

Curso de programación en C moderno (II Edición)

Neira Ayuso, Pablo Falgueras García, Carlos

Tema 13 **Depuración**

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio

Breakpoints

Paso a paso

Estado del programa

Otros comandos útiles

1 Introducción

2 GDB

3 Cómo utilizar GDB

4 Comandos

- Inicio
- Breakpoints
- Paso a paso
- Estado del programa
- Otros comandos útiles

Introducción

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio
Breakpoints
Paso a paso
Estado del
programa
Otros
comandos
útiles

- **¿Qué es depurar (debugging)?**: El proceso por el cual buscas y/o eliminas fallos (bugs) de tu programa
- **¿Qué es un depurador (debugger)?**: Una herramienta que facilita la depuración al permitir detener tu programa en cualquier instante y mirar que está pasando
- **¿Por qué es importante usarlo?**: . El método tradicional de los printf de depuración está bien, pero se queda corto en cuanto el programa crece en complejidad. En este punto, el depurador se vuelve la manera más eficiente y fácil de depurarlo.

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio
Breakpoints
Paso a paso
Estado del
programa
Otros
comandos
útiles

- Depurador del proyecto GNU
- Un proyecto muy maduro (30 años)
- Repleto de utilidades para depurar
- Capaz de depurar varios lenguajes (C, C++, Ada, Objective-C, Free Pascal, Fortran, Java ...)
- Soporta muchas arquitecturas (x86, x86-64, ARM, AVR, ...)
- Sistema cliente-servidor que permite depuración remota
- Base de muchos depuradores gráficos actuales (CodeBlocs, Eclipse, NetBeans, VisualStudio, QtCreator, ...)

Cómo utilizar GDB

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio

Breakpoints

Paso a paso

Estado del programa

Otros

comandos útiles

- 1 Compilar con símbolos de depuración `gcc -g main.c -o miprog (-ggdb para símbolos específicos de gdb)`
- 2 Arrancar nuestro programa con gdb: `gdb miprog`
- 3 Utilizar los comandos de gdb para depurar

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

- Inicio
- Breakpoints
- Paso a paso
- Estado del programa
- Otros comandos útiles

COMANDOS

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio

Breakpoints

Paso a paso

Estado del

programa

Otros

comandos

útiles

El programa se inicia con el comando `run`. Si nuestro programa recibe parámetros, se los pasamos a `run`:

```
(gdb) run -w 10 -h 15
```

Breakpoints

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio

Breakpoints

Paso a paso

Estado del programa

Otros comandos útiles

Un **breakpoint** define el momento en el que se va a detener nuestro programa para permitirnos examinarlo.

Podemos elegir que se detenga cuando:

- Alcance cierta línea de código:
`(gdb) break main.c:15`
- Entre en una función determinada:
`(gdb) break world_alloc`
- Se modifique cierta variable:
`(gdb) watch contador`

Breakpoints

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio

Breakpoints

Paso a paso

Estado del programa

Otros comandos útiles

Otros comandos:

- Listar los breakpoints (y watchpoints):
`(gdb) info breakpoints`
- Elimiar breakpoint:
`(gdb) delete <num que sale al listar>`
- Elimiar todos los breakpoint:
`(gdb) delete breakpoints`

Paso a paso

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio
Breakpoints
Paso a paso
Estado del
programa
Otros
comandos
útiles

Tenemos varios comandos para realizar una ejecución “paso a paso” de nuestro programa:

- Ejecutar hasta el siguiente breakpoint:
(gdb) `continue`
- Ejecutar una línea (sin entrar en funciones):
(gdb) `next`
- Ejecutar una línea (entrado en funciones):
(gdb) `step`
- Salir de la función actual:
(gdb) `finish`

Estado del programa

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio

Breakpoints

Paso a paso

Estado del programa

Otros comandos útiles

- Ver una variable:
`(gdb) print contador`
- Ver una estructura:
`(gdb) print *world`
- Ver el resultado de una expresión:
`(gdb) print size_x * size_y`
- Ver el resultado de una función:
`(gdb) print world_get_cell(w, 1, 2)`
- Ver las variables locales:
`(gdb) info locals`
- Ver algunas líneas de código de alrededor:
`(gdb) list`
- Ver la pila de llamadas:
`(gdb) backtrace`

Otros comandos útiles

Portada

Introducción

GDB

Cómo utilizar GDB

Comandos

Inicio Breakpoints Paso a paso Estado del programa **Otros comandos útiles**

- Ver clases de comandos:
(gdb) help
- Ver comandos de un clase (por ejemplo, breakpoints):
(gdb) help breakpoints
- Ver ayuda sobre un comando (por ejemplo, break):
(gdb) help break
- Para salir:
(gdb) quit
- Puedes abreviar un comando siempre que no sea ambiguo:
(gdb) help n == (gdb) help next
- Para ejecutar el comando anterior pulsar ENTER
- Para autocompletar (comandos, nombres de funciones y variables, etc) pulsar TAB
- Chuleta: http://lab46.corning-cc.edu/_media/opus/fall2014/mgardne8/70120637.png