MANUAL DEL PROGRAMADOR

CHMAQUINA

LICETH JUANITA USMA LONDOÑO

UNIVERSIDAD DE CALDAS

2023

Índice

Introducción

Este Manual tiene como objetivo proporcionar una guía detallada sobre cómo utilizar el programa “CHMaquina”. El programa es una aplicación que realiza una simulación gráfica de un **chcomputador** ficticio de funcionamiento básico el cual permite la lectura de los archivos de programas con extensión “.ch”, también hace chequeo de sintaxis; está escrita en Python y su interfaz fue construida utilizando la biblioteca Tkinter

Requerimientos

- Sistema Operativo: Windows 7 o superior, o Linux
- Python 3x instalado (Tkinter viene incluido)
- Biblioteca Numpy instalada
- Biblioteca Pillow instalada

Instalación

Para instalar el programa “CHMaquina en python Tkinter”, siga los siguientes pasos:

1. Descargue el archivo rar del programa desde el modle
2. Descomprima el archivo en una carpeta de su elección.
3. Ejecute el archivo “main.py” desde la línea de comandos utilizando Python.

Instalación de Numpy

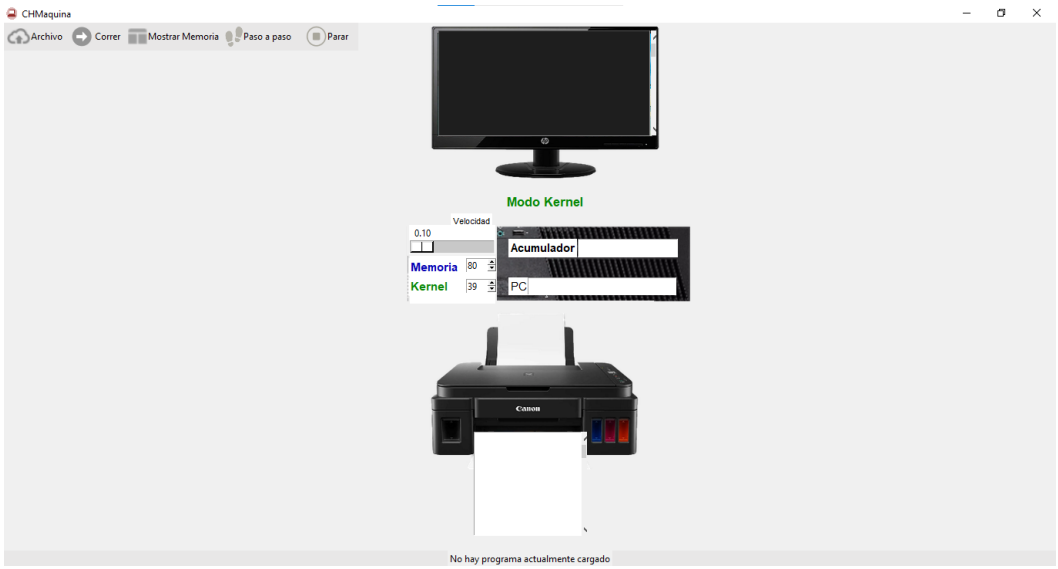
Para instalar Numpy solo hay que entrar en el cmd para escribir el siguiente comando `pip install numpy` (hay que tener instalado python)

Instalación de Pillow

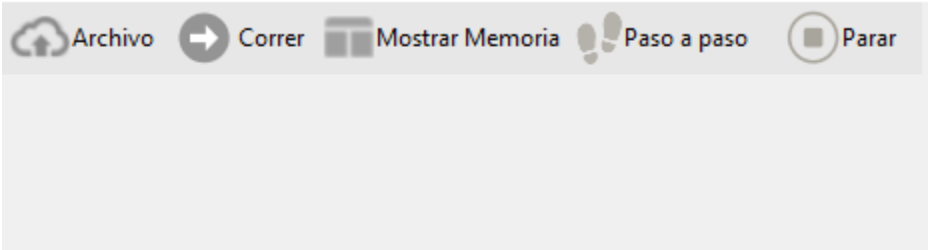
Para instalar Pillow solo hay que entrar en el cmd para escribir el siguiente comando `pip install Pillow` (hay que tener instalado python)

