



**INSTITUTO POLITECNICO
NACIONAL**

Escuela Superior de Computo

INSTITUTO POLITÉCNICO NACIONAL



**PRACTICA 1 “Mini
Compilador”**

ALUMNO: González Barrientos Giovanni Daniel

BOLETA: 2020630148

CARRERA: Ingeniería en Sistemas Computacionales

GRUPO: 1CV3

PROF: Flores Mendoza Yaxkin

MATERIA: Estructuras De Datos

Fecha: 10 de Noviembre del 2020

INTRODUCCION

Para esta práctica, el objetivo era crear un programa que simulara la función de un compilador en c, donde a partir de un conjunto de archivos de prueba, cada uno siendo el código fuente de un programa sencillo, se lograra analizar y comprobar si las llaves utilizadas (Sean las llaves "{ }", corchetes "[]" o paréntesis "()") estaban bien colocadas. Si es el caso de que están bien implementadas las llaves, el programa mostraría un mensaje confirmando que estas estaban balanceadas, de lo contrario, el programa mostraría cualquiera de los 3 errores planteados, mostrando la línea y columna aproximada donde se presentó el error. Los posibles errores son los siguientes:

1. Error 1: Falta colocar la llave de cierre "}".
2. Error 2: Falta colocar la llave de apertura "{".
3. Error 3: Las llaves no son compatibles al momento de implementarlas, es decir, se intentó cerrar "(" o "[" con una llave diferente a su tipo.

Este programa podría ir dirigido a aquellos programadores que quieran comprobar si sus códigos tienen en este caso, una buena implementación de llaves en sus códigos. Solo sería cuestión de reconfigurar el programa para ajustarlo a esas necesidades más adelante.

FUNCIONAMIENTO:

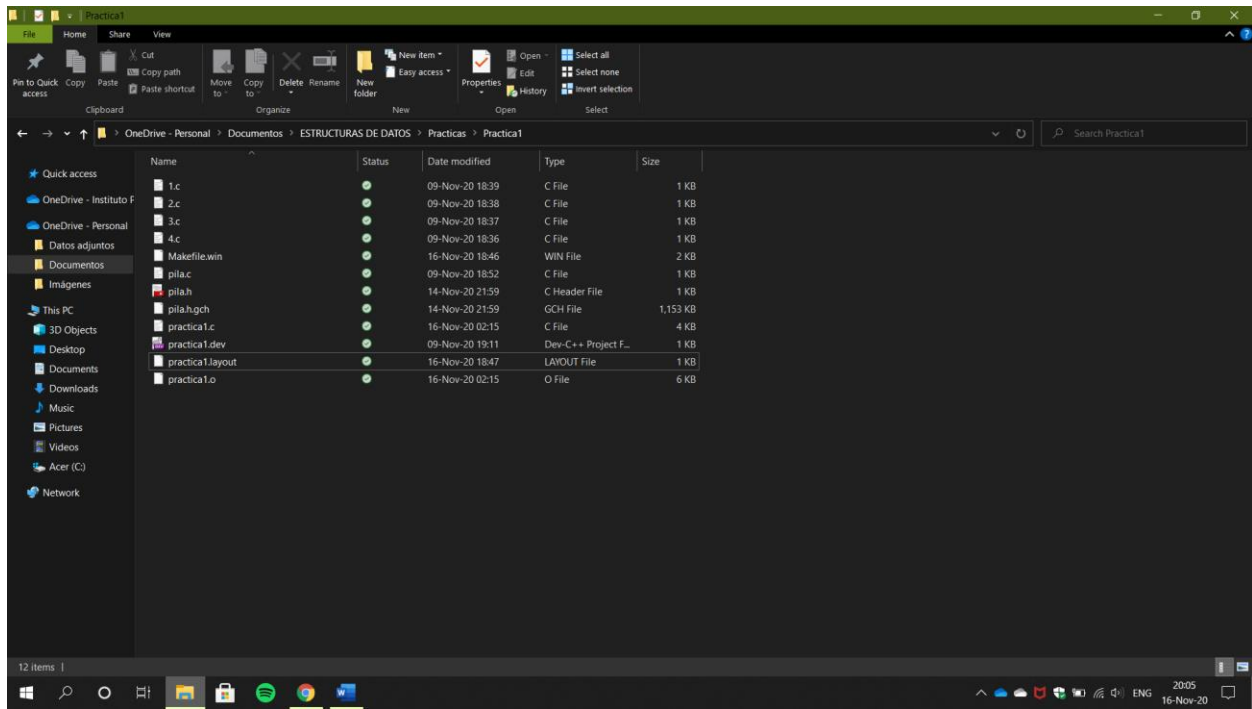
El programa se implementaría por medio de un menú, donde se mostrarían los archivos disponibles para analizar, donde cada uno tendría su respectivo inciso para poder escogerlos, de igual forma estaría la opción de salirse del programa. En caso de que se escoja una opción no disponible, el programa mostrara un mensaje indicando que la opción ingresada no es válida, dando oportunidad a intentarlo de nuevo.

