

# Sistemas Operativos

## Tarea 1

Martin Felipe Espinal Cruces

July 2, 2021

### 1. ¿Qué diferencias hay entre Java y C?

La principal diferencia radica en que Java es un lenguaje de programación orientada a objetos mientras que C es un lenguaje de programación estructurada.

Otra de las diferencias se encuentra en la velocidad de cada uno de ellos, ya que el compilador de C convierte el lenguaje de alto nivel a código objeto que son instrucciones que directamente el ordenador puede ejecutar e incluso se puede optimizar aún más su ejecución con ayuda de algunas banderas que se pasan por la línea de comandos. Por otro lado tenemos que Java a pesar de que también se compila, el resultado de dicha compilación es un código entendible para su intérprete Java Virtual Machine.

Pero no todo es malo en Java ya que a pesar de no ser tan rápido como C, el proceso de convertir el código de alto nivel a bytecode le permite ser ejecutado en todo dispositivo siempre y cuando cuente con el JVM instalado. Mientras que para lograr lo mismo en C tendríamos que escribir en ANSI C (C estándar), lo cual sería un poco más difícil.

Volviendo al tema de la velocidad, otro punto a favor que tiene C es el uso de apuntadores, el cual te permite acceso libre a la memoria con lo cual podrás desarrollar de una manera más optimizada mientras que Java a pesar de tener una sintaxis más amigable que la de C, no permite gestionar libremente la memoria.

Finalmente otra diferencia a destacar es el recolector de basura en Java, el cual te permite liberar memoria de manera automática, mientras que en C es el programador el encargado de liberar memoria que ya no use, de lo contrario seguirá utilizada dicha sección de memoria.

### 2. ¿Qué es el preprocesador y qué salida genera?

Es un programa invocado por el compilador que se encarga de procesar y sustituir las directivas, las cuales comienzan con el símbolo `#` y sustituyen todas las apariciones de identificador por el texto de sustitución con la restricción de que dichos textos o Macros funcionen como constantes y por tanto son de carácter inmutable.

Finalmente, la salida que genera el proceso de preprocesador es un fichero con la sustitución textual de las Macros en todo el código.

### 3. Para cada secuencia de instrucciones, indica si es correcto o no; en caso de que no lo sea, explica la razón

(a) -

```
int main() {  
    long int a = 34 ;  
    char b = 'A' ;  
    long c = a * b ;  
    return 0 ;  
}
```

Es correcto ya que al realizar la multiplicación de la variable **a** que es de tipo long int con la variable **b** que es de tipo char el cual para realizar la multiplicación de **a** con **b** toma el valor de ASCII, que en este caso es 65.

(b) -

```
#include "stdio.h"
#define EQ(x,y) (x == y)
int main() {
    unsigned int a = -1 ;
    unsigned int b = a << 1 ;
    if (eq(a,b) == EQ(a,b)) {
        printf("Pase con %d y %d", a, b) ;
    }
    return 0 ;
}
int eq(int a, int b) {
    return a == b ;
}
```

Es incorrecto por varias razones. La primera es que la biblioteca stdio no es una cabecera y por ende debería estar entre diamantes, el segundo error se encuentra en la variable **a** ya que está declarada como un entero sin signo sin embargo se le asigna el valor de -1, otra de las razones por la cual es incorrecto es que la función **eq** que recibe dos enteros no tiene una declaración antes del main

(c) -

```
#define VARIABLE 777
int main() {
    if (VARIABLE) {
        VARIABLE = 1 ;
    }
    for (;;) {
        return 0 ;
    }
}
```

Es incorrecto ya que la variable **VARIABLE** definida en la parte superior no puede ser reasignada.

4. Escribe un programa en C que calcule el área de una figura geométrica (círculo, triángulo y cuadrilátero, que puede ser cuadrado o rectángulo).

El programa se ejecutará como sigue:

```
$ ./programa < bandera > < datos > +
```

< bandera > indica el tipo de figura geométrica: -c círculo, -t triángulo, -dc cuadrado y -dr rectángulo.

Por ejemplo, si se desea calcular el área de un triángulo el programa debe invocarse así:

```
$ ./programa -t < base > < altura >
```

Si el programa recibe parámetros incorrectos debe enviar un mensaje indicando la sintaxis correcta.

Para desarrollar el programa utiliza directivas del preprocesador, también macros que funcionan como objetos y como funciones. Separa el código de tal forma que en un archivo se encuentren las funciones que implementes, incluido a **main()** y en otro archivo todas las definiciones (macros, bibliotecas de C que utilices y prototipos de funciones).

5. Escribe un programa en C que imprima el mensaje **Hola mundo!**

Revisar el archivo resultante (el que contiene el programa ejecutable) y reporta su tamaño en bytes. Este archivo es la versión que carga las funciones de biblioteca en forma dinámica.

Vuelve a compilar el programa indicando que quieres la versión que carga las bibliotecas en forma estática. Reporta el tamaño del archivo resultante.

Explica la razón de la diferencia en los tamaños.

Explica por qué crees que el archivo generado tiene el tamaño que reportas.

Sugerencia: ver la conferencia que les recomendé el 10 de junio.

**Archivo biblioteca dinámica:**

Tamaño en bytes: 49K

**Archivo biblioteca estática:**

Tamaño en bytes: 752B

**Explicación:** Cuando se compila un programa en C, antes de que se convierta en el programa ejecutable final, los archivos de código fuente individuales se convierten en código objeto, almacenado

como archivos objeto. El código de objeto es código de máquina, pero aún no es ejecutable. Una vez que se crean los archivos de objeto, un programa llamado enlazador puede actuar sobre diferentes archivos de objeto para formar un programa, o se pueden consolidar muchos archivos de objeto en una biblioteca estática. Luego, el usuario puede hacer referencia a este archivo de biblioteca consolidado en la futura compatibilidad de sus programas, en lugar de tener que preocuparse por hacer referencia a todas las dependencias del programa individualmente

6. Elige una de las tres siguientes computadoras:

- (a) **IBM 360**
- (b) **VAX 780**
- (c) **burrroughs 6700**

**Justifica tu elección.**

La razón de elegir este computador fue que a primer instancia la confundí con la IBM-650 que por lo que recuerdo de mis clases de arquitectura de computadoras fue la primer computadora que tuvo la U.N.A.M.

**Describe las características principales de la arquitectura de la computadora elegida.**

Al ser una computadora perteneciente a la tercer generación la arquitectura de las IBM-360 contaba con circuitos integrados, su memoria RAM variaba entre 8 KB y 8 MB. El tamaño fue considerablemente menor a sus predecesoras.

**Describe cuál es el sistema operativo que se usaba en esa computadora y cuáles son sus características. De los tipos sistemas operativos visto en clase ¿a cuál corresponde?**

Con el nombre de CP/CMS, era un sistema operativo desarrollado por IBM con la innovación de ser de los primeros sistemas operativos en definir la arquitectura de máquina virtual. Este sistema operativo pertenece a la categoría de sistema monolítico.

**Si gustas, puedes proponer otra computadora, el requisito es que haya sido ampliamente usada entre 1960 y 1980.**