EBook Gratis

APRENDIZAJE GDB

Free unaffiliated eBook created from
Stack Overflow contributors.

#gdb

Tabla de contenido

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: gdb

It is an unofficial and free GDB ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official GDB.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con GDB

Observaciones

GDB (depurador de proyectos GNU) es un depurador de base de línea de comandos que es bueno para analizar programas en ejecución y de núcleo. De acuerdo con el manual del usuario, GDB admite C, C ++, D, Go, Objective-C, Fortran, Java, OpenCL C, Pascal, Rust, assembly, Modula-2 y Ada.

GDB tiene el mismo conjunto de funciones que la mayoría de los depuradores, pero es diferente de la mayoría de los que he usado, ya que todo se basa en los comandos de escritura en lugar de hacer clic en los elementos de la GUI. Algunas de estas características incluyen:

- Establecer puntos de ruptura
- Imprimiendo el valor de las variables.
- Establecer el valor de las variables para probar cosas.
- Viendo la pila

Examples

¿Qué es GDB?

GDB, abreviatura de GNU Debugger, es el depurador más popular para los sistemas UNIX para depurar programas C y C ++. El depurador de GNU, que también se llama gdb, es el depurador más popular para los sistemas UNIX para depurar los programas C y C ++.

GNU Debugger le ayuda a obtener información sobre lo siguiente:

- Si ocurrió un volcado de memoria, ¿en qué declaración o expresión se bloqueó el programa?
- Si se produce un error al ejecutar una función, ¿qué línea del programa contiene la llamada a esa función, y cuáles son los parámetros?
- ¿Cuáles son los valores de las variables del programa en un punto particular durante la ejecución del programa?
- ¿Cuál es el resultado de una expresión particular en un programa?

A partir de GDB

Para iniciar GDB, en el terminal,

```
gdb <executable name>
```

Para el ejemplo anterior con un programa llamado main, el comando se convierte en

```
gdb main
```

Establecer puntos de interrupción

Probablemente querrá que el programa se detenga en algún momento para que pueda revisar la condición de su programa. La línea en la que desea que el programa se detenga temporalmente se denomina punto de interrupción.

```
break <source code line number>
```

Ejecutando tu programa

Para ejecutar su programa, el comando es, como ha adivinado,

```
run
```

Abriendo un núcleo

```
gdb -c coreFile pathToExecutable
```

Trabajando con un archivo Core

Creado este programa realmente malo

```
 #include <stdio.h>
 #include <ctype.h>

// forward declarations


void bad_function()
{

   int *test = 5;

   free(test);

}


int main(int argc, char *argv[])
{
   bad_function();
   return 0;
}


gcc -g ex1.c

./a.out   //or whatever gcc creates
Segmentation fault (core dumped)
```

```
gdb -c core a.out

Core was generated by `./a.out'.
```

Programa finalizado con señal SIGSEGV, fallo de segmentación. # 0 __GI___libc_free (mem = 0x5) en malloc.c: 2929 2929 malloc.c: No existe tal archivo o directorio.

```
(gdb) where
```

0 __GI___libc_free (mem = 0x5) en malloc.c: 2929 # 1 0x0000000000400549 in bad_function () en ex1.c: 12 # 2 0x0000000000000000564 en main (argc = 1, argv = 0x7fffb825bd68)

Desde que compilé con -g puedes ver esa llamada donde me dice que no le gustó el código en la línea 12 de bad_function ()

Luego puedo examinar la variable de prueba que intenté liberar.

```
(gdb) up
```

1 0x0000000000400549 en bad_function () en ex1.c: 12 12 gratis (prueba);

```
(gdb) print test
```

$ 1 = (int *) 0x5

```
(gdb) print *test
```

No se puede acceder a la memoria en la dirección 0x5

En este caso, el error es bastante obvio. Traté de liberar un puntero al que se le asignó la dirección 5 que no fue creada por malloc, por lo que no tiene idea de qué hacer con él.

Lea Empezando con GDB en línea: https://riptutorial.com/es/gdb/topic/4964/empezando-con-gdb

Creditos

S. No	Capítulos	Contributors
1	Empezando con GDB	Community, PSN, Travis