

GDB Cheatsheet

Command	Description
help <i>[command]</i>	Gives a list of commands or a brief summary of <i>command</i> .
run <i>program</i>	Run <i>program</i> from the beginning until interrupted.
start <i>program</i>	Like run, but halts the program at the beginning of main.
continue	Continue execution until interrupted, such as by a breakpoint or segfault.
next	Execute a line of code.
step	Like next, but enters into function calls.
until <i>function</i> until <i>[file:]line</i>	Like continue, but stops at location.
up <i>[n]</i> down <i>[n]</i>	Move up and down the function call stack.
list list <i>function</i> list <i>[file:]line</i>	Prints a portion of the source code. Either continues off the previous print, or prints the current position or the argument.
bt/backtrace	Prints the call stack at the execution point.
print <i>expression</i>	Print the value of <i>expression</i> .
watch <i>expression</i>	Break whenever the value of <i>expression</i> changes.
info locals	Prints the local variables with their values.
info args	Prints the arguments to the current function.
edit	Edits a file at the current location.
make	Runs make in the current directory.
! <i>command</i>	Run <i>command</i> as a shell command.
break <i>function</i> break <i>[file:]line</i>	Sets a breakpoint at the position.
disable <i>id*</i>	Disable, but not delete, breakpoint <i>id</i> .
ignore <i>id* count</i>	Ignore breakpoint <i>id</i> until it's been hit <i>count</i> times.
condition <i>id* expr</i> break <i>fn if expr</i>	Modifies breakpoint <i>id</i> or creates a new breakpoint at <i>fn</i> which stops only if <i>expr</i> is true.
Ctrl-z	Halt program and open prompt.
Ctrl-L	Clear/refresh screen.
Ctrl-x a	Open source code view pane.
reverse-continue reverse-step reverse-next	Same as their other counterparts but runs backwards. Make sure to use "target record-full" to start recording.

* Each breakpoint gets an id when created.

GDB Cheatsheet

Command	Example	Abbreviated
help <i>[command]</i>	help help print	h h print
run <i>program</i>	run a.out arg1 arg2 run a.out < in.txt	r a.out arg1 arg2 r a.out < in.txt
start <i>program</i>	start a.out arg1 arg2	start a.out arg1 arg2
continue	continue	c
next	next	n
step	step	s
up <i>[n]</i> down <i>[n]</i>	up down 3	up do 3
until <i>function</i> until <i>[file:]line</i>	until addNode until tree.c:40	u addNode u tree.c:40
list list <i>function</i> list <i>[file:]line</i>	list list main list tree.c:40	l l main l tree.c:40
bt /backtrace	backtrace	bt
print <i>expression</i>	print x[y+3]	p x[y+3]
watch <i>expression</i>	watch *ptr	wa *ptr
info locals	info locals	inf locals
info args	info args	inf args
edit	edit	ed
make	make	mak
! <i>command</i>	! mv tree.c src/	! echo a b c > out.txt
break <i>function</i> break <i>[file:]line</i>	break main break tree.c:40	b main b tree.c:40
disable <i>id</i>	disable 2	dis 2
ignore <i>id count</i>	ignore 2 100	ig 2 100
condition <i>id expr</i> break <i>fn if expr</i>	break foo if !bar condition 7 i == 100	b foo if !bar cond 7 i == 100
reverse-continue reverse-step reverse-next	target record-full run reverse-step	tar record-f r rs