Al momento de seleccionar el archivo deseado, el programa comprobara si el archivo se pudo abrir con éxito, si fue posible abrir el archivo, se ejecutara el análisis de las llaves, de lo contrario mostrara un mensaje de error de lectura y detendrá la ejecución.

Al realizar el análisis, al final el programa mostrara el resultado, indicando si las llaves están balanceadas o el tipo de error encontrado, (las cuales se mencionaron previamente). En caso de que salga uno de los errores planteados en la colocación de llaves, este mismo detendría la ejecución del programa.

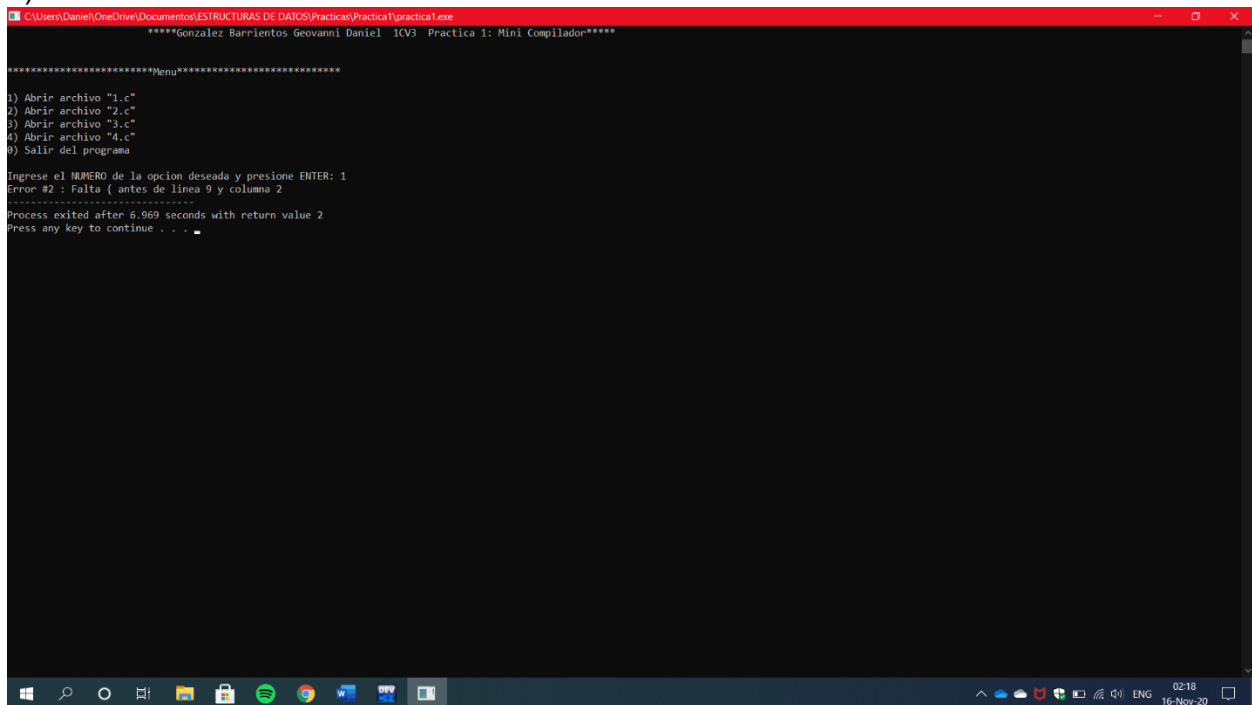
RESULTADOS

Teniendo los archivos muestra (1.c, 2.c, 3.c y 4.c) y los archivos “pila.c” y “pila.h” en la misma carpeta, podemos ejecutar el programa “practica.c”

