



Fundamentos del Software

Ejercicio 2 - Módulo II

Plazo de entrega: 27 de mayo de 2020

1 Instrucciones

Los ejercicios a resolver deberán estar adecuadamente documentados. Se entregará un archivo PDF con la memoria, donde se incluirá el enunciado del ejercicio seguido de su solución con los comandos en formato texto utilizados y las capturas de pantalla necesarias para demostrar que la ejecución la ha realizado el estudiante.

2 Enunciado

Ejercicio 1.

Utilizando los archivos disponibles en la carpeta de la sesión 8 (*main.cpp*, *factorial.cpp* y *hello.cpp*). Ejecuta la siguiente orden:

```
g++ main.cpp factorial.cpp hello.cpp -o ejecutable
```

Una vez compilado el código, utiliza la herramienta **gdb** para ver el código del programa principal. Describe qué sucede al intentar ver las líneas del código del *main* y cómo harías para ver el código por medio del depurador. Indica cómo se puede salir del depurador.

Ejercicio 2.

¿Cómo harías para ejecutar la función factorial del ejercicio anterior? Muestra una captura de pantalla indicando el resultado y cómo has llegado a él.

Ejercicio 3.

Compila el código del siguiente programa y haz todas las configuraciones necesarias con **gdb** para mostrar el valor de la variable **final2** justo antes de que se le asigne ningún valor dentro del bucle *for*. Describe todos los pasos que has seguido e incluye una captura de pantalla de los mismos.



Código:

```
#include <iostream>

/* Incrementa en 2 una variable */
int cuenta (int y) {
    int tmp;
    tmp = y + 2;
    return tmp;
}

/* Calcula la multiplicación de dos números */
int multiplica (int x, int y) {
    int final;
    int i;

    final = 0;
    for (i = 0; i < x; i++) {
        final = final + y;
    }

    return final;
}

int main (void) {
    int final1;
    int final2;
    int i;

    final1 = multiplica(3, 2);

    for (i = 0; i < 100; i++)
        final2 = cuenta(i);

    std::cout << final1 << "\n";

    return 0;
}
```

Ejercicio 4.

Siguiendo el ejercicio anterior, haz todas las configuraciones necesarias utilizando el depurador **gdb** para obtener el valor de la variable **final2** cuando **i** vale 50. Muestra todos los comandos utilizados y el valor de las variables **final2** e **i**.

Ejercicio 5.

Con el mismo código del ejercicio anterior no parece que la variable **final2** pueda ser mayor de 101. Utilizando el depurador **gdb** haz, que sin tocar la variable **final2**, esta variable tenga un valor por encima de 1000 al final del programa. Describe todos los pasos que has seguido para conseguirlo e incluye la captura de pantalla correspondiente.