(directorio))!=NULL){  *//Esta condicional la usamos para evitar que nos aparezcan*  *//"." y ".." como archivos*  **if**(strcmp(archivo->d\_name,".") && (strcmp(archivo->d\_name,".."))){  *//Primero imprimimos el nombre del archivo*  printf("%s**\t\t\t\t**", archivo->d\_name);  *//Ahora vemos cual es su estatus*  stat(archivo->d\_name, &estatus);  *//Por último imprimimos su tamaño y fecha de ultimo acceso*  printf("%ld **\t\t\t**", estatus.st\_size);  printf("%s**\n**", ctime(&estatus.st\_atime));  }  }  **return** 0;  } |

### Compilación



### Ejecución

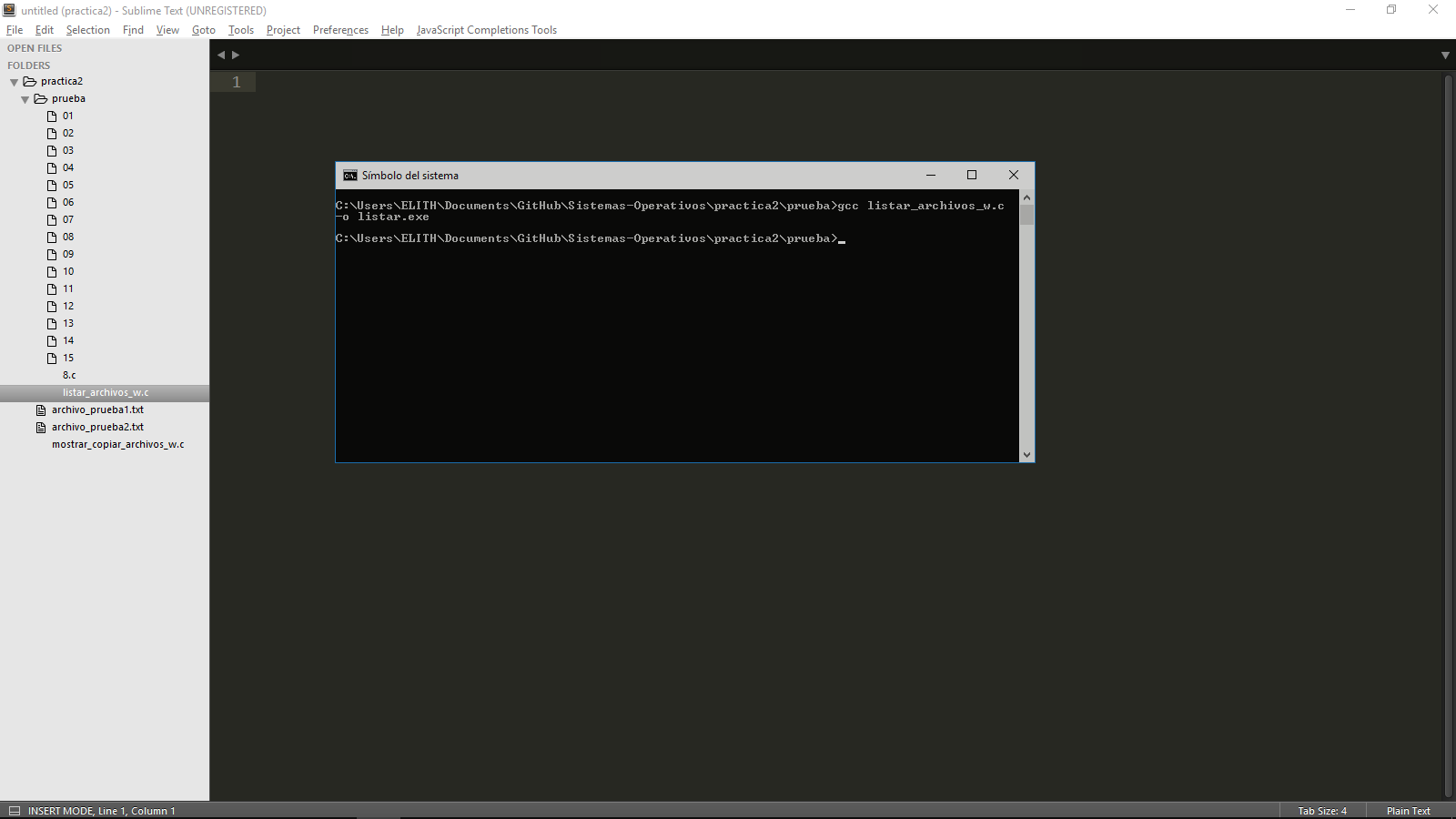


## Windows

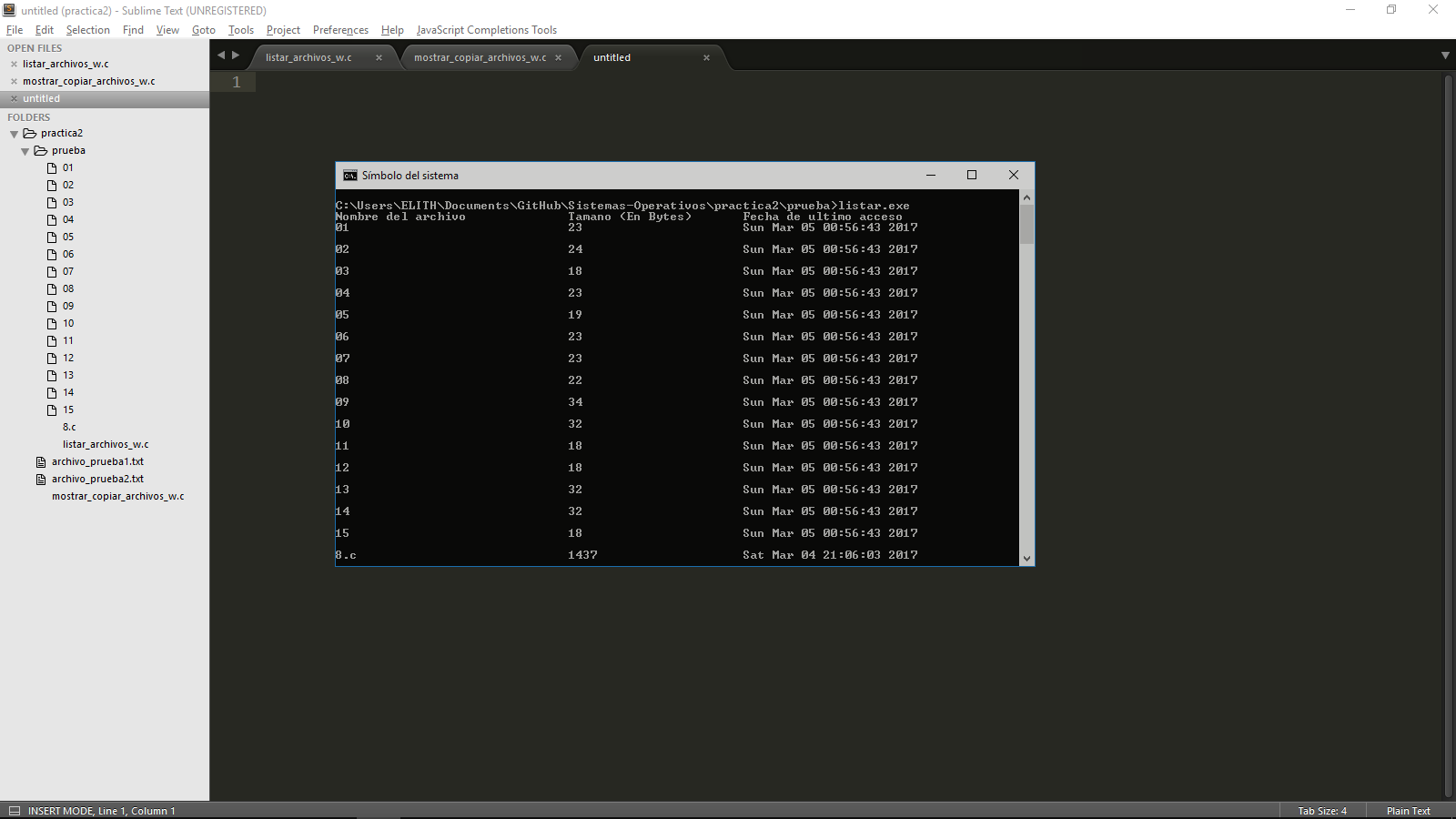
### Código

|  |
| --- |
| #include <sys/stat.h>  #include <dirent.h>  #include <time.h>  #include <stdio.h>  #include <string.h>  **int** main(**int** argc, **char** **const** \*argv[])  {  *//Las estructuras a usar*  **struct** stat estatus;  **struct** dirent \*archivo;  **DIR** \*directorio;  **struct** tm \*tiempo;  **char** dir[10]="./";  *//Abrimos el directorio actual*  directorio=opendir(dir);  printf("Nombre del archivo**\t\t**Tamano (En Bytes)**\t**Fecha de ultimo acceso**\n**" );  *//Ciclo que nos permite revisar cada uno de los archivos o directorios*  **while**((archivo=readdir(directorio))!=NULL){  *//Esta condicional la usamos para evitar que nos aparezcan*  *//"." y ".." como archivos*  **if**(strcmp(archivo->d\_name,".") && (strcmp(archivo->d\_name,".."))){  *//Primero imprimimos el nombre del archivo*  printf("%s**\t\t\t\t**", archivo->d\_name);  *//Ahora vemos cual es su estatus*  stat(archivo->d\_name, &estatus);  *//Por último imprimimos su tamaño y fecha de ultimo acceso*  printf("%ld **\t\t\t**", estatus.st\_size);  printf("%s**\n**", ctime(&estatus.st\_atime));  }  }  **return** 0;  } |

