SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TLAXIACO

Practica 7: Simulación de administración de memoria en la CPU

Presenta:
Hernández Martínez Adriana

Numero de control: 22620083

Carrera:
Ingeniería en Sistemas Computacionales

Grupo:5B

Asignatura:
Arquitectura de computadoras

Docente:
Osorio Salinas Edward

Tlaxiaco, Oaxaca, a 5 de diciembre de 2024

"Educación, Ciencia y Tecnología, Progreso día con día"®

Paso 1: Cargar el programa en memoria

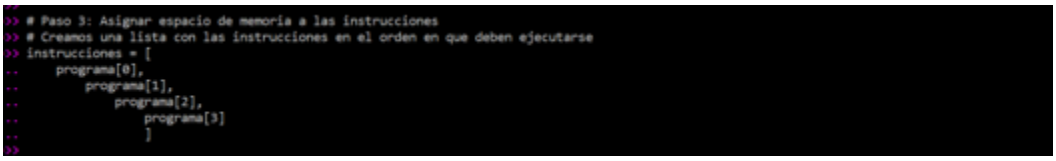
Creamos un diccionario llamado programa que simula las instrucciones de u n programa cargado en memoria. Cada clave representa una dirección de memoria.

Paso 2: Asignar espacio de memoria a las variables

Simulamos la memoria del sistema operativo utilizando un diccionario llamado memoria, donde cada clave representa una dirección de memoria, y el valor es el contenido almacenado en esa posición.

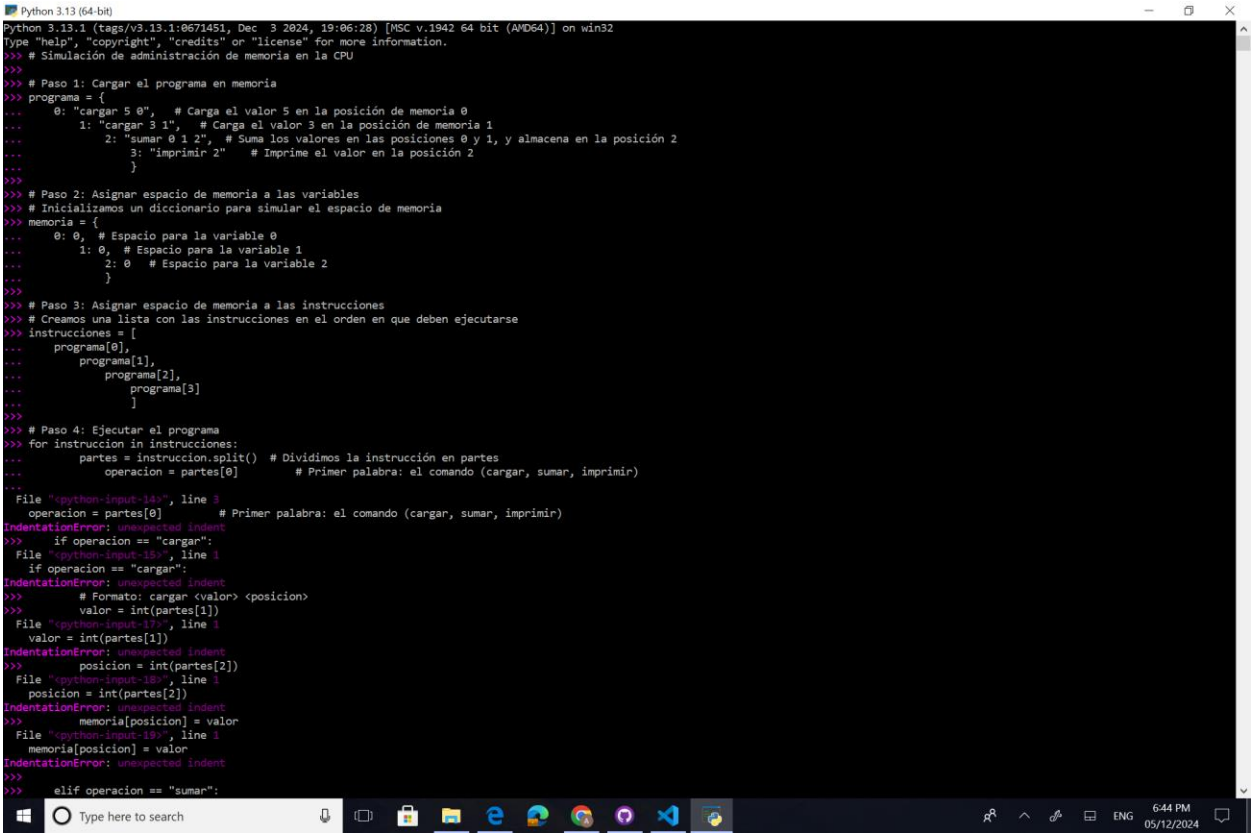
Paso 3: Asignar espacio de memoria a las instrucciones

