

Tarea #1

Inyerman Alexander Xap Chin
202100297

Escuela de Mecánica Eléctrica,
Facultad de Ingeniería, Universidad
de San Carlos de Guatemala

A. Resumen

En esta actividad se preparo la máquina para poder programar en lenguaje C y Python para su posterior uso en el curso.

Se descargo, instalo y configuro GCC y Python. Para esto se descargo GCC de su pagina oficial y luego se extrajo con ayuda del programa 7z luego se ejecuto el instalador. Luego se descargó de Microsoft Store Python. Luego se añadieron en variables del entorno y en variable del usuario y en path de ambos.

Luego verificamos que se hayan instalado correctamente y ya con esto procedimos a programar nuestro código en lenguaje c y Python y los ejecutamos.

B. Objetivos

Objetivo General: Preparar la máquina para empezar a programar.

Objetivos específicos: Instalar GCC y Python y configurarlos para su posterior uso en el curso.

Programar el primer código que muestre hola mundo en lenguaje c y Python.

Programar un código que muestre en la consola mi nombre y credencial en lenguaje c y Python.

C. Marco Teórico

- GCC:

Es un compilador integrado del proyecto GNU para C, C++, Objective C y Fortran; es capaz de recibir un programa fuente en cualquiera de estos lenguajes y generar un programa ejecutable binario en el lenguaje de la maquina donde ha de correr. La sigla GCC significa "GNU Compiler Collection".

Algunos de los comandos que utilizamos son:

1. `$ gcc -o circulo circulo.c`

Se utiliza para producir un ejecutable con fuente de un solo archivo.

2. `$ gcc -c circulo.c`

Se utiliza para crear un modulo objeto, con el mismo nombre del fuente y extensión .o

3. `$ gcc -o circulo circulo.o`

Se utiliza para enlazar un módulo objeto

- Lenguaje C:

C es un lenguaje de programación considerado como uno de los mas importantes en la actualidad con el cual se desarrollan tanto aplicaciones como sistemas operativos a la vez que forma la base de otros lenguajes mas actuales como Java, C++ o C#.

Algunas de las características del lenguaje c son:

Lenguaje estructurado

Programación de nivel medio (se beneficia de las ventajas de la programación de alto y bajo nivel)

No depende del hardware, por lo que se puede migrar a otros sistemas.

Objetivos generales. No es un lenguaje para una tarea específica, pudiendo programar tanto un sistema operativo, una hoja de cálculo o un juego.

Ofrece un control absoluto de todo lo que sucede en el ordenador.

Organización del trabajo con total libertad.

Los programas son producidos de forma rápida y son bastante potentes.

Rico en tipo de datos, operadores y variables en C.

C también tiene su propia sintaxis lo cual significa que tiene una serie de reglas a seguir para que el compilador pueda entender y crear un programa C válido. Es decir, se tiene que establecer como comienza una línea de código en c, como termina o cuando usar por ejemplo comillas o llaves. El lenguaje c hace distinción entre mayúsculas y minúsculas por lo que se programa en minúsculas.

La sintaxis básica en C determina la forma en que se agrupan los caracteres para formar *tokens*, que son la unidad mínima de programación en C. Tomando como ejemplo el programa "Hola Mundo", usado para la introducción a la

mayoría de los lenguajes de programación, tendríamos el siguiente código en C:

```
#include int main() { printf("Hola Mundo"); // línea sencilla de comentarios return 0; /* línea múltiple de comentarios */ }
```

Si seleccionamos sólo la línea de código `printf`, tendríamos los siguientes *tokens*:

printf, (, " , **Hola Mundo**, " ,), y ;. Por lo tanto, podemos decir, que estos *tokens* son los pequeños bloques con los que se programa en C, pudiendo ser variables, identificadores, constantes, palabras clave, símbolos que comprenda el lenguaje o sentencias en lenguaje C.

El comienzo de todo programa en c debe comenzar por `#include` cuya función es inicializar el entorno de trabajo, en nuestro ejemplo vinculando el archivo `stdio.h` (biblioteca de c) que a su vez contiene la orden `printf`.

