

GDB QUICK REFERENCE

GDB Version 4

Essential Commands

<code>gdb program [core]</code>	debug <i>program</i> [using <i>coredump core</i>]
<code>b [file:]function</code>	set breakpoint at <i>function</i> [in <i>file</i>]
<code>run [arglist]</code>	start your program [with <i>arglist</i>]
<code>bt</code>	backtrace: display program stack
<code>p expr</code>	display the value of an expression
<code>c</code>	continue running your program
<code>n</code>	next line, stepping over function calls
<code>s</code>	next line, stepping into function calls

Starting GDB

gdb	start GDB, with no debugging files
gdb <i>program</i>	begin debugging <i>program</i>
gdb <i>program core</i>	debug coredump <i>core</i> produced by <i>program</i>
gdb --help	describe command line options

Stopping GDB

quit	exit GDB; also q or EOF (eg C-d)
INTERRUPT	(eg C-c) terminate current command, or send to running process

Getting Help

help	list classes of commands
help <i>class</i>	one-line descriptions for commands in <i>class</i>
help <i>command</i>	describe <i>command</i>

Executing your Program

run <i>arglist</i>	start your program with <i>arglist</i>
run	start your program with current argument list
run ... <i><inf>outf</i>	start your program with input, output redirected
kill	kill running program
tty <i>dev</i>	use <i>dev</i> as stdin and stdout for next run
set args <i>arglist</i>	specify <i>arglist</i> for next run
set args	specify empty argument list
show args	display argument list
show env	show all environment variables
show env <i>var</i>	show value of environment variable <i>var</i>
set env <i>var string</i>	set environment variable <i>var</i>
unset env <i>var</i>	remove <i>var</i> from environment

Shell Commands

<code>cd <i>dir</i></code>	change working directory to <i>dir</i>
<code>pwd</code>	Print working directory
<code>make ...</code>	call “make”
<code>shell <i>cmd</i></code>	execute arbitrary shell command string

[] surround optional arguments ... show one or more arguments

Breakpoints and Watchpoints

break [<i>file</i>]: <i>line</i>	set breakpoint at <i>line</i> number [<i>in file</i>]
b [<i>file</i>]: <i>line</i>	eg: break main.c:37
break [<i>file</i>]: <i>func</i>	set breakpoint at <i>func</i> [<i>in file</i>]
break + <i>offset</i>	set break at <i>offset</i> lines from current stop
break - <i>offset</i>	
break * <i>addr</i>	set breakpoint at address <i>addr</i>
break	set breakpoint at next instruction
break ... if <i>expr</i>	break conditionally on nonzero <i>expr</i>
cond <i>n</i> [<i>expr</i>]	new conditional expression on breakpoint <i>n</i> ; make unconditional if no <i>expr</i>
tbreak ...	temporary break; disable when reached
rbreak <i>regex</i>	break on all functions matching <i>regex</i>
watch <i>expr</i>	set a watchpoint for expression <i>expr</i>
catch <i>x</i>	break at C++ handler for exception <i>x</i>
info break	show defined breakpoints
info watch	show defined watchpoints

clear	delete breakpoints at next instruction
clear [<i>file</i>]: <i>fun</i>	delete breakpoints at entry to <i>fun</i> ()
clear [<i>file</i>]: <i>line</i>	delete breakpoints on source line
delete [<i>n</i>]	delete breakpoints [or breakpoint <i>n</i>]

<code>disable [n]</code>	disable breakpoints [or breakpoint <i>n</i>]
<code>enable [n]</code>	enable breakpoints [or breakpoint <i>n</i>]
<code>enable once [n]</code>	enable breakpoints [or breakpoint <i>n</i>]; disable again when reached
<code>enable del [n]</code>	enable breakpoints [or breakpoint <i>n</i>]; delete when reached
<code>ignore n count</code>	ignore breakpoint <i>n</i> , <i>count</i> times

commands <i>n</i>	execute GDB <i>command-list</i> every time
[silent]	breakpoint <i>n</i> is reached. [silent
<i>command-list</i>	suppresses default display]
end	end of <i>command-list</i>

Program Stack

backtrace $[n]$	print trace of all frames in stack; or of n frames—innermost if $n>0$, outermost if $n<0$
bt $[n]$	
frame $[n]$	select frame number n or frame at address n ; if no n , display current frame
up n	select frame n frames up
down n	select frame n frames down
info frame $[addr]$	describe selected frame, or frame at $addr$
info args	arguments of selected frame
info locals	local variables of selected frame
info reg $[rn] \dots$	register values [for regs rn] in selected
info all-reg $[rn]$	frame; all-reg includes floating point
info catch	exception handlers active in selected frame

Execution

```

continue [count]
c [count]

step [count]
s [count]

stepi [count]
si [count]

next [count]
n [count]

nexti [count]
ni [count]

until [location]
finish
return [expr]

signal num
jump line
jump *address
set var=expr

```

Display

```

print [ /f ] [ e
p [ /f ] [ expr
      x
      d
      u
      o
      t
      a
      c
      f
call [ /f ] expr
x [ /Nuf ] expr

N
u

```

Automatic

```
display [/f]
display
undisplay n

disable disp
enable disp
info display
```

Expressions

<i>expr</i>	an expression in C, C++, or Modula-2 (including function calls), or:
<i>addr@len</i>	an array of <i>len</i> elements beginning at <i>addr</i>
<i>file::nm</i>	a variable or function <i>nm</i> defined in <i>file</i>
<i>{ type } addr</i>	read memory at <i>addr</i> as specified <i>type</i>
<i>\$</i>	most recent displayed value
<i>\$n</i>	<i>n</i> th displayed value
<i>\$\$</i>	displayed value previous to <i>\$</i>
<i>\$\$n</i>	<i>n</i> th displayed value back from <i>\$</i>
<i>\$_</i>	last address examined with <i>x</i>
<i>\$_</i>	value at address <i>\$_</i>
<i>\$var</i>	convenience variable; assign any value
<i>show values [n]</i>	show last 10 values [or surrounding <i>\$n</i>]
<i>show conv</i>	display all convenience variables