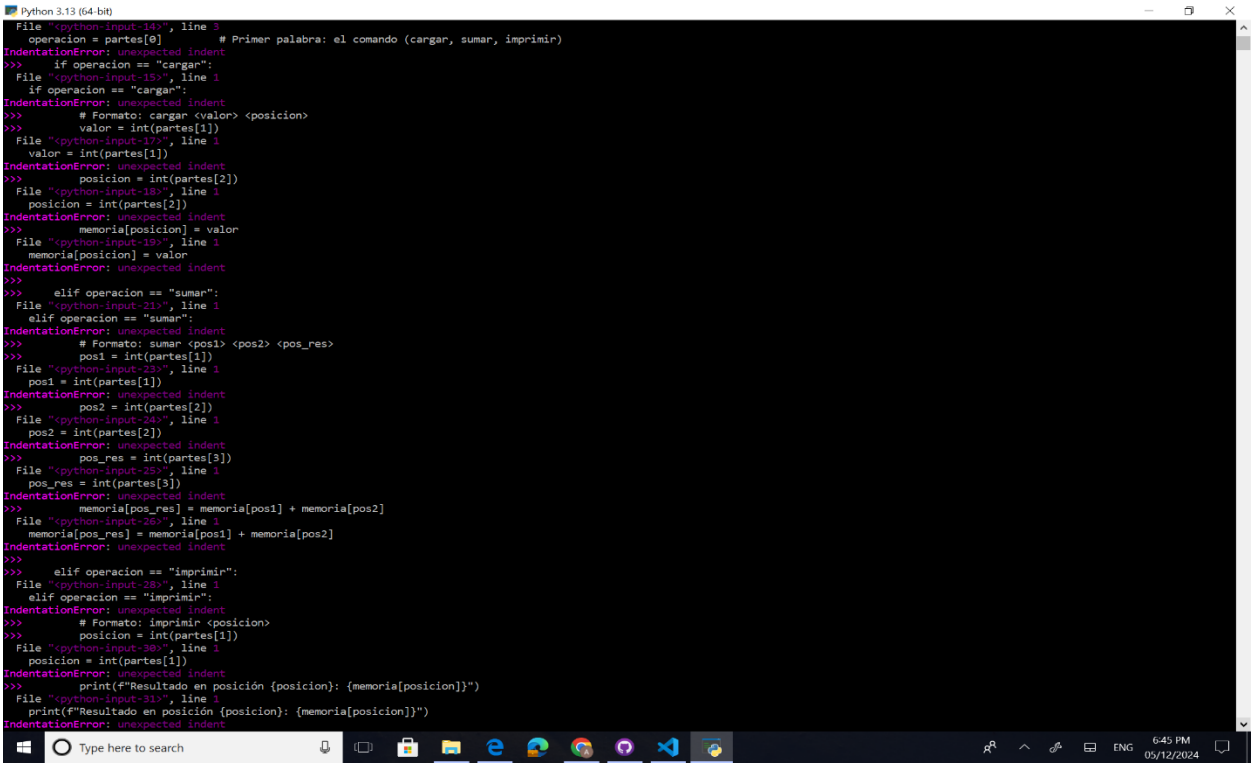
Creamos una lista llamada instrucciones que contiene las instrucciones del programa en el orden en que deben ejecutarse.