La función en lenguaje C `int main()` hace que el programa vuelva a ese punto de retorno tras ejecutarse y su orden está delimitada entre `{}`.

Al hablar de bibliotecas en lenguaje C nos referimos a una serie de librerías que son usadas en este lenguaje de programación y están formadas por una serie de código con el que podemos leer el teclado, realizar operaciones matemáticas, imprimir en pantalla... La biblioteca referente es la estándar de ANSI C formada por 24 archivos de cabecera. Según su tarea, se

clasifican en categorías, como por ejemplo matemáticas, de salida y entrada, de uso de textos, de gestión de memoria...

Entre las librerías más destacadas encontramos `stdio.h` (de entrada y salida), `conio.h` (potente gestión de textos), `math.h` (encargada de operaciones matemáticas), `time.h` (obtiene fecha y hora actual para poder utilizarla según sus necesidades), etcétera.

- Python

Python es un lenguaje de programación de código abierto, creado por Guido van Rossum en 1991. Se trata de un lenguaje orientado a objetos, fácil de interpretar y con una sintaxis que permite leerlo de manera semejante a como se lee el inglés. Es un lenguaje interpretado, esto significa que el código de programación se convierte en bytecode y luego se ejecuta por el intérprete, que, en este caso, es la máquina virtual de Python.

Algunas de las características de Python son:

Python es un lenguaje interpretado, lo que significa que ejecuta directamente el código línea por línea. Si existen errores en el código del programa, su ejecución se detiene. Así, los programadores pueden encontrar errores en el código con rapidez.

Python utiliza palabras similares a las del inglés. A diferencia de otros

lenguajes de programación, Python no utiliza llaves. En su lugar, utiliza sangría.

Los programadores no tienen que anunciar tipos de variables cuando escriben código porque Python los determina en el tiempo de ejecución. Debido a esto, es posible escribir programas de Python con mayor rapidez.

Python es más cercano a los idiomas humanos que otros lenguajes de programación. Por lo tanto, los programadores no deben preocuparse de sus funcionalidades subyacentes, como la arquitectura y la administración de la memoria.

Python considera todo como un objeto, pero también admite otros tipos de programación, como la programación estructurada y la funcional.

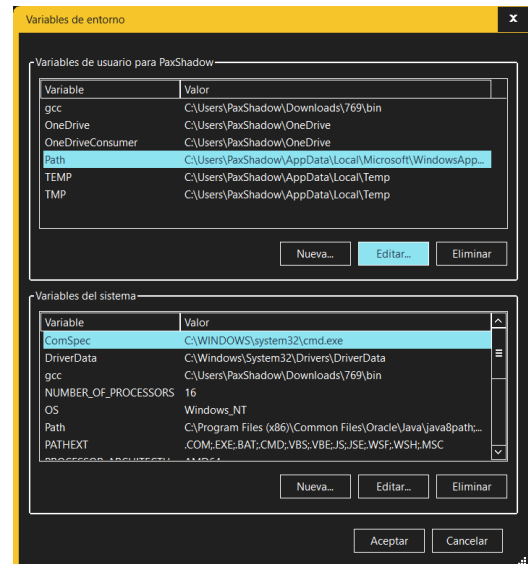
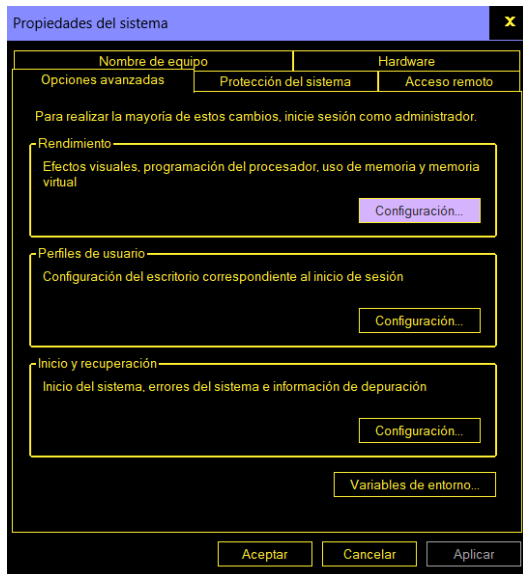
D. Marco Practico

Lo primero que se realizó fue descargar gcc desde el enlace dado por el ingeniero.