### Compilación



### Ejecución



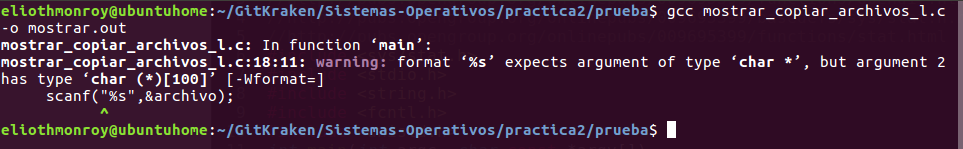
# Programa 11

## Linux

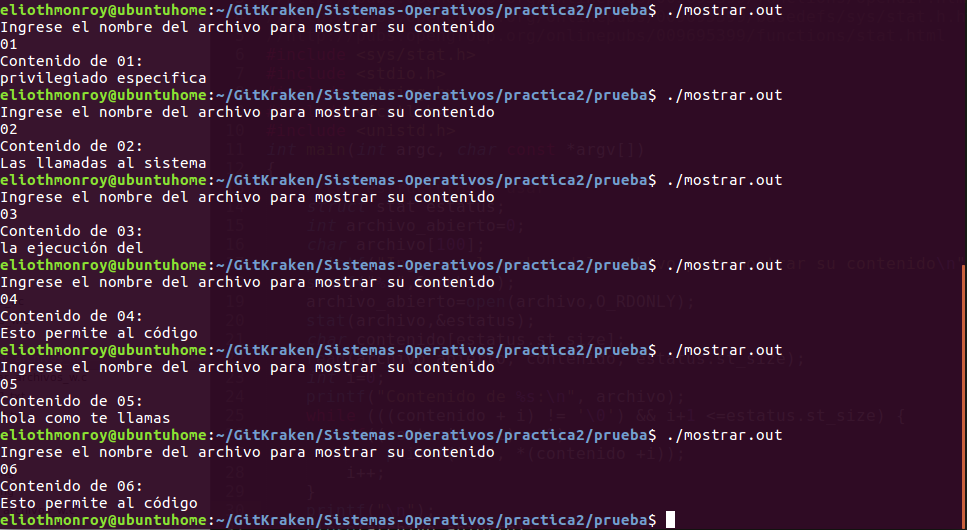
### Código

|  |
| --- |
| #include <sys/stat.h>  #include <stdio.h>  #include <string.h>  #include <fcntl.h>  #include <unistd.h>  **int** main(**int** argc, **char** **const** \*argv[])  {  *//Las estructuras a usar*  **struct** stat estatus;  **int** archivo\_abierto=0;  **char** archivo[100];  printf("Ingrese el nombre del archivo para mostrar su contenido**\n**");  scanf("%s",&archivo);  archivo\_abierto=open(archivo,O\_RDONLY);  stat(archivo,&estatus);  **char** contenido[estatus.st\_size];  read(archivo\_abierto, contenido, estatus.st\_size);  **int** i=0;  printf("Contenido de %s:**\n**", archivo);  **while** (((contenido + i) != '\0') && i+1 <=estatus.st\_size) {  **if**((contenido +i) > 0)  printf("%c", \*(contenido +i));  i++;  }  close(archivo\_abierto);  **return** 0;  } |