Uso del Programa

El programa “CHMaquina” es una simulación simple de un chcomputador. Al iniciar el programa, aparecerá una ventana como la siguiente:

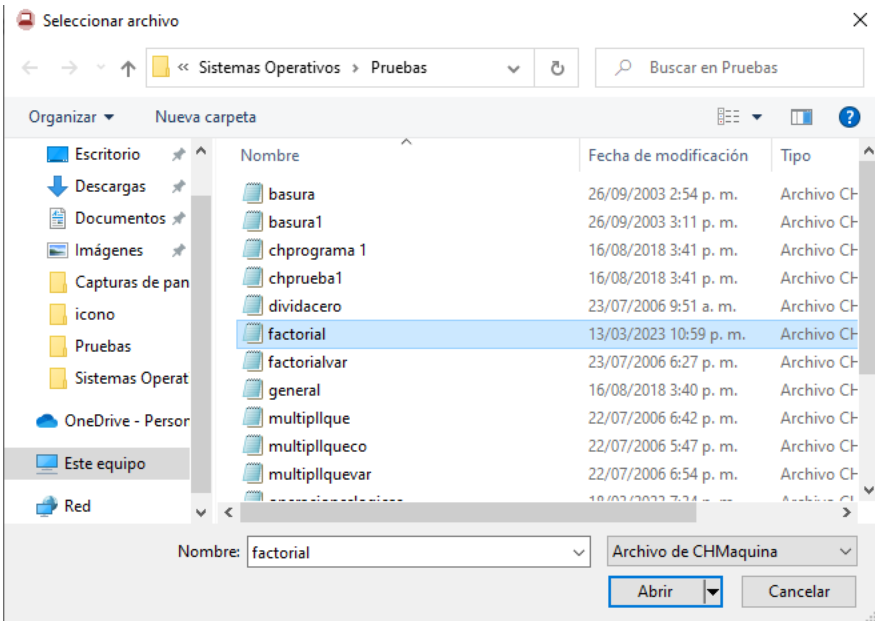


Opciones del Header

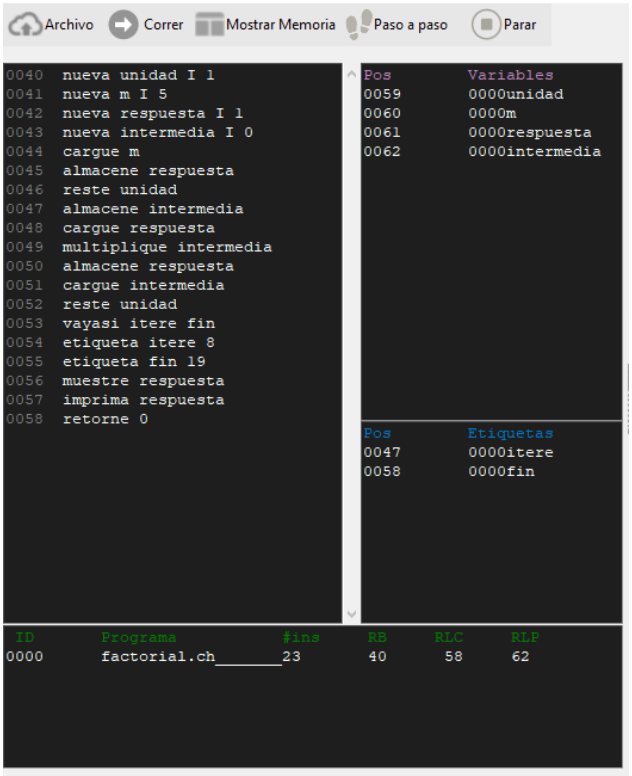


Archivo

Esta opción abre una ventana como la mostrada a continuación para buscar el archivo que se desea cargar:

