

1. Describe cómo el software que has creado se ha relacionado con los componentes físicos del dispositivo (memoria RAM, procesador, periféricos, etc.) durante la ejecución de los tres lenguajes (interpretado, compilado y en máquina virtual).

Python: El intérprete de python tradujo y ejecutó cada línea para así poder ejecutar el programa, introduce el dato por el periférico y se almacena en las celdas de la memoria RAM hasta ser necesario para imprimirlo por pantalla.

C: El código en C se compila directamente en lenguaje máquina y como el programa anterior utiliza las celdas de memoria para guardar los datos introducidos desde los periféricos hasta ser necesario imprimirlo por pantalla.

Java: El código Java se pasa a bytecode, el procesador ejecuta las instrucciones del bytecode, en memoria se almacenan objetos y se ejecuta el programa y mediante los periféricos introducimos los datos que se almacenan en la memoria principal hasta ser necesarios para imprimirlo por pantalla

2.1 Explica cómo el código fuente que escribiste se transformó en código objeto y ejecutable en el caso de los lenguajes compilados. ¿Generaste archivos intermedios (código objeto)? ¿Qué nombres tomaron estos archivos?

En C con la herramienta gcc se puede obtener el código ejecutable directamente, mantiene su nombre pero tiene una extensión .exe.

En Java usé javac para poder compilarlo creando un archivo bytecode. El nombre se mantuvo el mismo pero cambiando la extensión.

2.2 Para los lenguajes interpretados, describe cómo el código fuente se ejecutó directamente, sin generar archivos de código objeto o ejecutable.

Se va ejecutando a la vez que traduciendo sin necesidad de usar un compilador.

2.3 Para el lenguaje que genera código intermedio (Java, C#), explica cómo el código fuente se transformó en código intermedio (bytecode) y cómo este fue ejecutado por la máquina virtual.

Se desarrolla el código en alto nivel y después se transforma a código intermedio que la JVM es capaz de ejecutar.

3.1 Describe el proceso de generación de código intermedio (bytecode) en el lenguaje que utilizaste que emplea una máquina virtual (por ejemplo, Java o C#).

Tras desarrollar el código fuente, usando la javac generamos un código intermedio (bytecode) y se interpreta con la JVM

3.2 Explica qué rol juega la máquina virtual en la ejecución del código y cómo difiere de la ejecución directa en un sistema operativo como ocurre con los lenguajes compilados e interpretados.

La máquina virtual hace de intermediario entre el procesador y el código, de esta forma se puede ejecutar en cualquier máquina, de forma directa sobre el S.O, se crea el ejecutable para esa máquina con ese hardware y sistema operativo.