Luego extrajimos los datos con la ayuda del programa 7z.

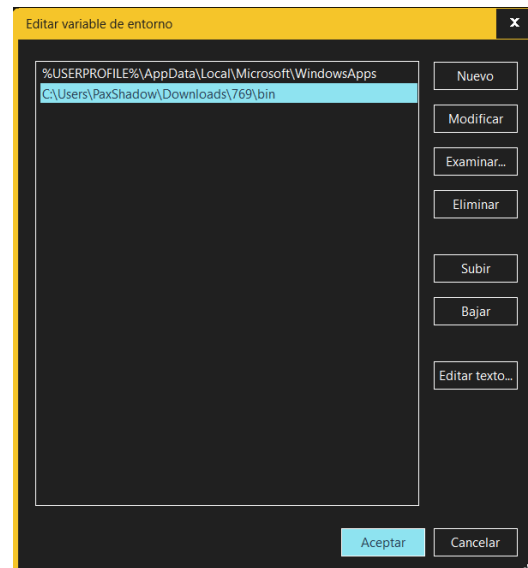
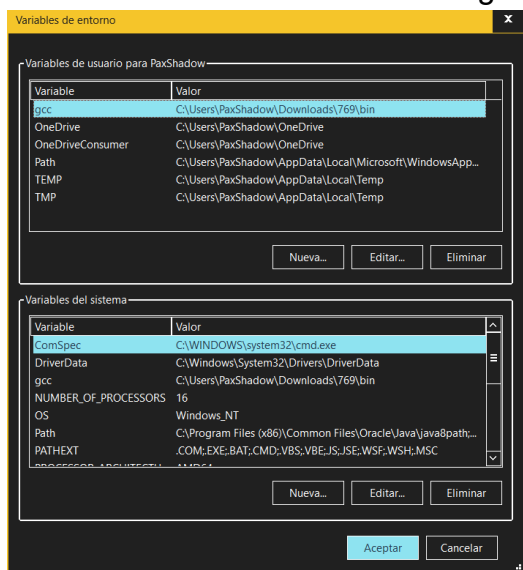
Luego descargamos Python de Microsoft Store.

Luego debemos ir a cambiar variables del entorno del sistema.

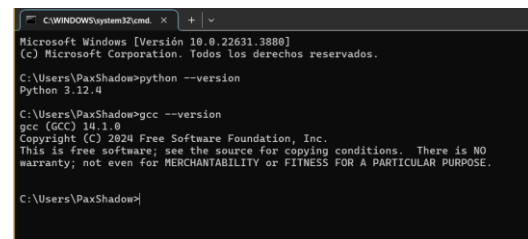


Luego agregamos el directorio y lo nombramos gcc en ambos.

Luego tenemos que agregar los directorios a las variables del entorno del sistema y en variables de usuario a las cuales nombramos gcc



Por último, abrimos un cmd y verificamos que estén instalados gcc y Python.

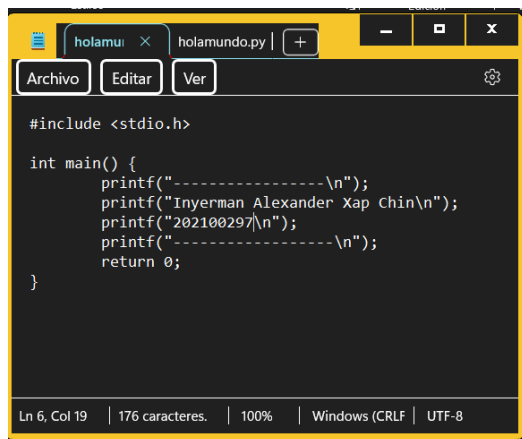


Luego solo creamos una carpeta para guardar nuestros códigos en C.

