

Práctica 8

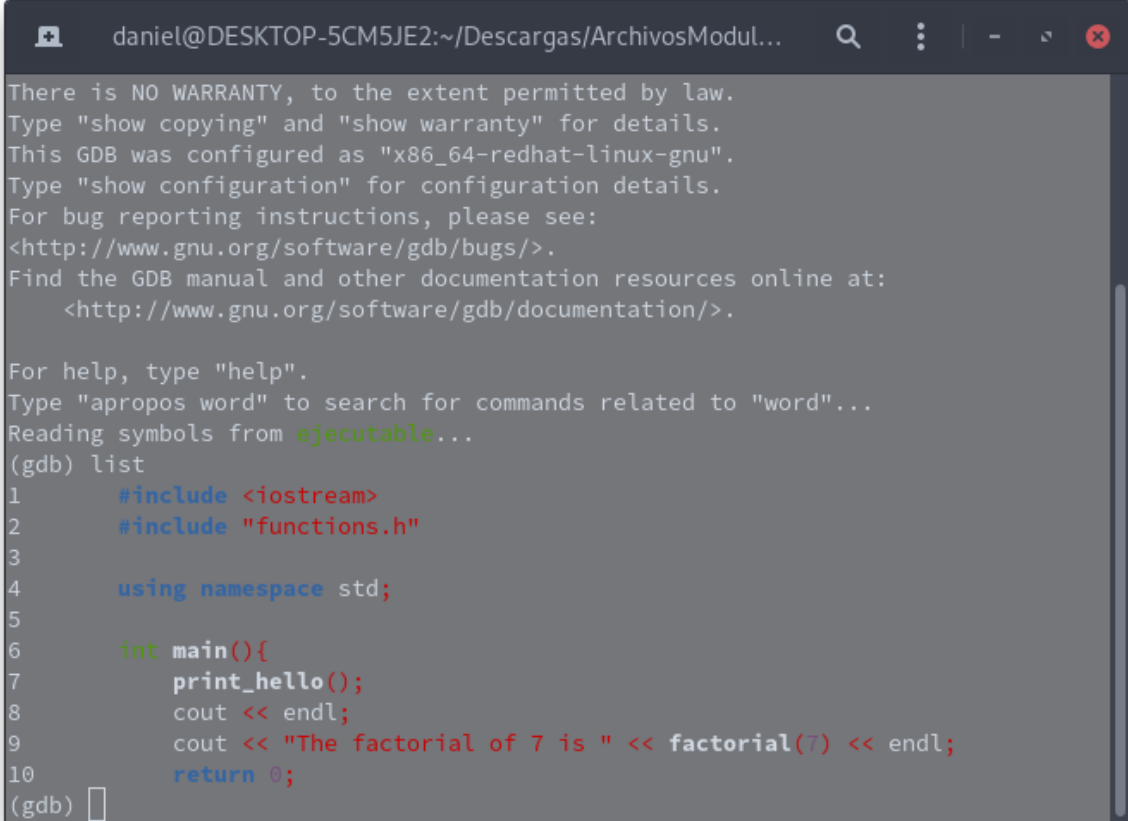
Daniel Alconchel Vázquez

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 de código del *main* y como harías para ver el código por medio del depurador. Indica cómo se puede salir del depurador.



```
daniel@DESKTOP-5CM5JE2:~/Descargas/ArchivosModul...
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-redhat-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      #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) 
```

(Voy a ir pasando capturas de la terminal para verificar que ha sido hecho por mi)

```
[daniel@DESKTOP-5CM5JE2 sesion08]$ g++ -g main.cpp factorial.cpp hello.cpp -o
ejecutable
[daniel@DESKTOP-5CM5JE2 sesion08]$ gdb
Display all 2802 possibilities? (y or n)
[daniel@DESKTOP-5CM5JE2 sesion08]$ gdb ejecutable
GNU gdb (GDB) Fedora 9.1-5.fc32
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-redhat-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  #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) Quit
(gdb)
```

Tras compilar el programa se nos crea un ejecutable. Una vez ejecutado el comando **gdb** sobre dicho ejecutable, utilizamos la orden **list**, que nos muestra por pantalla el código de nuestro archivo fuente, el *main.cpp*.

Mirando el código que nos muestra, vemos que no aparece la llave de cierre del main. Esto se debe a que la orden mencionada anteriormente muestra 10 líneas, por defecto, si no le especificamos con parámetros la cantidad de líneas que nos muestran.

```
(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     }
```

Para detener la ejecución del depurador podemos:

```
(gdb) quit
```

```
(gdb) #Ctrl+Z
```