### Compilación



### Ejecución

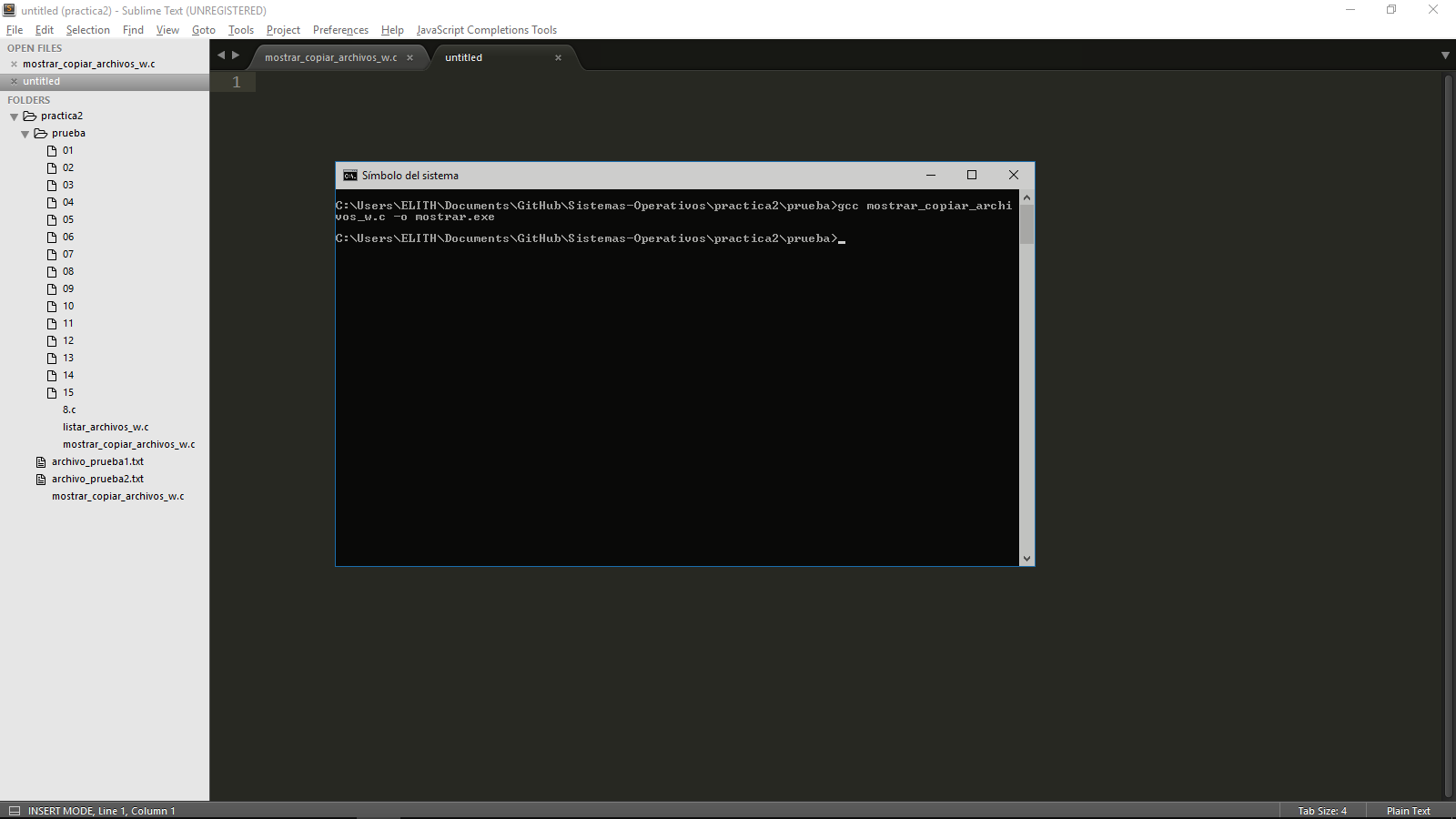


## Windows

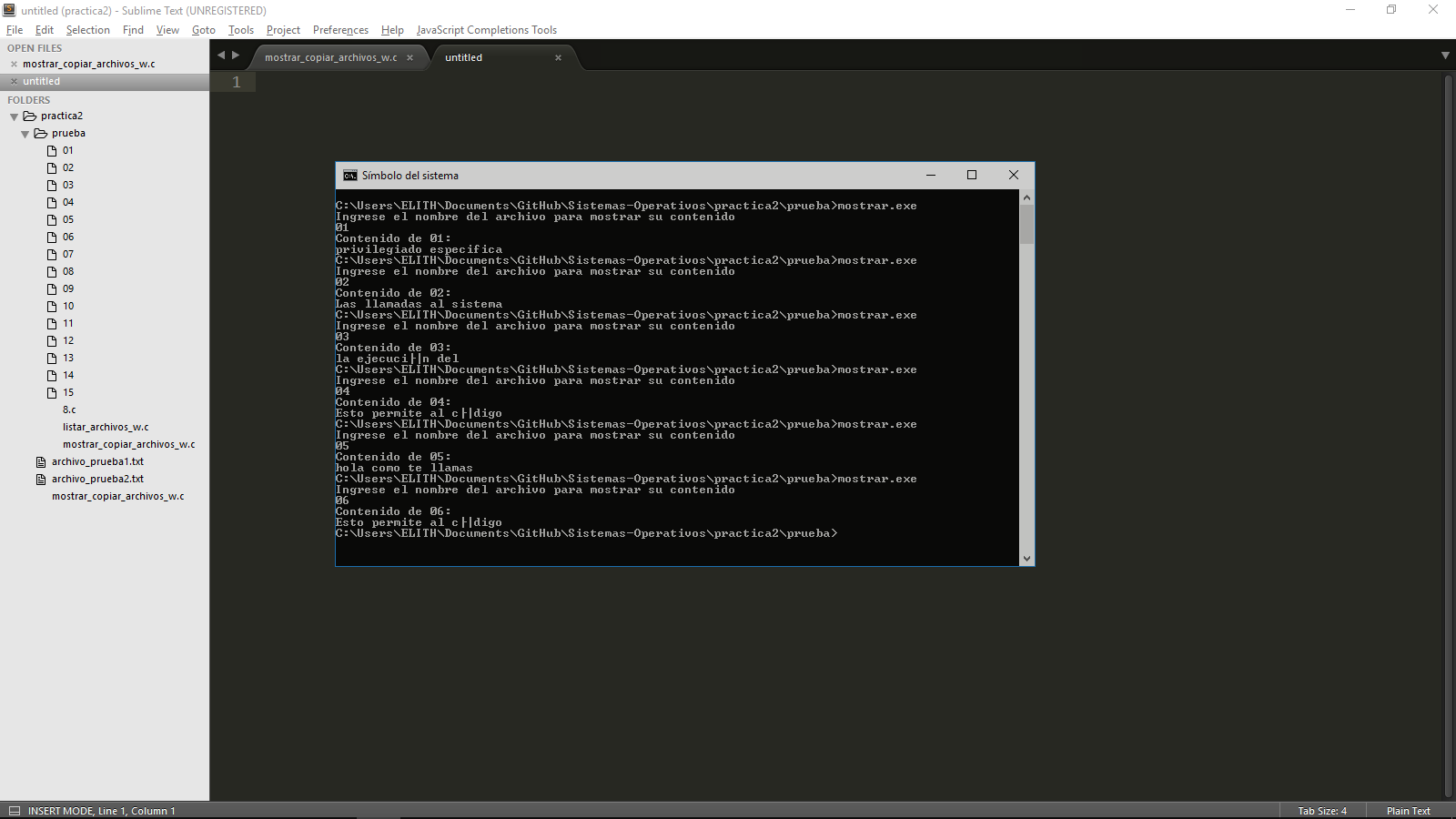
### Código

|  |
| --- |
| *//Librerías a usar*  *//Links revisados:*  *//http://pubs.opengroup.org/onlinepubs/009695399/functions/opendir.html*  *//http://pubs.opengroup.org/onlinepubs/009695399/basedefs/sys/stat.h.html*  *//http://pubs.opengroup.org/onlinepubs/009695399/functions/stat.html*  #include <sys/stat.h>  #include <stdio.h>  #include <string.h>  #include <fcntl.h>  #include <unistd.h>  **int** main(**int** argc, **char** **const** \*argv[])  {  *//Las estructuras a usar*  **struct** stat estatus;  **int** archivo\_abierto=0;  **char** archivo[100];  