

Depuración

Depuración



■ Errores

- ◆ Segmentation fault
- ◆ Bus Error
- ◆ Fallo de memoria

■ Causa principal

- ◆ Acceso a posiciones ilegales de memoria

■ Ejemplos

- ◆ No poner el & al leer un dato con scanf
 - Intentamos almacenar el dato en una dirección errónea
- ◆ Nos salimos de rango en un vector o cadena
- ◆ No asignamos una dirección de memoria válida a un puntero



Depuración



- Errores en tiempo de ejecución
 - ◆ Muy difíciles de detectar en el código
 - ◆ El compilador no nos da información sobre la línea en la que se produce el fallo de memoria

- Solución
 - ◆ Compilar con la opción -g
 - ◆ Utilizar depuradores de código
 - Depurador gdb
 - Depurador kdbg
 - Depurador ddd



Depuración



- Opción de compilación -g
 - ◆ Introduce cierta información para que los depuradores puedan hacer uso de ella y
 - Determinar donde se están produciendo fallos
 - Hacer una traza del programa
- GDB
 - ◆ Depurador en modo texto
 - ◆ Nos permite ver qué ocurre cuando un programa se ejecuta o qué estaba haciendo cuando falló
 - ◆ Nos informa de la línea en la que se produjo el fallo.



Depuración



■ Uso de GDB

- ◆ **Compilar con la opción -g**
 - gcc -g prueba.c -o salida.exe
- ◆ **Ejecutar *gdb <ejecutable>***
 - gdb salida.exe
- ◆ **Opciones dentro del gdb**
 - **run** : ejecutar el programa
 - Si el programa lleva argumentos se pondrán a continuación del run
 - **bt**: backtracking del fallo que produjo al fin del programa
 - Nos da la línea donde se dio el fallo
 - **quit**: salir del depurador
 - **help**: ayuda sobre otras opciones



Ejemplo ejecución gdb



```
mluque@rabinfl149:~/docencia/curso_04-05/MTP/codigos/gdb$ gdb a.out
```

```
GNU gdb 6.3-debian
```

```
Copyright 2004 Free Software Foundation, Inc.
```

```
GDB is free software, covered by the GNU General Public License, and you are  
welcome to change it and/or distribute copies of it under certain  
conditions.
```

```
Type "show copying" to see the conditions.
```

```
There is absolutely no warranty for GDB. Type "show warranty" for details.
```

```
This GDB was configured as "i386-linux"...Using host libthread_db library "  
lib/tls/i686/cmov/libthread_db.so.1".
```

```
(gdb) run
```

```
Starting program: /home/mluque/docencia/curso_04-05/MTP/codigos/gdb/a.out
```

```
Numero de filas: 2
```

```
Numero de columnas: 2
```

```
Program received signal SIGSEGV, Segmentation fault.
```

```
0x08048602 in rellenamMatriz (m=0x80486e8, fil=2, col=2) at ejemplo1.c:68  
68          m[i][j]=rand()%1000;
```

```
(gdb) bt
```

```
#0 0x08048602 in rellenamMatriz (m=0x80486e8, fil=2, col=2) at ejemplo1.c:68  
#1 0x080484a6 in main () at ejemplo1.c:15
```

```
(gdb) Quit
```

```
(gdb)
```

Línea en la que se llamó a la función
en la que se produjo el error

Línea del error



Kdbg



a.out (/tmp/suma.c) - KDbg

File View Execution Breakpoint Window Settings Help

Source code view:

```
+ int main() {  
+   int v[5]={1,2,3,4,5};  
+   int * ptr;  
+   int suma=0;  
+   int i;  
+   int a, b,c;  
+  
+   ptr=v;  
+   for (i=0;i<5;i++){  
+       printf("%p (%d)\n",ptr, *ptr);  
+       suma=suma+ (*ptr++);  
+       printf("%p (%d)\n",ptr, *ptr);  
+   }  
+   printf("%d\n",suma);  
+  
+   a=7;  
+   b=a++;  
+   c=++a;  
+   printf("%d %d %d\n",a,b,c);  
+  
+   return 0;  
+ }
```

Variable Value

+	v	
+	ptr	(int *) 0xb7f25df0
	suma	0
	i	-1074983832
	a	134518324
	b	-1208995852
	c	134513316

Stack Breakpoints Output

main () at suma.c:10

active Line 10