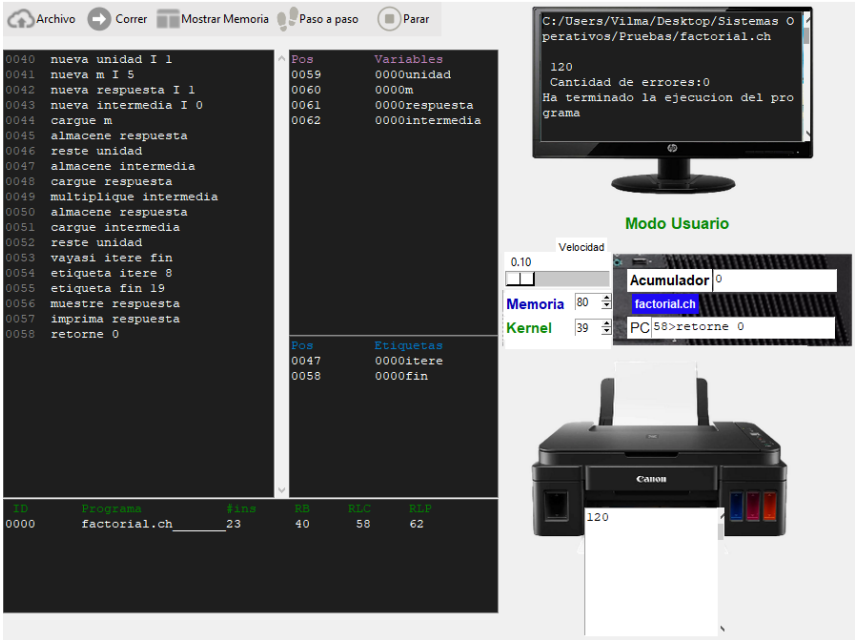


cuando se carga el archivo aparece unas cajas de texto, una tiene el contenido, otra el listado de variables, el listado de etiquetas y por ultimo algunos datos de programa como la cantidad de instrucciones, el inicio, el final etc... Aquí un ejemplo:



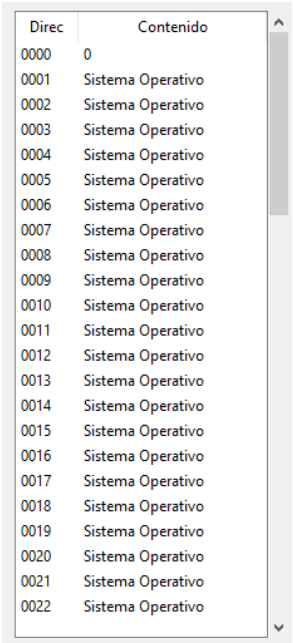
Correr

Esta opción corre el programa si no hay errores aparecerá el resultado en la caja de texto del monitor y en la caja del texto de la impresora dependiendo de la instrucción de los usuarios. Aquí un ejemplo:



Mostrar Memoria

Esta opción muestra el array donde se guardan cada una de las líneas de los programas en una tabla. Aquí una ejemplo:



Pasó a Paso
Esta opción le permite al usuario ver instrucción por instrucción y observar de una mejor forma como cambia el acumulador.

Parar
Detiene la ejecución del programa hasta que el usuario le da click nuevamente.

Otras Opciones

Memoria
Aquí el usuario escoge la capacidad de la memoria.

Kernel
Aquí el usuario escoge la cantidad de espacios que va ocupar el kernel.

Velocidad
Aquí el usuario escoge la velocidad con que se va a correr el chprograma.

Código Fuente

El código fuente del programa “CHMaquina” solo se encuentra en un archivo llamado “main.py” este contiene una combinación de la parte gráfica y la parte lógica del programa.

