Meza Martínez Emmanuel
No. Cuenta: 311309649

Tarea 01 / Sistemas Operativos

1. Diferencias entre Java y C

    - En sintaxis es prácticamente idéntica
    - Orientación a objetos: En C se pueden escribir programas orientados a objetos, no orientados a objetos o mezcla de ambos (tener clases con funciones globales en el mismo programa). En Java es puramente orientado a objetos; cualquier función debe pertenecer a alguna clase (método).
    - Control sobre plataforma: en C el programador tiene control del procesador. En Java el programa se ejecuta sobre un procesador virtual (JVM) que abstrae al programador de la plataforma concreta.
    - Código de máquina generado: En C el compilador genera código de máquina para una plataforma concreta. En Java el compilador genera código de máquina (bytecode) para el procesador JVM; dicho código debe ser interpretado a posteriori.
    - Liberación de memoria: En C está a cargo del programador (delete). En Java es automática; los objetos creados son destruidos automáticamente cuando se queda sin referencias. Un proceso de la JVM llamado recolector de basura.
    - Punteros: En C existen punteros con una aritmética bien definida, y entre otras cosas, permiten al programador explorar la memoria del computador físico. En Java no existe el concepto de punteros; los objetos son accedidos mediante referencias (una referencia se puede considerar como el propio identificador del objeto); no existe aritmética de referencias.
    - Paquetes: En C no existe tal concepto; lo más aproximado son los archivos de biblioteca. En Java un paquete es una entidad organizativa que permite agrupar clases, interfaces y excepciones.

2. Que es el preprocesador y que salida genera

Es el primer programa invocado por el compilador y procesa directivas como #include, #define e #if. Estas directivas no son específicas de C.
El preprocesador utiliza 4 etapas denominadas Fases de traducción:

- Tokenizado léxico : El preprocesador reemplaza la secuencia de trigrafos por los caracteres que representan.
- Empalmado de líneas: Las líneas de código que continúan con secuencias de escape de nueva línea son unidas para formar líneas lógicas.
- Tokenización: Reemplaza los comentarios por espacios en blanco. Divide cada uno de los elementos a preprocesar por un carácter de separación.

- Expansión de macros y gestión de directivas: Ejecuta las líneas con directivas de preprocesado incluyendo las que incluyen otros archivos y las de compilación condicional. Además expande las macros.

3. Mencionar si es correcto o no

a) Es correcto, ya que el tipo char en c tiene un valor integer así que la operación se resuelve sin problemas y en este caso lo que regresa el programa es 0 no el resultado de la operación.
b) El error se encuentra en la VARIABLE =1; porque VARIABLE es una constante que no cambia.
c) El error es que eq debe declararse antes de la función principal.

4. Pasaría sin error por el preprocesador y compilador?

Por el preprocesador pasaría sin error, porque no encontraría un error al momento de definir las cosas, pero en el compilador manda error por que no encuentra la definición de PI para eso se tiene que definir _MIMACRO_ antes de la etiqueta #ifdef