

# DEPURACIÓN CON GDB EN C/C++

SERGIO GÓMEZ

## INTRODUCCIÓN

# ¿QUÉ ES UN DEBUGGER?

Aplicación que permite localizar errores en otros programas. Permite al programador detener la ejecución en cualquier punto y examinar el valor de las variables.

## INTRODUCCIÓN

# SÍMBOLOS DE DEPURACIÓN

Necesitamos indicar al compilador que genere información de depuración con el parámetro: “-g”.

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

USO DE GDB

# EJECUCIÓN

Para depurar gdb pasándole como parámetro la aplicación

A continuación podemos ejecutar el comando run para iniciar la depuración. Si la aplicación necesita argumentos se pasan a través de run.

## USO DE GDB

# EJECUCIÓN

```
bash$ gcc -g matrix.c -o matrix
```

```
bash$ gdb matrix
```

```
GNU gdb (Ubuntu 8.3-0ubuntu1) 8.3
```

```
(gdb) run arg1 arg2 arg3
```

USO DE GDB

# EJECUCIÓN CON INTERFAZ DE USUARIO

```
bash$ gcc -g matrix.c -o matrix  
bash$ gdb --tui matrix
```

## USO DE GDB

# BREAKPOINT

Permiten interrumpir la ejecución del programa.

break [N]	b	Interrumpe la línea actual o la línea N
break [fichero]:funcion	b	Interrumpe la función del archivo actual o el especificado
break (punto) if (condicion)	b	Interrumpe en el punto si la condición es cierta
info breakpoints	i b	Lista los puntos de ruptura
delete breakpoints N	d	Borra el punto N
enable N	en	Activa el punto N
disable N	dis	Desactiva el punto N
watch var		Interrumpe si la variable cambia de valor

## USO DE GDB

# FLUJO

## Podemos controlar el flujo del programa

run	b	Inicia la aplicación
continue	b	Continúa la aplicación
fin		Ejecuta hasta el final de la función
step	s	Ejecuta hasta cambiar de línea
step N	s	Ejecuta hasta cambiar N líneas
stepi [N]		Ejecuta 1 o N instrucciones
next [N]	n	Como step pero no entra en subrutinas
nexti [N]		Como stepi pero no entra en subrutinas

## USO DE GDB

# IMPRIMIR INFORMACIÓN

## Podemos visualizar los datos

<code>print var</code>	<code>p</code>	Imprime la variable <code>var</code>
<code>print var@N</code>		Imprime un array de <code>N</code> elementos
<code>print {var1,var2, ...}</code>	<code>p</code>	Imprime varias variables
<code>display var</code>		Como <code>print</code> , pero se imprime después de cada interrupción
<code>display {var1,var2,...}</code>		Como <code>display</code> pero con varias variables
<code>undisplay N</code>		Borra el <code>display N</code>

## USO DE GDB

# IMPRIMIR INFORMACIÓN

## Podemos visualizar los datos

set print array on		Imprime un elemento por fila
set print array-index on		Imprime los índices
info locals	i loc	Muestra las variables locales
info args	i arg	Muestra los argumentos de la función actual
info line	i line	Muestra la línea actual
info functions	i fun	Muestra todas las funciones
list [args]	l	Muestra las líneas de código (en la posición actual o la indicada)
frame		Imprime la línea que se está depurando

USO DE GDB

## **TRAZAS**

Podemos ver la traza de ejecución del programa y movernos entre funciones

## USO DE GDB

# TRAZAS

Podemos ver la traza de ejecución del programa y movernos entre funciones.

backtrace	bt	Imprime la traza de ejecución
backatrace full		Imprime la traza y las variables locales
up		Sube un nivel en la traza
down		Baja un nivel en la traza

# DEPURACIÓN CON GDB EN C/C++

SERGIO GÓMEZ