

GDB Debugger

Para depurar un programa en C a nivel terminal hay que realizar lo siguiente:

1. Escribir un programa con un error a proposito (esto con fines de demostración).
En este caso, devolveremos un factorial erroneo.

```
#include <stdio.h>

int main() {
    int i, num, j;
    printf ("Ingresa un numero: ");
    scanf ("%d", &num );

    for (i=1; i<num; i++)
        j=j*i;

    printf("El factorial de %d es %d\n",num,j);
}
```

2. Compilamos el programa con la opción de debuggear habilitada:

```
gcc -g main.c -o main
```

Esto permite que el compilador recoja la información de depuración. El comando anterior crea el archivo main.out que se utilizará para la depuración y a su vez, una carpeta donde contendra la información del depurador.



3. Ejecutamos GDB

```
gdb nombreArchivo
```

```
GDB TEST|⇒ gdb main
GNU gdb (GDB) 13.2
Copyright (C) 2023 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-apple-darwin22.4.0".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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 main...
Reading symbols from /Users/fernandocazares/Desktop/Escuela/PTI/GDB TEST/main.dSYM/Contents/Resources/DWARF/main...
(gdb)
```

Ahora pondremos un breakpoint. La sintaxis es:

```
break numeroDeLinea
```

```
GDB TEST|⇒ gdb main
GNU gdb (GDB) 13.2
Copyright (C) 2023 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-apple-darwin22.4.0".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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 main...
Reading symbols from /Users/fernandocazares/Desktop/Escuela/PTI/GDB TEST/main.dSYM/Contents/Resources/DWARF/main...
(gdb) break 9
Breakpoint 1 at 0x100003f42: file main.c, line 9.
(gdb)
```

Mientras el programa se ejecuta, el depurador se detendrá en el punto de interrupción y proporcionará el indicador para depurar.

Antes de iniciar el programa, colocaremos el breakpoint donde sospechemos que hay un error.

4. Correremos el programa en el depurador. El comando es

```
run args (si es que los lleva)
```

Es posible comenzar a ejecutar el programa utilizando el comando **run** en el depurador gdb. También se puede proporcionar argumentos de línea de comandos al programa mediante **run args**, donde args, son los argumentos. El programa de ejemplo que usamos aquí no requiere argumentos de línea de comandos, así que con ejecutar run es suficiente.

```
raspberrynando@raspberrypi:~/PT1/GDBtest$ gdb ./main
GNU gdb (Ubuntu 13.1-2ubuntu2) 13.1
Copyright (C) 2023 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 "arm-linux-gnueabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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 ./main...
(gdb) break 10
Breakpoint 1 at 0x61a: file main.c, line 10.
(gdb) run
Starting program: /home/raspberrynando/PT1/GDBtest/main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/arm-linux-gnueabi/libthread_db.so.1".
Ingresa un numero: 8

Breakpoint 1, main () at main.c:10
10      for (i=1; i<num; i++)
(gdb)
```

Podemos ocupar comandos extras como:

```
print [nombreVariable]
```

Es muy intuitivo la manera en como funciona este comando.

```
raspberrynando@raspberrypi:~/PT1/GDBtest$ gdb ./main
GNU gdb (Ubuntu 13.1-2ubuntu2) 13.1
Copyright (C) 2023 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 "arm-linux-gnueabi".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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 ./main...
(gdb) break 12
Breakpoint 1 at 0x638: file main.c, line 13.
(gdb) run
Starting program: /home/raspberrynando/PT1/GDBtest/main
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/arm-linux-gnueabi/libthread_db.so.1".
Ingresa un numero: 7

Breakpoint 1, main () at main.c:13
13      printf("El factorial de %d es %d\n",num,j);
(gdb) print i
$1 = 7
(gdb) |
```

Hay tres tipos de operaciones de gdb que puede elegir cuando el programa se detiene en un punto de interrupción. Continúan hasta el siguiente punto de interrupción, interviniendo o pasando por encima de las siguientes líneas del programa.

- c o continuar: el depurador continuará ejecutándose hasta el siguiente punto de interrupción.
- n o siguiente: el depurador ejecutará la siguiente línea como una sola instrucción.
- s o paso: Igual que el siguiente, pero no trata la función como una sola instrucción, sino que ingresa a la función y la ejecuta línea por línea.

Atajos de comando gdb

Los siguientes atajos para la mayoría de las operaciones frecuentes de gdb son:

- l – lista
- pag – imprimir
- c – continuar
- s – paso
- ENTER: al presionar la tecla Enter se ejecutará nuevamente el comando ejecutado anteriormente.

Comandos de gdb

- **Comando l:**

El comando gdb l o list sirve para imprimir el código fuente en el modo de depuración. Utiliza el número de línea para ver un número de línea específico o para ver una función específica.

- **bt: backtrack**

Imprime el seguimiento de todos los marcos de la pila o los marcos COUNT más internos.

- help

Ver ayuda para un tema de gdb en particular: ayuda TOPICNAME.

- **quit**

Salir del depurador de gdb.