Symbol Table

<i>info address s</i>	show where symbol <i>s</i> is stored
<i>info func [regex]</i>	show names, types of defined functions (all, or matching <i>regex</i>)
<i>info var [regex]</i>	show names, types of global variables (all, or matching <i>regex</i>)
<i>whatis [expr]</i>	show data type of <i>expr</i> [or <i>\$</i>] without evaluating; p <i>type</i> gives more detail
<i>p</i> <i>type [expr]</i>	
<i>p</i> <i>type type</i>	describe type, struct, union, or enum

GDB Scripts

<i>source script</i>	read, execute GDB commands from file <i>script</i>
<i>define cmd</i> <i>command-list</i>	create new GDB command <i>cmd</i> ; execute script defined by <i>command-list</i>
<i>end</i>	end of <i>command-list</i>
<i>document cmd</i> <i>help-text</i>	create online documentation for new GDB command <i>cmd</i>
<i>end</i>	end of <i>help-text</i>

Signals

<i>handle signal act</i>	specify GDB actions for <i>signal</i> :
<i>print</i>	announce signal
<i>noprint</i>	be silent for signal
<i>stop</i>	halt execution on signal
<i>nostop</i>	do not halt execution
<i>pass</i>	allow your program to handle signal
<i>nopass</i>	do not allow your program to see signal
<i>info signals</i>	show table of signals, GDB action for each

Debugging Targets

<i>target type param</i>	connect to target machine, process, or file
<i>help target</i>	display available targets
<i>attach param</i>	connect to another process
<i>detach</i>	release target from GDB control

Controlling GDB

<i>set param value</i>	set one of GDB's internal parameters
<i>show param</i>	display current setting of parameter
Parameters understood by set and show :	
<i>complaint limit</i>	number of messages on unusual symbols
<i>confirm on/off</i>	enable or disable cautionary queries
<i>editing on/off</i>	control readline command-line editing
<i>height lpp</i>	number of lines before pause in display
<i>language lang</i>	Language for GDB expressions (auto , c or modula-2)
<i>listsize n</i>	number of lines shown by list
<i>prompt str</i>	use <i>str</i> as GDB prompt
<i>radix base</i>	octal, decimal, or hex number representation
<i>verbose on/off</i>	control messages when loading symbols
<i>width cpl</i>	number of characters before line folded
<i>write on/off</i>	Allow or forbid patching binary, core files (when reopened with exec or core)
<i>history ...</i>	groups with the following options:
<i>h ...</i>	
<i>h exp off/on</i>	disable/enable readline history expansion
<i>h file filename</i>	file for recording GDB command history
<i>h size size</i>	number of commands kept in history list
<i>h save off/on</i>	control use of external file for command history
<i>print ...</i>	groups with the following options:
<i>p ...</i>	
<i>p address on/off</i>	print memory addresses in stacks, values
<i>p array off/on</i>	compact or attractive format for arrays
<i>p demangl on/off</i>	source (demangled) or internal form for C++ symbols
<i>p asm-dem on/off</i>	demangle C++ symbols in machine-instruction output
<i>p elements limit</i>	number of array elements to display
<i>p object on/off</i>	print C++ derived types for objects
<i>p pretty off/on</i>	struct display: compact or indented
<i>p union on/off</i>	display of union members
<i>p vtbl off/on</i>	display of C++ virtual function tables
<i>show commands</i>	show last 10 commands
<i>show commands n</i>	show 10 commands around number <i>n</i>
<i>show commands +</i>	show next 10 commands

Working Files

<i>file [file]</i>	use <i>file</i> for both symbols and executable; with no arg, discard both
<i>core [file]</i>	read <i>file</i> as coredump; or discard
<i>exec [file]</i>	use <i>file</i> as executable only; or discard
<i>symbol [file]</i>	use symbol table from <i>file</i> ; or discard
<i>load file</i>	dynamically link <i>file</i> and add its symbols
<i>add-sym file addr</i>	read additional symbols from <i>file</i> , dynamically loaded at <i>addr</i>
<i>info files</i>	display working files and targets in use
<i>path dirs</i>	add <i>dirs</i> to front of path searched for executable and symbol files
<i>show path</i>	display executable and symbol file path
<i>info share</i>	list names of shared libraries currently loaded

Source Fil

<i>dir names</i>	
<i>dir</i>	
<i>show dir</i>	
<i>list</i>	
<i>list -</i>	
<i>list lines</i>	
<i>[file:]num</i>	
<i>[file:]function</i>	
<i>+off</i>	
<i>-off</i>	
<i>*address</i>	
<i>list f,l</i>	
<i>info line num</i>	
<i>info source</i>	
<i>info sources</i>	
<i>forw regex</i>	
<i>rev regex</i>	

GDB und

<i>M-x gdb</i>
<i>C-h m</i>
<i>M-s</i>
<i>M-n</i>
<i>M-i</i>
<i>C-c C-f</i>
<i>M-c</i>
<i>M-u</i>
<i>M-d</i>
<i>C-x &</i>
<i>C-x SPC</i>

GDB Lice

<i>show copying</i>
<i>show warrant</i>

Copyright (

The author a

This card may
General Public
Please contr

GDB itself is f
it under the te
absolutely no v