La lógica del programa se basa en un array que funciona como memoria del programa que almacena cada programa cargado y de un diccionario llamado principal que almacena información de cada programa como etiquetas y variables este tiene la siguiente estructura: (el 0 indica el ID del programa)

```
Principal = {

    0:{
        Nombre:"",  #el nombre del archivo
        Dirección: "", #la dirección del archivo
        Inicio: 0, #el valor desde donde inicia el archivo en el array
        Final: 0, #el valor desde donde finaliza el archivo en el array
        string_errores: "",
        contador_errores: 0,
        Variables:{
            "key": posición  # key: nombre de la variable
                        #posición: donde se encuentra la variable en el array
        },
        Etiquetas:{
            "key": inicio  # key: nombre de la variable
                           # Posición: donde se debe devolver el programa
        }
    }
}
```

Cada que se carga un archivo se crea un diccionario dentro del Principal.

Correr

Al correr el archivo lo que sucede es que se recorre con un range el contador de programas que es el que tiene la cantidad de programas que se han cargado, y despues adentro de este se recorre el array principal, por consiguiente se manipula el array y el diccionario para ejecutar cada una de las instrucciones que

están en el archivo a partir de if anidados los cuales validan diversas palabras claves que indican operaciones y al entrar en estos se ejecutan dichas operaciones. Aquí un ejemplo con operaciones de enteros:

```
for i in range(0, programas ): # recorre de 0 a la cantidad de programas que hayan
    while inicio <= final:  #correr de cierta posicion en el array  inicio fin diccionario
        k = my_array[inicio]  #obtiene la linea del archivo
        if "sume" in k: #si sume se encuentra en la cadena
            lista = k.split() #partimos la cadena
            tipo = principal[i]["variables"][lista[1]]["tipo"] # carga el tipo de la varaible
            if  tipo == "I" or tipo == "R": #solo se aceptan este tipo de variables
                if cargar_variable(lista[1], i):  #funcion que permite saber si la variable se encuentra
en el diccionario
                    numero = int(my_array[principal[i]["variables"][lista[1]]["posicion"]])
                    #lo convierte a entero
                    acumulador += numero #se lo pasa al acumulador
                else:
                    principal[i]["contador_errores"] +=1  #contador de errores de cada programa
                    #string con los errores del programa
                    principal[i]["string_errores"]  += f"La variable que intenta cargar no existe no
invente error linea {inicio} \n"
            else:
                principal[i]["string_errores"]  += f"El tipo de variable no corresponde con la operacion
¡Error! en la linea {inicio} \n"
```

Pasó a Paso

El paso a paso está dentro de las funciones de correr el archivo que este también es un if que solo puede ser accedido cuando una variable booleana se convierta en verdadera que es cuando el usuario le da click, este entra y comienzan a aparecer un letrero tipo mensaje que tiene un tipo de valor booleano. Aquí el código:

```
if paso.get():  #mensaje que se muestra cuando paso es true
    paso.set((messagebox.askokcancel("Programa", "Desea seguir en el modo paso a paso?")))
    # muestra un mensaje al usuario dependiendo de lo que escoja
    # el valor booleano de paso se cambia
```

Parar

El parar también es un if que está dentro de la función de correr archivo que al igual que el de paso a paso solo se puede acceder cuando la variable booleana se convierta en verdadera pero aquí lo que hay es un ciclo infinito con la función Sleep de python que se encarga de detener el programa, este ciclo solo se rompe cuando la variable sea falsa que esto se logra dando click nuevamente. Aquí el código:

```
if parar.get():  # parar: valor booleano de tkinter
    while parar.get():#ciclo con sleep que se detiene cuando parar sea false
        t.sleep(1)     #duerme el programa por un segundo
        ventana.update()   #actualiza la ventana
```

Solución de Problemas

Si tiene problemas para ejecutar el programa, asegúrese de tener Python y las librerías Numpy y Pillow instalados en su sistema. Si el archivo muestra algún error o no funciona correctamente, revise el código fuente y asegúrese de que este correctamente escrito y estructurado.

Referencias

Para obtener mayor información sobre la programación en python y Tkinter, consulte la documentación oficial de python y Tkinter en línea.