

# Ejercicio Práctica 8

Salvador Romero Cortés

## 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
```

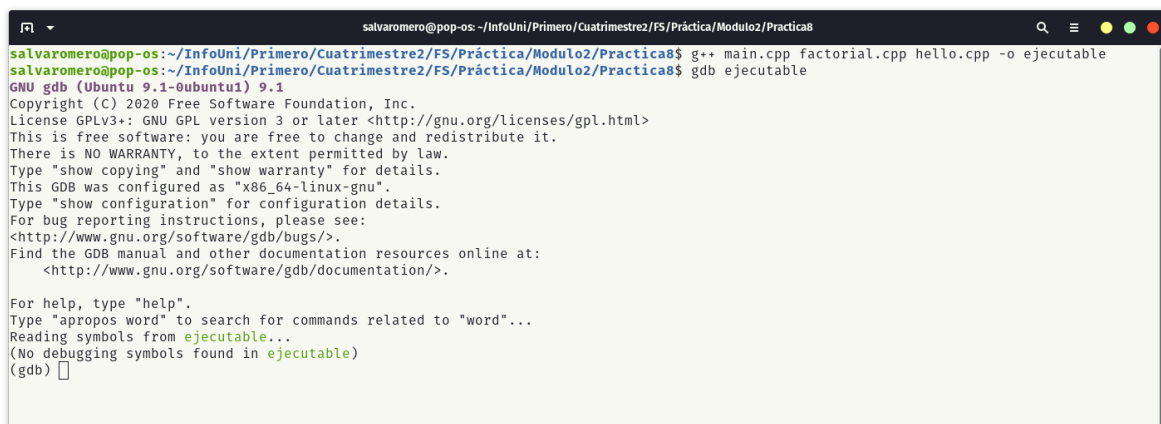
Una vez compilado el código, utiliza la herramienta **gdb** para ver el código del programa principal. Describe qué sucede al intentar ver las líneas del código del main y cómo harías para ver el código por medio del depurador. Indica cómo se puede salir del depurador.

Ejecución de la línea



```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$ g++ main.cpp factorial.cpp hello.cpp -o ejecutable
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$
```

Ejecutamos gdb



```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$ g++ main.cpp factorial.cpp hello.cpp -o ejecutable
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$ gdb ejecutable
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejecutable...
(No debugging symbols found in ejecutable)
(gdb)
```

Intentamos ver las líneas del código de main

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$ g++ main.cpp factorial.cpp hello.cpp -o ejecutable
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$ gdb ejecutable
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejecutable...
(no debugging symbols found in ejecutable)
(gdb) list
No hay tabla de simbolos cargada. Use la orden «file».
(gdb) █
```

Da error, porque al compilar no hemos incluido información de depuración con el parámetro `-g`.

Para poder ver las líneas, volvemos a compilar el programa con el parámetro `-g`. Quedaría así la orden:

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

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$ g++ -g main.cpp factorial.cpp hello.cpp -o ejecutable
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$ gdb ejecutable
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejecutable...
(gdb) █
```

Como podemos ver, ahora no da error.

Para ver el código de main usamos la orden sentencia `list`.

```
list main
```

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejecutable...
(gdb) list main
1      #include <iostream>
2      #include "functions.h"
3
4      using namespace std;
5
6      int main(){
7          print_hello();
8          cout << endl;
9          cout << "The factorial of 7 is " << factorial(7) << endl;
10         return 0;
(gdb) █
```

Ahí podemos ver las 10 primeras líneas de la función main, podemos ver más líneas si añadimos al final de la orden list un rango de líneas. Por ejemplo:

```
list 1,11
```

```
salvaromero@pop-os: ~/InfoUni/Primero/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejecutable...
(gdb) list 1,11
1      #include <iostream>
2      #include "functions.h"
3
4      using namespace std;
5
6      int main(){
7          print_hello();
8          cout << endl;
9          cout << "The factorial of 7 is " << factorial(7) << endl;
10         return 0;
11     }
(gdb)
```

Para cerrar el depurador gdb, escribimos la orden `quit` (también podemos usar `q`)

```
salvaromero@pop-os: ~/InfoUni/Primero/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejecutable...
(gdb) list 1,11
1      #include <iostream>
2      #include "functions.h"
3
4      using namespace std;
5
6      int main(){
7          print_hello();
8          cout << endl;
9          cout << "The factorial of 7 is " << factorial(7) << endl;
10         return 0;
11     }
(gdb) quit
salvaromero@pop-os: ~/InfoUni/Primero/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$
```

## 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.*

Para ejecutar la función factorial podemos poner un breakpoint en su llamada. Para ello usamos la orden `break`. Podemos usar el nombre de la función o la línea.

```
break factorial
```

A continuación ejecutamos el programa con la orden `run`. Y podemos ejecutar la función factorial paso a paso usando la orden `next`.