printf("Ingrese el nombre del archivo para mostrar su contenido**\n**");  scanf("%s",&archivo);  archivo\_abierto=open(archivo,O\_RDONLY);  stat(archivo,&estatus);  **char** contenido[estatus.st\_size];  read(archivo\_abierto, contenido, estatus.st\_size);  **int** i=0;  printf("Contenido de %s:**\n**", archivo);  **while** (((contenido + i) != '\0') && i+1 <=estatus.st\_size) {  **if**((contenido +i) > 0)  printf("%c", \*(contenido +i));  i++;  }  close(archivo\_abierto);  **return** 0;  } |

### Compilación



### Ejecución



Comentarios:

Está práctica resulto un tanto complicada debido a que tuvimos que usar llamadas al sistema para poder trabajar con los ficheros, y esto requirió buscar bastante información para poder realizar los programas. En especial con la parte de abrir un archivo para posteriormente mostrar su contenido, debido a que primero tenemos que saber dónde está el archivo, posteriormente abrirlo, y luego trabajar con un ciclo while para poder mostrar carácter por carácter lo que el archivo contiene, situación que se vuelve un tanto complicada. Así mismo, al leer el archivo, ese necesario conocer su tamaño en bytes para poderlo leer correctamente, esto genera otra situación, en la cual se vuelve necesario el tener que revisar el estatus (a través de stat) del archivo con el fin de conocer su tamaño para posteriormente usar este valor en el método read.