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*, procederemos a una depuración del main, estableciendo un *breakpoint* la línea de llamada a la función *factorial* (línea 9) por medio del comando **b 9**, equivalente a **break 9**, y, a continuación ejecutamos la depuración utilizando **run**.

```
daniel@DESKTOP-5CM5JE2:~/Descargas/ArchivosModul...
[daniel@DESKTOP-5CM5JE2 sesion08]$
[daniel@DESKTOP-5CM5JE2 sesion08]$ gdb ejecutable
GNU gdb (GDB) Fedora 9.1-5.fc32
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-redhat-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) b 9
Breakpoint 1 at 0x4011a3: file main.cpp, line 9.
(gdb) run
Starting program: /home/daniel/Descargas/ArchivosModulo2/ArchivosModuloII/sesion
08/ejecutable
Missing separate debuginfos, use: dnf debuginfo-install glibc-2.31-2.fc32.x86_64
```

```
daniel@DESKTOP-5CM5JE2:~/Descargas/ArchivosModul...
Starting program: /home/daniel/Descargas/ArchivosModulo2/ArchivosModuloII/sesion
08/ejecutable
Missing separate debuginfos, use: dnf debuginfo-install glibc-2.31-2.fc32.x86_64
Hello World!

Breakpoint 1, main () at main.cpp:9
9      cout << "The factorial of 7 is " << factorial(7) << endl;
Missing separate debuginfos, use: dnf debuginfo-install libgcc-10.1.1-1.fc32.x86
_64 libstdc++-10.1.1-1.fc32.x86_64
(gdb) step
factorial (n=7) at factorial.cpp:4
4      if(n!=1){
(gdb) s
5          return(n * factorial(n-1));
(gdb) s
factorial (n=6) at factorial.cpp:4
4      if(n!=1){
(gdb) s
5          return(n * factorial(n-1));
(gdb) s
factorial (n=5) at factorial.cpp:4
4      if(n!=1){
(gdb) s
5          return(n * factorial(n-1));
```

```
daniel@DESKTOP-5CM5JE2:~/Descargas/ArchivosModul...
4         if(n!=1){
(gdb) s
5             return(n * factorial(n-1));
(gdb) s
factorial (n=4) at factorial.cpp:4
4         if(n!=1){
(gdb) s
5             return(n * factorial(n-1));
(gdb) s
factorial (n=3) at factorial.cpp:4
4         if(n!=1){
(gdb) s
5             return(n * factorial(n-1));
(gdb) s
factorial (n=2) at factorial.cpp:4
4         if(n!=1){
(gdb) s
5             return(n * factorial(n-1));
(gdb) s
factorial (n=1) at factorial.cpp:4
4         if(n!=1){
(gdb) s
7         else return 1;
(gdb) s
```

Como vemos, el programa se ha cargado desde la línea 9. Podemos acceder a la ejecución de la función *factorial* por medio de la orden **step**.

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.

Para mostrar la variable **final2** antes de ser modificada por el bucle, tomaremos un breakpoint en la línea de definición del bucle y mostraremos en valor de la variable mediante la función **print**:

```
daniel@DESKTOP-5CM5JE2:~/Descargas/ArchivosModul...
[daniel@DESKTOP-5CM5JE2 sesion08]$
[daniel@DESKTOP-5CM5JE2 sesion08]$ g++ -g ejercicio3.cpp -o ejercicio3
[daniel@DESKTOP-5CM5JE2 sesion08]$ gdb ejercicio3
GNU gdb (GDB) Fedora 9.1-5.fc32
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-redhat-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) b 30
Breakpoint 1 at 0x4011c6: file ejercicio3.cpp, line 30.
(gdb) run
Starting program: /home/daniel/Descargas/ArchivosModulo2/ArchivosModuloII/sesion
08/ejercicio3
```

```
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

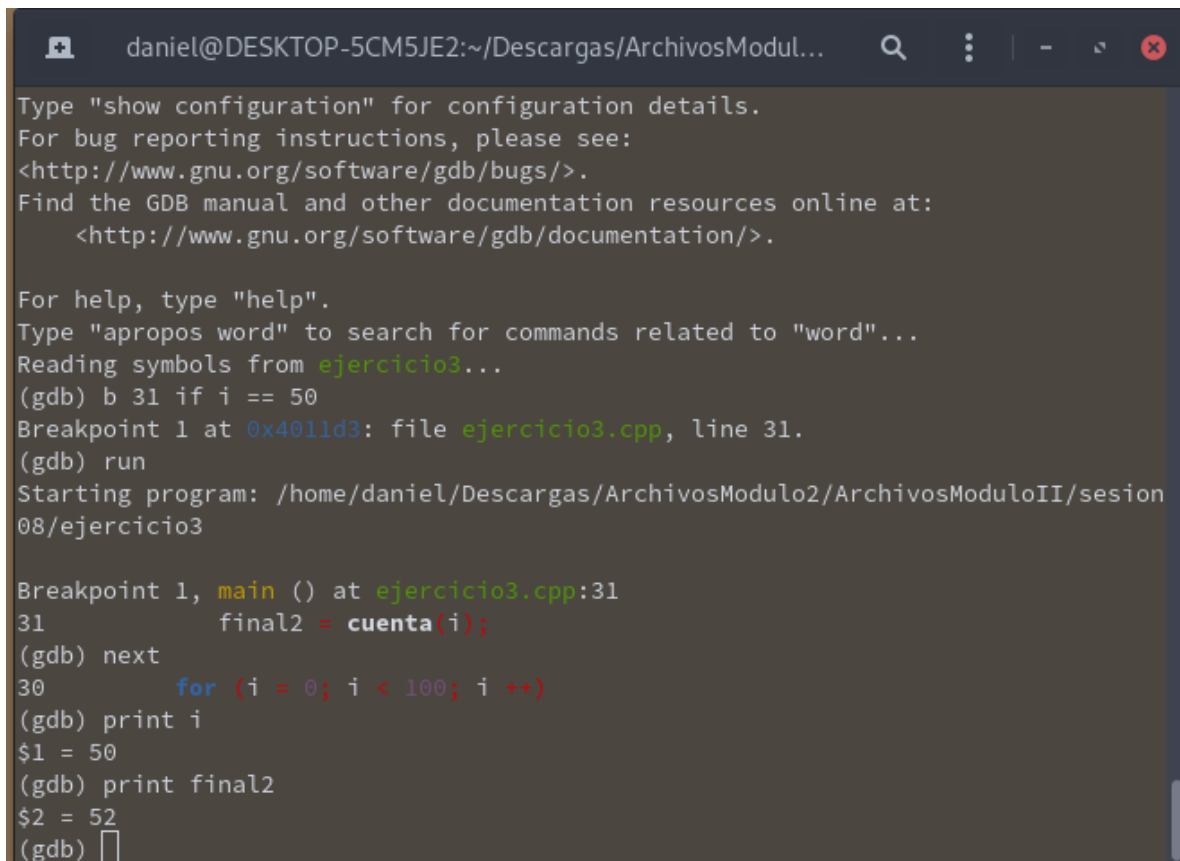
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ejercicio3...
(gdb) b 30
Breakpoint 1 at 0x4011c6: file ejercicio3.cpp, line 30.
(gdb) run
Starting program: /home/daniel/Descargas/ArchivosModulo2/ArchivosModuloII/sesion
08/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:30
30      for (i = 0; i < 100; i++)
(gdb) print final2
$1 = 32767
(gdb) next
31      final2 = cuenta(i);
(gdb) next
30      for (i = 0; i < 100; i++)
(gdb) print final2
$2 = 2
(gdb)
```