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejecutable...
(gdb) break factorial
Punto de interrupción 1 at 0x12a2: file factorial.cpp, line 3.
(gdb) run
Starting program: /home/salvaromero/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8/ejecutable
Hello World!

Breakpoint 1, factorial (n=32767) at factorial.cpp:3
3   int factorial(int n){
(gdb) next
4       if(n!=1){
(gdb) next
5           return(n * factorial(n-1));
(gdb) next

Breakpoint 1, factorial (n=0) at factorial.cpp:3
3   int factorial(int n){
(gdb) next
4       if(n!=1){
(gdb) next
5           return(n * factorial(n-1));
(gdb) next

Breakpoint 1, factorial (n=21845) at factorial.cpp:3
3   int factorial(int n){
(gdb) next
4       if(n!=1){
(gdb) next
5           return(n * factorial(n-1));
(gdb) next

Breakpoint 1, factorial (n=21845) at factorial.cpp:3
3   int factorial(int n){
(gdb) next
4       if(n!=1){
(gdb) next
5           return(n * factorial(n-1));
(gdb) next

Breakpoint 1, factorial (n=0) at factorial.cpp:3
3   int factorial(int n){
(gdb) next
4       if(n!=1){
(gdb) next
5           return(n * factorial(n-1));
(gdb) next

Breakpoint 1, factorial (n=32767) at factorial.cpp:3
3   int factorial(int n){
(gdb) next
4       if(n!=1){
(gdb) next
5           return(n * factorial(n-1));
(gdb) next

Breakpoint 1, factorial (n=21845) at factorial.cpp:3
3   int factorial(int n){
(gdb) next
4       if(n!=1){
(gdb) next
7       else return 1;
(gdb) next
8       }
(gdb) next
factorial (n=2) at factorial.cpp:5
5       return(n * factorial(n-1));
(gdb) next
8       }
(gdb) next
factorial (n=3) at factorial.cpp:5
5       return(n * factorial(n-1));
(gdb) next
8       }
(gdb) next
factorial (n=4) at factorial.cpp:5
5       return(n * factorial(n-1));
(gdb) next
8       }
(gdb) next
factorial (n=5) at factorial.cpp:5
5       return(n * factorial(n-1));
(gdb) next
8       }
(gdb) next
factorial (n=6) at factorial.cpp:5
5       return(n * factorial(n-1));
(gdb) next
8       }
(gdb) next
factorial (n=7) at factorial.cpp:5
5       return(n * factorial(n-1));
(gdb) next
8       }
(gdb) next
The factorial of 7 is 5040
main () at main.cpp:10
10     return 0;
(gdb) □
```

Aquí vemos como se ejecuta la función factorial. Al haber usado `break factorial`, aparecen paso a paso todas las llamadas a factorial. Si quisieramos aun más pasos podríamos haber utilizado la orden `step` en lugar de `next`.

### 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.

Compilamos el programa

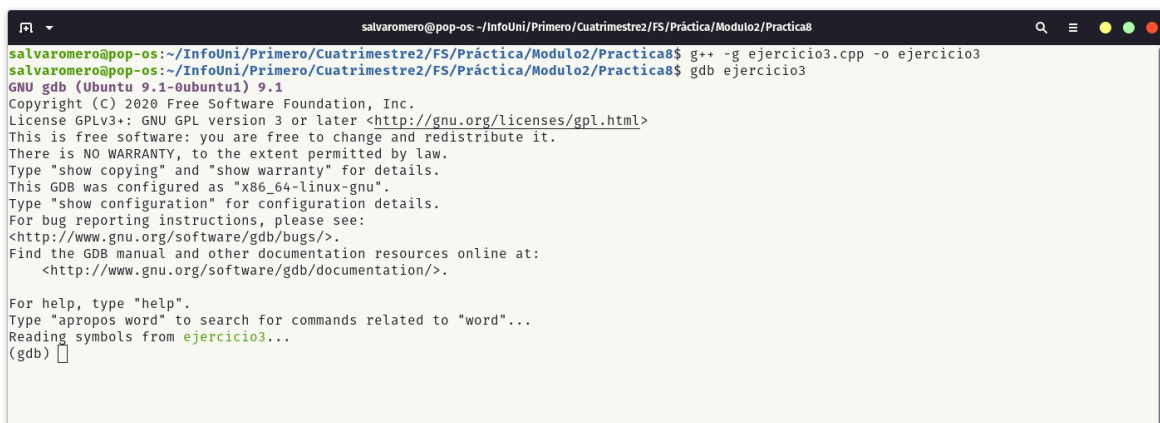
```
g++ -g ejercicio3.cpp -o ejercicio3
```