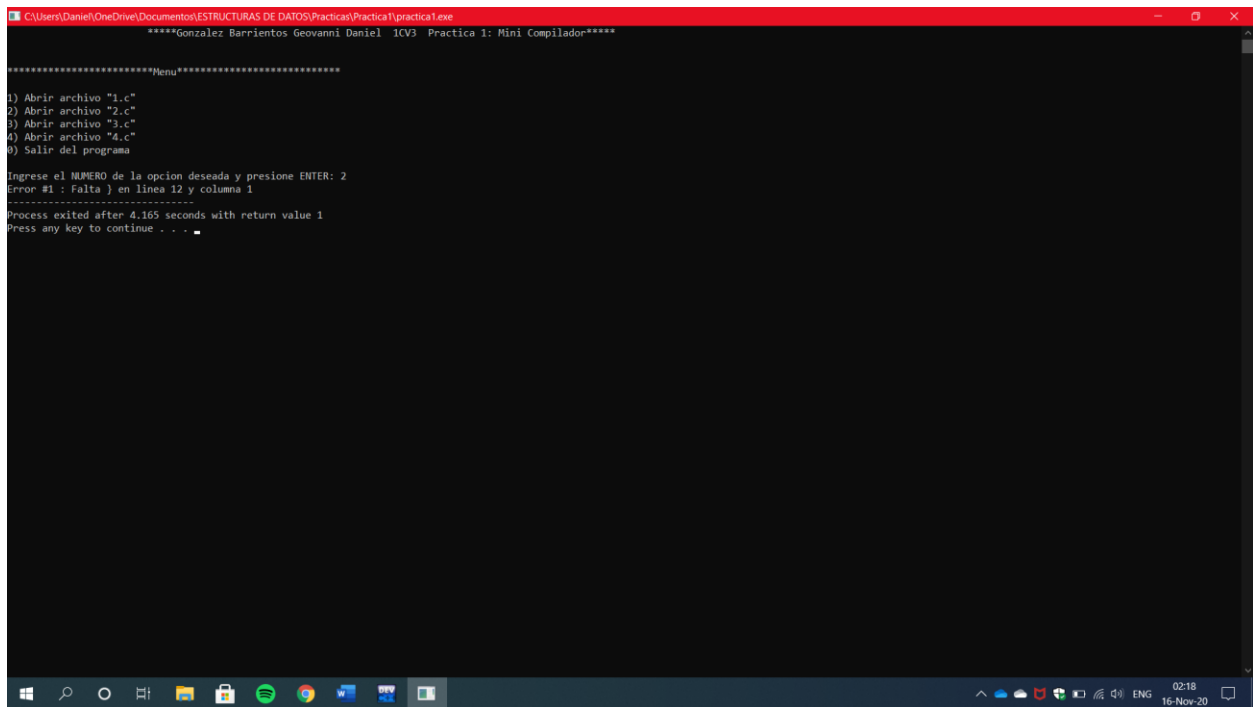


A continuación, se muestran los resultados obtenidos para cada archivo de prueba.

1) Archivo 1.c



2) Archivo 2.c



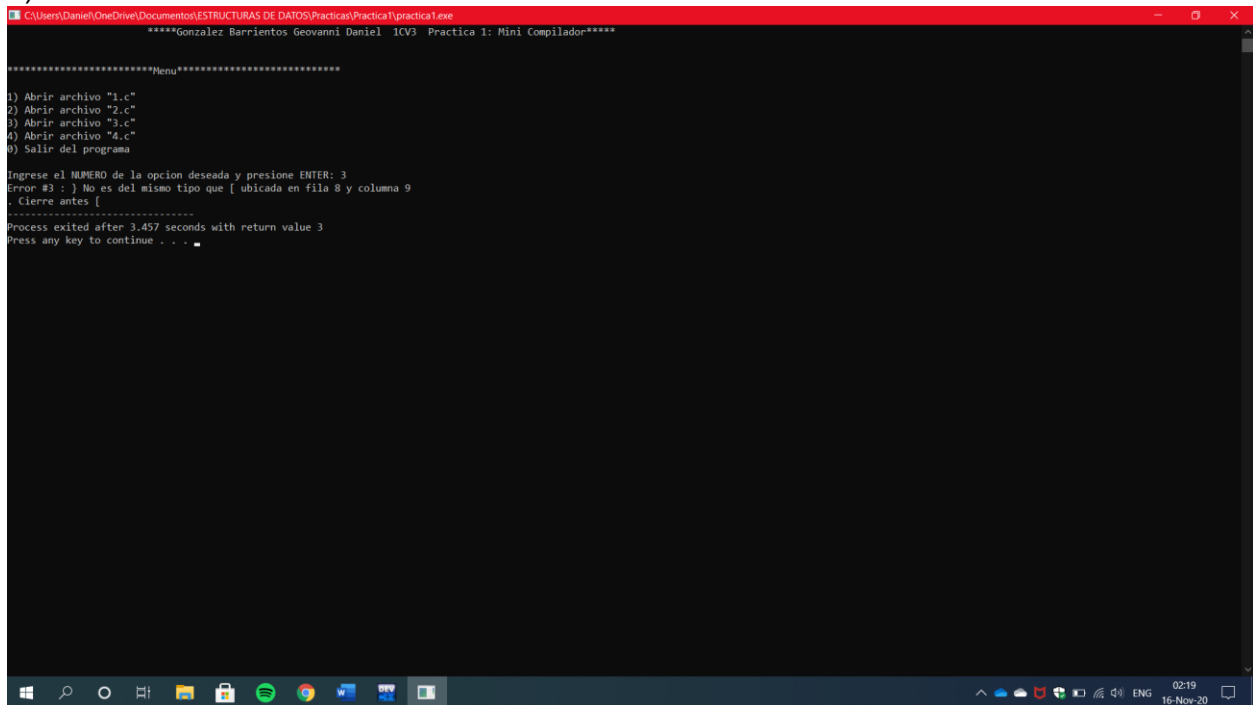
```
C:\Users\Daniel\OneDrive\Documentos\ESTRUCTURAS DE DATOS\Practicas\Practica 1\practica1.exe
*****Gonzalez Barrientos Giovanni Daniel 1CV3 Practica 1: Mini Compilador*****

*****Menu*****

1) Abrir archivo "1.c"
2) Abrir archivo "2.c"
3) Abrir archivo "3.c"
4) Abrir archivo "4.c"
5) Salir del programa

Ingrese el NUMERO de la opcion deseada y presione ENTER: 2
Error #1 : Falta ) en linea 12 y columna 1
-----
Process exited after 4.165 seconds with return value 1
Press any key to continue . . .
```