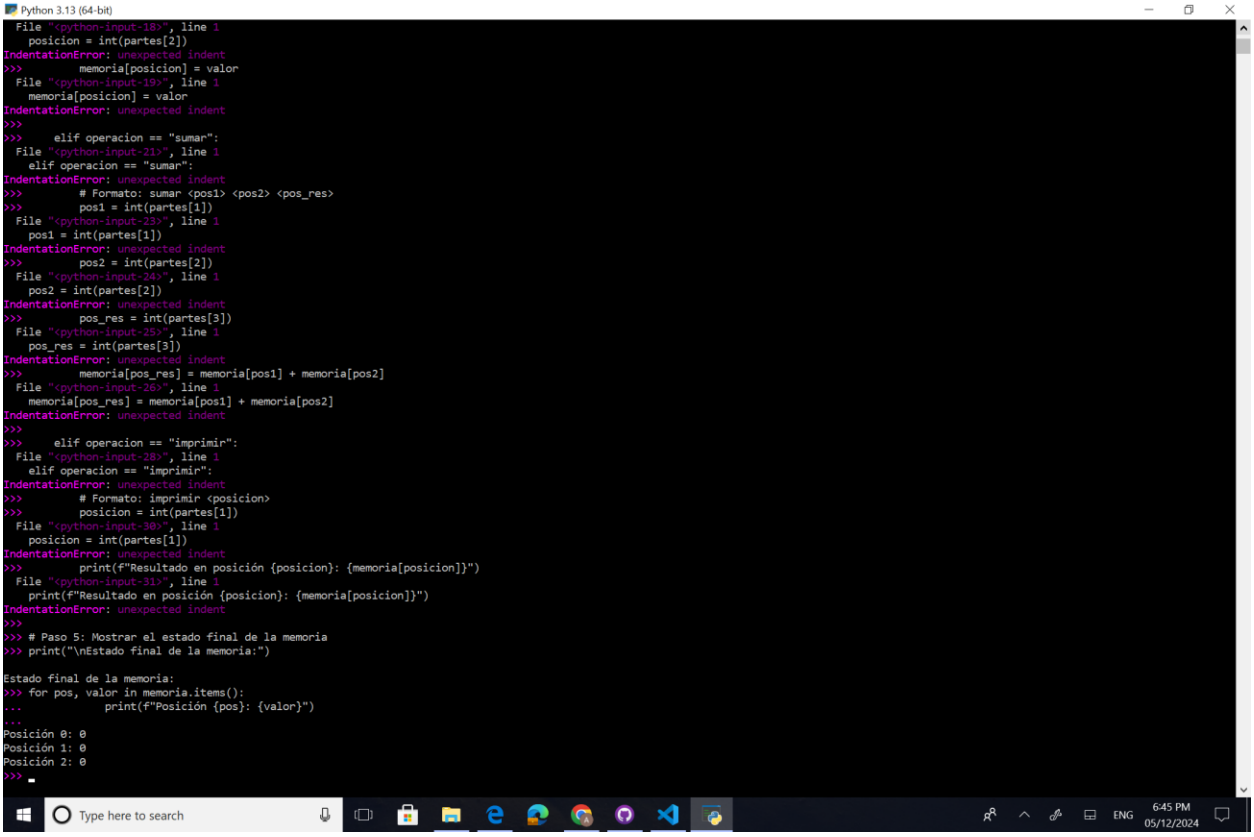


Paso 4: Ejecutar el programa

Recorremos la lista de instrucciones y ejecutamos cada una según el tipo de operación (cargar, sumar, imprimir).


Python 3.13 (64-bit)
Python 3.13.1 (tags/v3.13.1:0671451, Dec 3 2024, 19:06:28) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> # Simulación de administración de memoria en la CPU
>>>
>>> # Paso 1: Cargar el programa en memoria
>>> programa = {
... 0: "cargar 5 0", # Carga el valor 5 en la posición de memoria 0
... 1: "cargar 3 1", # Carga el valor 3 en la posición de memoria 1
... 2: "sumar 0 1 2", # Suma los valores en las posiciones 0 y 1, y almacena en la posición 2
... 3: "imprimir 2" # Imprime el valor en la posición 2
... }
>>>
>>> # Paso 2: Asignar espacio de memoria a las variables
>>> # Inicializamos un diccionario para simular el espacio de memoria
>>> memoria = {
... 0: 0, # Espacio para la variable 0
... 1: 0, # Espacio para la variable 1
... 2: 0 # Espacio para la variable 2
... }
>>>
>>> # Paso 3: Asignar espacio de memoria a las instrucciones
>>> # Creamos una lista con las instrucciones en el orden en que deben ejecutarse
>>> instrucciones = [
... programa[0],
... programa[1],
... programa[2],
... programa[3]
...]
>>>
>>> # Paso 4: Ejecutar el programa
>>> for instruccion in instrucciones:
... partes = instruccion.split() # Dividimos la instrucción en partes
... operacion = partes[0] # Primer palabra: el comando (cargar, sumar, imprimir)
...
File "<python-input-14>", line 3
operacion = partes[0] # Primer palabra: el comando (cargar, sumar, imprimir)
IndentationError: unexpected indent
>>> if operacion == "cargar":
File "<python-input-15>", line 1
if operacion == "cargar":
IndentationError: unexpected indent
>>> # Formato: cargar <valor> <posicion>
>>> valor = int(partes[1])
File "<python-input-17>", line 1
valor = int(partes[1])
IndentationError: unexpected indent
>>> posicion = int(partes[2])
File "<python-input-18>", line 1
posicion = int(partes[2])
IndentationError: unexpected indent
>>> memoria[posicion] = valor
File "<python-input-19>", line 1
memoria[posicion] = valor
IndentationError: unexpected indent
>>>
>>> elif operacion == "sumar":
Type here to search
ENG
6:44 PM
05/12/2024