```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimstre2/FS/Práctica/Modulo2/Practica8
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimstre2/FS/Práctica/Modulo2/Practica8$ g++ -g ejercicio3.cpp -o ejercicio3
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimstre2/FS/Práctica/Modulo2/Practica8$
```

Ejecutamos gdb

```
gdb ejercicio3
```



```
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimstre2/FS/Práctica/Modulo2/Practica8$ g++ -g ejercicio3.cpp -o ejercicio3
salvaromero@pop-os:~/InfoUni/Primer/Cuatrimstre2/FS/Práctica/Modulo2/Practica8$ gdb ejercicio3
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb)
```

Para ello ponemos un punto de ruptura donde aparece la variable **final2**. Lo podemos ver con `list`.

```
list 1,34
```

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimetre2/FS/Práctica/Modulo2/Practica8
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) list 1,34
1      #include <iostream>
2
3      /* Incrementa en 2 una variable */
4      int cuenta (int y) {
5          int tmp;
6          tmp = y + 2;
7          return tmp;
8      }
9
10     /* Calcula la multiplicación de dos números */
11     int multiplica (int x, int y) {
12         int final;
13         int i;
14         final = 0;
15         for (i = 0; i < x; i++) {
16             final = final + y;
17         }
18         return final;
19     }
20
21     int main (void) {
22         int final1;
--Type <RET> for more, q to quit, c to continue without paging--
23         int final2;
24         int i;
25
26         final1 = multiplica(3, 2);
27
28         for (i = 0; i < 100; i++)
29             final2 = cuenta(i);
30
31         std::cout << final1 << "\n";
32
33         return 0;
34     }
(gdb) □
```

Vemos que es en la línea 29, ponemos un punto de ruptura ahí.

```
break 29
```

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimetre2/FS/Práctica/Modulo2/Practica8
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimetre2/FS/Práctica/Modulo2/Practica8$ g++ -g ejercicio3.cpp -o ejercicio3
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimetre2/FS/Práctica/Modulo2/Practica8$ gdb ejercicio3
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) □
```

Ahora ejecutamos el programa y cuando se detenga en el punto de ruptura mostramos el valor de la variable. Usamos las órdenes `run` para ejecutar el programa y la orden

```
display final2
```

para mostrar la variable.

Ejecutamos el programa y se detiene en el punto de ruptura:

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimetre2/FS/Práctica/Modulo2/Practica8
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) run
Starting program: /home/salvaromero/InfoUni/Primer/Cuatrimetre2/FS/Práctica/Modulo2/Practica8/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:29
29         final2 = cuenta(i);
(gdb) □
```

Ahora usamos `display`

```
salvaromero@pop-os: ~/InfoUni/Primero/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) run
Starting program: /home/salvaromero/InfoUni/Primero/Cuatrimestre2/FS/Práctica/Modulo2/Practica8/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:29
29      final2 = cuenta(i);
(gdb) display final2
1: final2 = 0
(gdb)
```

Por tanto, podemos ver que la variable **final2** tiene el valor 0 antes de ser asignado otro por el bucle for.

## Ejercicio 4

Usamos un punto de ruptura condicional que detenga el programa cuando la variable *i* valga 50.

```
break 29 if i==50
```

Una vez creado el punto de ruptura, ejecutamos con la orden `run` y vemos los valores de las variables con la orden `display`.

```
salvaromero@pop-os: ~/InfoUni/Primero/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$ gdb ejercicio3
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29 if i==50
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) run
Starting program: /home/salvaromero/InfoUni/Primero/Cuatrimestre2/FS/Práctica/Modulo2/Practica8/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:29
29      final2 = cuenta(i);
(gdb) display final2
1: final2 = 51
(gdb) display i
2: i = 50
(gdb)
```

Sin embargo, la variable *final2* puede tener dos valores cuando *i* vale 50. Puede tener el valor antes de la asignación del bucle for (mostrado en la captura anterior). O el valor tras la asignación del bucle for. En ambos casos la variable *i* vale 50. Para ver el segundo caso usamos la orden `next`.

