

GDB

GNU Debugger

Franco Rodríguez Fabregues

29 de marzo de 2011

1 Introducción

- Qué significa GDB?
- Qué es GDB?
- Para qué sirve?
- Instalación

2 Primeros pasos

- Compilación
- Ejecutar nuestro programa
- Comandos básicos

3 Ejemplo

- Función mínimo
- Tips

4 Bibliografía

- Bibliografía
- Preguntas



Qué significa GDB?

Las sigla GDB significa GNU DeBugger



Qué es GDB?

GDB es el depurador estándar para el sistema operativo GNU. Es un depurador portable que se puede utilizar en varias plataformas Unix y funciona para varios lenguajes de programación como C, C++ y Fortran. GDB fue escrito por Richard Stallman en 1988. GDB es software libre distribuido bajo la licencia GPL.

Fuente: http://es.wikipedia.org/wiki/GNU_Debugger



Para qué sirve?

GDB, como su nombre lo indica, sirve para *debuggear* un programa. Ofrece la posibilidad de ver el *estado* del programa mientras se está ejecutando.

Esto es, se pueden consultar los valores de las variables, establecer puntos de detención, avanzar en la ejecución paso a paso, etcétera.



Instalación

En los sistemas operativos Ubuntu, se puede instalar directamente desde los repositorios:

Instalación Ubuntu

```
sudo apt-get install gdb
```

Existen distintos *front-ends* que han sido diseñados para GDB como DDD y KGDB.



- 1 **Introducción**
 - Qué significa GDB?
 - Qué es GDB?
 - Para qué sirve?
 - Instalación
- 2 **Primeros pasos**
 - Compilación
 - Ejecutar nuestro programa
 - Comandos básicos
- 3 **Ejemplo**
 - Función mínimo
 - Tips
- 4 **Bibliografía**
 - Bibliografía
 - Preguntas



Compilación

A la hora de compilar nuestro programa, debemos agregarle el flag `-g` para que luego podamos depurarlo con GDB. Esto lo hacemos usando `gcc` o incluyéndolo en el *Makefile*.

Compilando con `gcc`

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



Ejecutar nuestro programa

Supongamos que ya compilamos nuestro programa, y el ejecutable se llama `main`. Entonces lo corremos con:

```
gdb ./main
```



Comandos básicos

- `help`: Muestra algunos comandos a modo de ayuda
- `run`: Ejecuta el programa que queremos depurar
- `break 1`: Genera un *breakpoint* en la línea 1
- `print v`: Muestra el valor de una variable o expresión
- `list`: Imprime la siguiente línea que se va a ejecutar y su contexto
- `next`: Avanza un paso de la ejecución
- `continue`: Avanza hasta el siguiente *breakpoint* o hasta el final
- `step`: Avanza un paso introduciéndose en la función
- `clear 1`: Elimina el *breakpoint* de la línea 1
- `quit`: Sale de GDB



Comandos básicos

Comandos run y break

Si nuestro programa recibe argumentos de entrada:

```
run [arg1] [arg2] ... [argn]
```

Ejemplo run

Si nuestro programa espera dos enteros, podríamos hacer:

```
run 3 5
```

Un *breakpoint* es un punto de ruptura en donde GDB se detendrá y esperará por una nueva orden.



Comandos básicos

Comandos `print` y `list`

El comando `print v` muestra el valor de la variable `v` en ese momento. También puede hacerse `print <exp>` donde `exp` es una expresión.

El contexto que se muestra con el uso de `list`, es generalmente la siguiente línea de ejecución, las cinco líneas anteriores, y las cinco posteriores.



Comandos básicos

Comandos `next`, `continue` y `step`

Los comandos `next` y `step` ejecutan la siguiente línea del programa. Si la próxima línea a ejecutarse es una llamada a función, el comando `next` la ejecutará de forma atómica y luego seguirá con la siguiente sentencia. `step`, en cambio, ejecutará la primera sentencia de dicha función. `continue` avanza con la ejecución del programa tras cualquier tipo de parada.



Comandos básicos

Comandos clear y quit

El comando `clear 1` elimina el *breakpoint* antes creado con `break` en la línea 1. Si se quieren eliminar todos los *breakpoints* a la vez, se puede usar el comando `delete breakpoints`.

Y para saber información de todos los *breakpoints* `info breakpoints`.

`quit` termina con la ejecución del programa y sale de GDB.



- 1 Introducción
 - Qué significa GDB?
 - Qué es GDB?
 - Para qué sirve?
 - Instalación
- 2 Primeros pasos
 - Compilación
 - Ejecutar nuestro programa
 - Comandos básicos
- 3 Ejemplo
 - Función mínimo
 - Tips
- 4 Bibliografía
 - Bibliografía
 - Preguntas



Función Mínimo I

```
int main(void){  
    int x= 0, y= 0, min= 0;  
  
    printf(" Ingrese dos numeros enteros \n" );  
    scanf(" %d %d" , &x, &y);  
  
    min= minimo(x,y);  
  
    printf(" El minimo es %d \n" , min );  
    return 0;  
}
```



Función Mínimo II

```
int minimo(int x, int y){  
    int min= 0;  
  
    if(x>=y){  
        min= y;  
    }else{  
        min= x;  
    }  
    return min;  
}
```



Usando GDB I

```
$ gcc -o main main.c -g
$ gdb -q ./main
(gdb) list
1      #include <stdio.h>
2
3      int minimo(int x, int y);
4
5      int main(void){
6          int x= 0, y= 0, min= 0;
7
8          printf("Ingrese dos numeros enteros\n");
9          scanf("%i %i", &x, &y);
10
(gdb) break 11
Punto de interrupcion 1 at 0x80484bf: file main.c, line 11.
(gdb) run
Starting program: /home/francusa/main
Ingrese dos numeros enteros
3 6
```



Usando GDB II

```
Breakpoint 1, main () at main.c:11
11      min= minimo(x,y);
(gdb) print x
$1 = 3
(gdb) print y
$2 = 6
(gdb) step
minimo (x=3, y=6) at main.c:18
18      int min= 0;
(gdb) next
20      if(x>=y){
(gdb) next
23          min= x;
(gdb) next
25      return min;
(gdb) next
26      }
(gdb)
```



Usando GDB III

```
main () at ejemplo.c:13
13      printf("El minimo es %i\n", min);
(gdb) next
El minimo entre 3 y 6 es 3
14      return 0;
(gdb) next
15      }
(gdb)

Program exited normally.
(gdb) quit
```



Tips

- Para los comandos dados se pueden utilizar las primeras letras nada más, por ejemplo `b 11` es lo mismo que `break 11`
- Al apretar *Enter*, se ejecuta la última instrucción
- Se puede usar `print` con cualquier expresión ejecutable, por ejemplo `p x*5.7`.



- 1 **Introducción**
 - Qué significa GDB?
 - Qué es GDB?
 - Para qué sirve?
 - Instalación
- 2 **Primeros pasos**
 - Compilación
 - Ejecutar nuestro programa
 - Comandos básicos
- 3 **Ejemplo**
 - Función mínimo
 - Tips
- 4 **Bibliografía**
 - Bibliografía
 - Preguntas



Bibliografía

Se puede encontrar bibliografía en la web. La página oficial es
<http://www.gnu.org/software/gdb/documentation/>



Preguntas

????