3) Archivo 3.c



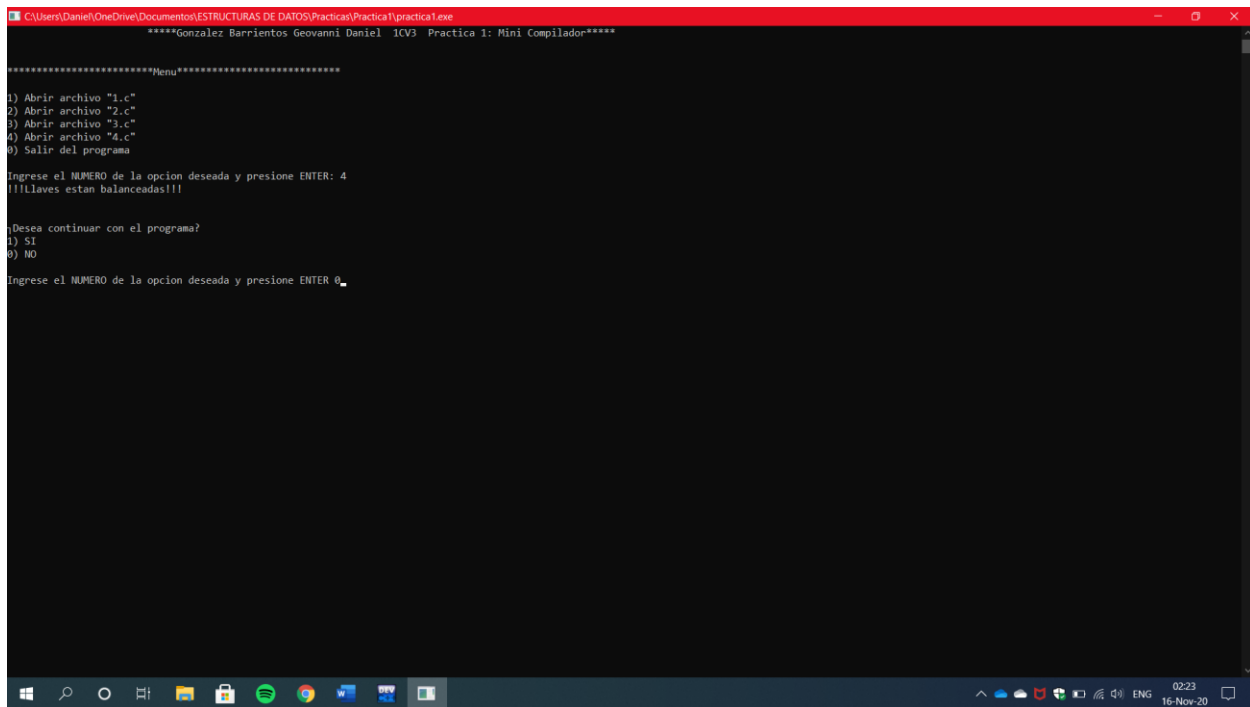
```
C:\Users\Daniel\OneDrive\Documentos\ESTRUCTURAS DE DATOS\Practicas\Practica 1\practica1.exe
*****Gonzalez Barrientos Giovanni Daniel 1CV3 Practica 1: Mini Compilador*****

*****Menu*****

1) Abrir archivo "1.c"
2) Abrir archivo "2.c"
3) Abrir archivo "3.c"
4) Abrir archivo "4.c"
5) Salir del programa

Ingrese el NUMERO de la opcion deseada y presione ENTER: 3
Error #1 : ) No es del mismo tipo que { ubicada en fila 8 y columna 9
. Cierre antes [
-----
Process exited after 3.457 seconds with return value 3
Press any key to continue . . .
```