Y ya con esto estamos listos para comenzar a programar.

E. Código

Código en lenguaje C

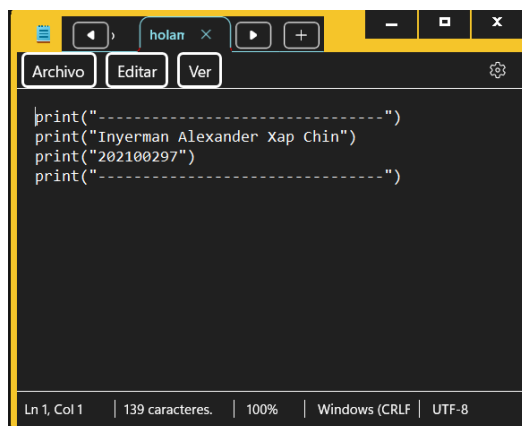


```
#include <stdio.h>

int main() {
    printf("-----\n");
    printf("Inyerman Alexander Xap Chin\n");
    printf("202100297\n");
    printf("-----\n");
    return 0;
}
```

Ln 6, Col 19 | 176 caracteres. | 100% | Windows (CRLF) | UTF-8

Código en Python



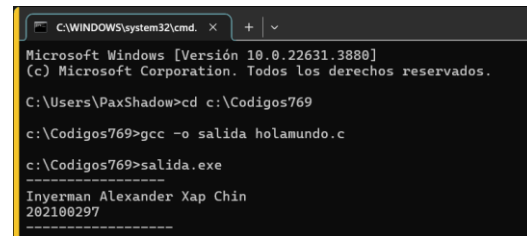
```
print("-----")
print("Inyerman Alexander Xap Chin")
print("202100297")
print("-----")
```

Ln 1, Col 1 | 139 caracteres. | 100% | Windows (CRLF) | UTF-8

F. Resultados

Código en lenguaje C corriendo.

Primero debimos compilar el código para luego poder ejecutarlo. Y mostro nuestras credenciales como se esperaba.

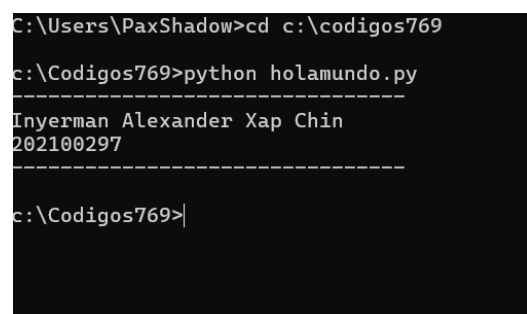


```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Versión 10.0.22631.3880]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\PaxShadow>cd c:\Codigos769
c:\Codigos769>gcc -o salida holamundo.c
c:\Codigos769>salida.exe
-----
Inyerman Alexander Xap Chin
202100297
-----
```

Código en Python corriendo

A diferencia del lenguaje C este no requiere que se deba compilar antes de ejecutarlo por lo que solo lo ejecutamos y como no hay errores el código corre sin problemas.



```
C:\Users\PaxShadow>cd c:\codigos769
c:\Codigos769>python holamundo.py
-----
Inyerman Alexander Xap Chin
202100297
-----
c:\Codigos769>
```

G. Conclusiones

Con la elaboración de esta actividad logramos preparar nuestra máquina para empezar a programar.

Instalamos GCC y Python y los configuramos para su correcta utilización a futuro.

Programamos nuestros primeros códigos de hola mundo en lenguaje C y Python.

Aprendimos el como se debe iniciar un código en lenguaje C, #include para inicializar el área de trabajo.

Aprendimos la librería stdio.h para poder utilizar el comando printf.

Aprendimos a compilar y ejecutar los códigos en lenguaje C.

Aprendimos las diferencias entre los lenguajes entre lenguaje c y Python.

H. Referencias

<https://aws.amazon.com/es/what-is/python/>

<https://openwebinars.net/blog/queues-c/>

https://www.fing.edu.uy/tecnoinf/maldonado/cursos/eda/material/otros/compilador_GCC.pdf