Python 3.13 (64-bit)
File "<python-input-14>", line 3
operacion = partes[0] # Primer palabra: el comando (cargar, sumar, imprimir)
IndentationError: unexpected indent
>>> if operacion == "cargar":
File "<python-input-15>", line 1
if operacion == "cargar":
IndentationError: unexpected indent
>>> # Formato: cargar <valor> <posicion>
>>> valor = int(partes[1])
File "<python-input-17>", line 1
valor = int(partes[1])
IndentationError: unexpected indent
>>> posicion = int(partes[2])
File "<python-input-18>", line 1
posicion = int(partes[2])
IndentationError: unexpected indent
>>> memoria[posicion] = valor
File "<python-input-19>", line 1
memoria[posicion] = valor
IndentationError: unexpected indent
>>>
>>> elif operacion == "sumar":
File "<python-input-21>", line 1
elif operacion == "sumar":
IndentationError: unexpected indent
>>> # Formato: sumar <pos1> <pos2> <pos_res>
>>> pos1 = int(partes[1])
File "<python-input-23>", line 1
pos1 = int(partes[1])
IndentationError: unexpected indent
>>> pos2 = int(partes[2])
File "<python-input-24>", line 1
pos2 = int(partes[2])
IndentationError: unexpected indent
>>> pos_res = int(partes[3])
File "<python-input-25>", line 1
pos_res = int(partes[3])
IndentationError: unexpected indent
>>> memoria[pos_res] = memoria[pos1] + memoria[pos2]
File "<python-input-26>", line 1
memoria[pos_res] = memoria[pos1] + memoria[pos2]
IndentationError: unexpected indent
>>>
>>> elif operacion == "imprimir":
File "<python-input-28>", line 1
elif operacion == "imprimir":
IndentationError: unexpected indent
>>> # Formato: imprimir <posicion>
>>> posicion = int(partes[1])
File "<python-input-30>", line 1
posicion = int(partes[1])
IndentationError: unexpected indent
>>> print(f"Resultado en posición {posicion}: {memoria[posicion]}")
File "<python-input-31>", line 1
print(f"Resultado en posición {posicion}: {memoria[posicion]}")
IndentationError: unexpected indent
Type here to search
ENG
6:45 PM
05/12/2024


Python 3.13 (64-bit)
File "<python-input-18>", line 1
posicion = int(partes[2])
IndentationError: unexpected indent
>>> memoria[posicion] = valor
File "<python-input-19>", line 1
memoria[posicion] = valor
IndentationError: unexpected indent
>>>
>>> elif operacion == "sumar":
File "<python-input-21>", line 1
elif operacion == "sumar":
IndentationError: unexpected indent
>>> # Formato: sumar <pos1> <pos2> <pos_res>
>>> pos1 = int(partes[1])
File "<python-input-23>", line 1
pos1 = int(partes[1])
IndentationError: unexpected indent
>>> pos2 = int(partes[2])
File "<python-input-24>", line 1
pos2 = int(partes[2])
IndentationError: unexpected indent
>>> pos_res = int(partes[3])
File "<python-input-25>", line 1
pos_res = int(partes[3])
IndentationError: unexpected indent
>>> memoria[pos_res] = memoria[pos1] + memoria[pos2]
File "<python-input-26>", line 1
memoria[pos_res] = memoria[pos1] + memoria[pos2]
IndentationError: unexpected indent
>>>
>>> elif operacion == "imprimir":
File "<python-input-28>", line 1
elif operacion == "imprimir":
IndentationError: unexpected indent
>>> # Formato: imprimir <posicion>
>>> posicion = int(partes[1])
File "<python-input-30>", line 1
posicion = int(partes[1])
IndentationError: unexpected indent
>>> print(f"Resultado en posición {posicion}: {memoria[posicion]}")
File "<python-input-31>", line 1
print(f"Resultado en posición {posicion}: {memoria[posicion]}")
IndentationError: unexpected indent
>>>
>>> # Paso 5: Mostrar el estado final de la memoria
>>> print("\nEstado final de la memoria:")
Estado final de la memoria:
>>> for pos, valor in memoria.items():
... print(f"Posición {pos}: {valor}")
...
Posición 0: 0
Posición 1: 0
Posición 2: 0
>>>
Type here to search
ENG
6:45 PM
05/12/2024