4) Archivo 4.c



```
C:\Users\Daniel\OneDrive\Documents\ESTRUCTURAS DE DATOS\Practicas\Practica 1\practica1.exe
*****Gonzalez Barrientos Giovanni Daniel 1CV3 Practica 1: Mini Compilador*****

*****Menu*****

1) Abrir archivo "1.c"
2) Abrir archivo "2.c"
3) Abrir archivo "3.c"
4) Abrir archivo "4.c"
0) Salir del programa

Ingrese el NUMERO de la opcion deseada y presione ENTER: 4
!!!!laves estan balanceadas!!!

¿Desea continuar con el programa?
1) SI
0) NO

Ingrese el NUMERO de la opcion deseada y presione ENTER: 0_
```

EXTENSION DE FUNCIONALIDADES

1. La primera funcionalidad que se puede agregar a este programa es la de permitir aplicar las correcciones desde el mismo programa en ejecución, es decir, si este detecta un error por la implementación de las mismas en cualquiera de los “archivos.C”, habría una opción disponible para poder agregar o eliminar una llave según sea necesario. Su funcionamiento podría aplicarse guardando la cadena de texto en un arreglo dinámico, donde aparte de almacenar en la pila el tipo de llave encontrada, junto con su línea y su columna, también guardaríamos su ubicación en este arreglo dinámico, y por medio de sentencias if, podríamos borrar la llave sobrante. Por otro lado, en caso de querer agregar una llave al código fuente, dejaríamos al usuario ingresar el numero de espacio del arreglo dinámico en donde quiere agregar la llave, para ello, deberemos imprimir la cadena de texto por partes en un bucle de tipo do-while, ya que aquí aprovecharemos el uso de un contador para imprimir caracteres en tramos definidos (de 10 en 10 por ejemplo), mostraremos el numero actual de ese contador que servirá como una referencia de el número de posición actual del arreglo y estaremos preguntando al usuario en cada final de tramo si ya sabe en que parte del arreglo agregara la llave o si desea seguir imprimiendo mas tramos del arreglo hasta ubicar el espacio correcto con la ayuda de esta guía. Al final de cada operación realizada, deberemos copiar todo el contenido del arreglo dinámico resultante en el archivo de origen, tratando

de mantener su estructura lo más parecido posible a como venía acomodado de forma inicial.

2. Otra extensión de funcionalidad para esta practica seria darle la opción de abrir cualquier otro archivo que desee analizar con solo teclear su nombre o dirección completa. y hasta tener la posibilidad de analizar 2 o más programas de forma automática si se requiere, mostrando al final un mensaje o lista de los archivos en los que hubo fallos y en cuáles estuvo bien, con el objetivo de evitar estar trabajando y seleccionando de uno en uno los archivos. Esta función sería principalmente cuando son varios archivos que se quiere analizar. Así nosotros podríamos realizar otras actividades mientras el programa se ejecuta.

PROGRAMAS QUE EMPLEAN MISMA ESTRUCTURA DE DATOS

1. Los editores de texto en general utilizan este tipo de estructura para guardar los cambios que se realizan al archivo cada cierto tiempo o después de cierto número de movimientos con la finalidad de poder deshacer las operaciones si es que el usuario sin querer cometió un error, o simplemente porque desea dejar el documento como estaba en algún punto anterior.
2. De igual forma los navegadores web en general utilizan la pila para ir guardando todas las direcciones web en el orden en el que el usuario visita en la sesión actual, con la finalidad de que pueda regresar a la página o páginas anteriores con tan sólo dar clic en el botón "Anterior" o el clásico icono de "←".
3. Otra aplicación de la pila es en los gestores de ventana, ya que guardan el orden en el que se abrieron o utilizaron las ventanas, con la finalidad de que cuando se cierre una, se ponga activa la ventana anterior utilizada a esta que se acaba de cerrar.