Seminario de lenguajes
Opción C

Práctica 1 - 2022

1. A continuación se nombran los distintos procesos involucrados en la generación de un ejecutable a partir de uno o varios fuentes. Con la lista de nombres, establezca el orden en que suceden durante el proceso de generación de un ejecutable y describa qué es lo que ocurre en cada uno de los pasos

    1. Proceso de linker o enlazador
    2. Generación de código ensamblador
    3. Código fuente: considere que existe más de un fuente que compone al programa y se emplean librerías del sistema
    4. Generación de código objeto relocalizable
    5. Ejecutable
    6. Preprocesador del fuente

2. Analice los tipos de productos mencionados a continuación en el mercado, considerando en todos los casos el tipo de licencia empleada por cada producto nombrado:

    1. Editores de código C
    2. Compiladores C
    3. Debuggers de código C
    4. IDE (Integrated Development Environment o Entorno de Desarrollo Integrado)

3. Compile con el compilador **GCC** el famoso *Hello World*. Llamaremos al archivo **hello.c**.

```c
#include <stdio.h>

int main() {
   printf("Hello world\n");
}
```

    (a) Para obtener el ejecutable llamado **hello.exe**, compile el código empleando:

```
gcc -o hello.exe hello.c
```

    (b) Pruebe el ejecutable **hello.exe**

    (c) Agregue la opción **-Wall** a la línea mencionada en (a). Esta opción hace que el compilador nos indique todos los *warnings*. **Los *warnings* son importantes de corregir en todo programa**.

```
gcc -Wall -o hello.exe hello.c
```

    (d) Solucione cualquier *warning* mencionado por el compilador hasta que la compilación usando la opción **-Wall** no devuelva ningún *warning*.

(e) Analizaremos ahora otras opciones del compilador. Para cada una (**-E**, **-S** y **-c**) indique qué es lo que hace.

Opción 1

```
gcc -E hello.c
```

Opción 2

```
gcc -S hello.c
```

Opción 3

```
gcc -c hello.c
```

4. Analice el siguiente caso de compilación de un ejecutable compuesto de varios archivos. Los archivos son:

main.c

```c
#include <stdio.h>
#include "matem.h"
int main()
{
  int desde = 1;
  int hasta = 10;
  int resultado = sumatoria(desde,hasta);

  printf("La sumatoria de %d a %d da: [ %d ]\n", desde, hasta, resultado);

  return 0;
}
```

matem.h

```c
int sumatoria(int, int);
```

matem.c

```c
#include "suma.h"

int sumatoria(int desde, int hasta)
{
       int sum=0;
       for (;desde<=hasta; desde++) {
              sum = sumar(sum, desde);
       }
       return sum;
}
```

suma.h

```c
int sumar(int, int);
```

suma.c

```c
int sumar(int op1, int op2) {
      return op1 + op2;
}
```

(a) Compile todos los archivos en un único ejecutable, empleando:

```
gcc -o main.exe -Wall main.c suma.c matem.c
```

(b) Ahora puede realizar lo mismo que en el punto anterior pero en dos pasos, empleando:

Paso 1: crea archivos objeto (.o)

```
gcc -Wall -c main.c suma.c matem.c
```

Paso 2: enlaza archivos en un único ejecutable

```
gcc -Wall -o main.exe main.o suma.o matem.o
```

5. Indique qué es lo que retornan cada una de las siguientes funciones.

funciones.c

```c
#include <limits.h> // Para usar UINT_MAX
char uno()
{
   char c;
   c = 'A' + 2;

   return (c);
}

int dos()
{
   int a = 10;

   return ((a++ == 10)?a:--a);
}

float tres()
{
   float x = (int) 3.7 + 4.5;

   return x;
}

int cuatro()
{
   float x=3.6;

   return (int)x;
```

```c
}

int cinco()
{
   int a = 17, b;
   b = ++a % 3;

   return (b > 1);
}

int seis()
{
   int a = 3;
   a <<= 3;

   return a;
}

int siete ()
{
   // Ver:
   // https://www.gnu.org/software/libc/manual/html_node/Range-of-Type.html
   unsigned a = 8, b;
   b = ~a;

   return (b == (UINT_MAX - 8));
}

int ocho()
{
   int i = 320;

   return (char) i;
}

double nueve()
{
   char c = 'A';

   return (double) c;
}
```

6. Escriba una función que reciba un número y retorne, si es posible, el carácter correspondiente al código ASCII determinado por dicho número.

7. Escriba una función que reciba dos enteros sin signo y retorne la división entre el mayor de ellos y el menor.

8. Escriba una función que realice la suma de los primeros n números naturales pares. El número n se recibe como parámetro.

   (a) Escriba un programa que imprima la suma de los primeros 250 números pares. (invoque a la función anterior)

9. Escriba una función que dado un carácter devuelva si es un dígito o no.

10. Escriba una función que dado un carácter devuelva si es mayúscula o minúscula.

11. Escriba una función que dado un carácter retorne su mayúscula, si está en minúscula y el mismo carácter en caso contrario.

12. Escriba una función que reciba dos enteros sin signo y devuelva el promedio

13. Escriba una función que reciba un entero y devuelva su factorial.

14. Escriba una función que reciba dos enteros y devuelva el menor de ellos. Realizarlo utilizando una sola sentencia dentro de la función.

15. ¿Qué es lo que hace el operador **sizeof**?