

## **Sistemas Operativos**

### **Tarea 1**

#### **Preguntas:**

##### **1.- *¿Qué diferencia hay entre Java y C?***

Java realiza una asignación de memoria de manera automática y también realiza un desalojo de memoria automático por medio de un recolector de basura. Java es del paradigma orientado a objetos. Puede ser tanto interpretado como compilado.

En C por otro lado, se tiene que hacer la asignación y el desalojo de memoria de manera manual por medio de apuntadores a la memoria. C es un lenguaje imperativo. Su compilación pasa por tres fases, preprocesado, compilación y enlazado.

##### **2.- *¿Qué es el preprocesador y cuál es la salida?***

Es la primera fase por la que pasa el código de c para su compilación. Procesa e interpreta las directivas de preprocesamiento, las cuales siempre son precedidas por un "#"; reemplaza todos los #include por el contenido del fichero especificado, #define es reemplazado en el código por el valor especificado, esto si el identificador no se encuentra entre comillas. Su salida es el código fuente preprocesado. Además de esto, también se encarga de eliminar los comentarios.

##### **3.-**

##### **a) El código es correcto.**

a)

```
1 int main(){
2     long int a = 34;
3     char b = 'A';
4     long c = a * b;
5     return 0;
6 }
```

**b) No es correcto.** Es incorrecto pues después del preprocesamiento VARIABLE se sustituye por el valor que se le definió arriba, entonces lo que realmente está ocurriendo es  $777 = 1$ , según el compilador se necesita de un lvalue a la izquierda.

```

1 #define VARIABLE 777
2 int main(){
3     if(VARIABLE){
4         VARIABLE = 1;
5     }
3     for(;;){
4         return 0;
5     }
6 }

```

c) No es correcto. Se hace uso de la función eq antes de haberla declarado.

c)

```

1 #include "stdio.h"
2 #define EQ(x,y) (x==y)
3 int main(){
4     unsigned int a = -1;
5     unsigned int b = a << 1;
6     if(eq(a,b) == EQ(a,b)){
7         printf("Pase con %d y %d",a,b);
8     }
9     return 0;
10 }
11 int eq(int a, int b){
12     return a == b;
12 }

```

4.- No pasará del preprocesador, pues tendrá el error de que no encontrará a PI cuando se utilice en el código. Pues el macro MIMACRO aún no está definido, por lo que PI nunca se define.

```

1 #ifdef _MIMACRO_
2 #define PI 3.1416
3 #endif
4 #define CONFIRMA(X) (X > 40.0)
4 #define MIMACRO
6 int reconfirma(float n);
5 int main(){
6     //radio del circulo
7     float radio = 3.4;

```

```
9 //area del circulo
8 float area = PI * radio * radio;
10 while(1){
11     /* si el area es menor que 40
12     va aumentando el area,
13     sino rompe el ciclo y termina */
14     if(area < 40.0){
15         area += 4.4;
16         continue;
17     }
18     break;
19 }
20 int resultado = reconfirma(area);
21 return 0;
22 }
23 //funcion para reconfirmar
23 int reconfirma(float n){
24     return CONFIRMA(n);
25 }
```