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29 if i==50
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) run
Starting program: /home/salvaromero/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:29
29             final2 = cuenta(i);
(gdb) display final2
1: final2 = 51
(gdb) display i
2: i = 50
(gdb) next
28             for (i = 0; i < 100; i++)
1: final2 = 52
2: i = 50
(gdb)
```

Como vemos, en el segundo caso la variable final2 vale 52 mientras que la i sigue valiendo 50.

## Ejercicio 5

Con el mismo código del ejercicio anterior no aparece 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.

Hay varias maneras de hacerlo, yo lo voy a hacer modificando el valor de retorno de la función `cuenta(int)` cuando la i valga 99 (última iteración). Para ello debemos hacer lo siguiente:

- 1. Iniciar el depurador **gdb**

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8$ gdb ejercicio3
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb)
```

- 2. Establecemos un punto de ruptura condicional en la línea 29 (donde se llama a la función cuenta) que se active cuando i valga 99 (última iteración).



```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29 if i==99
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb)
```

- 3. Establecemos otro punto de ruptura al final del programa, donde observaremos el valor de la variable final2.

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29 if i==99
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) break 33
Punto de interrupción 2 at 0x1255: file ejercicio3.cpp, line 33.
(gdb)
```

- 4. Ejecutamos el programa con la orden `run` y luego utilizamos la orden `step` hasta llegar al valor de retorno de la función

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29 if i==99
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) break 33
Punto de interrupción 2 at 0x1255: file ejercicio3.cpp, line 33.
(gdb) run
Starting program: /home/salvaromero/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:29
29         final2 = cuenta(i);
(gdb) step
cuenta (y=98) at ejercicio3.cpp:4
4         int cuenta (int y) {
(gdb) step
6             tmp = y + 2;
(gdb) step
7             return tmp;
(gdb)
```

- 5. Ahora modificamos el valor de `tmp` antes de que lo devuelva. Usamos la orden `set variable`

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29 if i==99
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) break 33
Punto de interrupción 2 at 0x1255: file ejercicio3.cpp, line 33.
(gdb) run
Starting program: /home/salvaromero/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:29
29         final2 = cuenta(i);
(gdb) step
cuenta (y=98) at ejercicio3.cpp:4
4     int cuenta (int y) {
(gdb) step
6         tmp = y + 2;
(gdb) step
7         return tmp;
(gdb) set variable tmp=1001
(gdb)
```

- 6. Continuamos con la ejecución del programa hasta llegar al final, donde habíamos puesto el segundo punto de ruptura

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29 if i==99
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) break 33
Punto de interrupción 2 at 0x1255: file ejercicio3.cpp, line 33.
(gdb) run
Starting program: /home/salvaromero/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:29
29         final2 = cuenta(i);
(gdb) step
cuenta (y=98) at ejercicio3.cpp:4
4     int cuenta (int y) {
(gdb) step
6         tmp = y + 2;
(gdb) step
7         return tmp;
(gdb) set variable tmp=1001
(gdb) continue
Continuando.
6
Breakpoint 2, main () at ejercicio3.cpp:33
33         return 0;
(gdb)
```

- 7. Y ahora usamos la orden `display` para ver cuanto vale **final2**

```
salvaromero@pop-os: ~/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) break 29 if i==99
Punto de interrupción 1 at 0x1222: file ejercicio3.cpp, line 29.
(gdb) break 33
Punto de interrupción 2 at 0x1255: file ejercicio3.cpp, line 33.
(gdb) run
Starting program: /home/salvaromero/InfoUni/Primer/Cuatrimestre2/FS/Práctica/Modulo2/Practica8/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:29
29         final2 = cuenta(i);
(gdb) step
cuenta (y=98) at ejercicio3.cpp:4
4     int cuenta (int y) {
(gdb) step
6         tmp = y + 2;
(gdb) step
7         return tmp;
(gdb) set variable tmp=1001
(gdb) continue
Continuando.
6
Breakpoint 2, main () at ejercicio3.cpp:33
33         return 0;
(gdb) display final2
1: final2 = 1001
(gdb)
```

Conseguimos de esta manera que la variable valga 1001 al final del programa.