Luego, final 2 comienza valiendo 32767 (como no está inicializada pues toma un valor aleatorio) y pasa a valer 2 tras una iteración.

Ejercicio4.

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**.



```
daniel@DESKTOP-5CM5JE2:~/Descargas/ArchivosModul...
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) b 31 if i == 50
Breakpoint 1 at 0x4011d3: file ejercicio3.cpp, line 31.
(gdb) run
Starting program: /home/daniel/Descargas/ArchivosModulo2/ArchivosModuloII/sesion
08/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:31
31      final2 = cuenta(i);
(gdb) next
30      for (i = 0; i < 100; i++)
(gdb) print i
$1 = 50
(gdb) print final2
$2 = 52
(gdb) 
```

El ejercicio es análogo al anterior, pero esta vez usamos **if i == 50** para indicar el valor inicial de **i** en este caso.

Ejercicio5.

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.

Siguiendo la misma metodología que los ejercicios anteriores, establecemos un punto de ruptura antes de finalizar y llamamos a la orden **set \$variable** que nos permite modificar el valor de una variable.

Usamos, además, la orden **display** para hacer un seguimiento de la variable:

```
daniel@DESKTOP-5CM5JE2:~/Descargas/ArchivosModul...
GNU gdb (GDB) Fedora 9.1-5.fc32
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-redhat-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) b 33
Breakpoint 1 at 0x4011e6: file ejercicio3.cpp, line 33.
(gdb) run
Starting program: /home/daniel/Descargas/ArchivosModulo2/ArchivosModuloII/sesion
08/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:33
33      std::cout << final1 << "\n";
```

```
daniel@DESKTOP-5CM5JE2:~/Descargas/ArchivosModul...
Reading symbols from ejercicio3...
(gdb) b 33
Breakpoint 1 at 0x4011e6: file ejercicio3.cpp, line 33.
(gdb) run
Starting program: /home/daniel/Descargas/ArchivosModulo2/ArchivosModuloII/sesion
08/ejercicio3

Breakpoint 1, main () at ejercicio3.cpp:33
33      std::cout << final1 << "\n";
(gdb) display final2
1: final2 = 101
(gdb) set final2=9999
(gdb) next
6
35      return 0;
1: final2 = 9999
(gdb) next
36  }
1: final2 = 9999
(gdb) next
__libc_start_main (main=0x4011ac <main()>, argc=1, argv=0x7fffffffef038,
init=<optimized out>, fini=<optimized out>, rtdl_fini=<optimized out>,
stack_end=0x7fffffffef028) at ../csu/libc-start.c:342
342      exit (result);
```