GDB QUICK REFERENCE GDB Version 4

Essential Commands

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

Starting GDB

gdb start gdb, with no debugging files
gdb program begin debugging program
gdb program core debug core dump core produced by program
gdb --help describe command line options

Stopping GDB

quit exit GDB; also q or EOF (eg C-d)

Getting Help

help list classes of commands
help class one-line descriptions for commands in class
help command describe command

Executing your Program

run arglist start your program with arglist
run ...<inf>outf start your program with input, output redirected
run ... start your program with current argument list
kill kill running program

tty dev use dev as stdin and stdout for next run
set args arglist specify arglist for next run
set args specify empty argument list
display arglist display argument list
display env show all environment variables
display env show environment variable value
set env var set environment variable var
unset env var remove var from environment

Shell Commands

cd dir change working directory to dir
ps print working directory
make ... call `make'
sHELL cmd execute arbitrary shell command string

Breakpoints and Watchpoints

break [file:]line set breakpoint at line number [in file]
bm [file:]line eg: break main.c:37
break [file:]func set breakpoint at func [in file]
break +OFFSET set break at offset lines from current stop
break -OFFSET break conditionally on nonzero expr
break *addr set breakpoint at address addr
break ... conditional break
break cond1 if expr new conditional expression on breakpoint n, make unconditional if no expr
temporary break; disable when reached
rbreak regex break on all functions matching regex
watch expr set a watchpoint for expression expr
catch ... break on C++ handler for exception
info break show defined breakpoints
info watch show defined watchpoints
disable n disable breakpoints [or breakpoint n]
able n enable breakpoints [or breakpoint n]
able once n enable breakpoints [or breakpoint n]; disable again when reached
eable del n enable breakpoints [or breakpoint n]; delete when reached
ignore n count ignore breakpoint n, count times
commands n count execute GDB command-list every time breakpoint n is reached
command-list suppresses default display

Program Stack

backtrace [n] print trace of all frames in stack; or of n frames—innermost if n>0, outermost if n<0
frame [n] select frame number n or frame at address n; if no frame, 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 frame arguments of selected frame
info frame locals variables of selected frame
info frame register values [for reg n] in selected frame
info frame all-regs frame; all-reg includes floating point
info catch exception handlers active in selected frame

Execution Control

continue [count] continue running; if count specified, ignore this breakpoint next count times
s [count] step until another line reached; repeat count times if specified
step [count] step by machine instructions rather than source lines
next [count] execute next line, including any function calls
nexti [count] next machine instruction rather than source line
until [location] run until next instruction (or location)
finish return expr pop selected stack frame without executing [setting return value]
signal num resume execution with signal s (none if 0)
jump line resume execution at specified line number
jump address evaluate expr without displaying it; use for altering program variables

Display

print [/] [expr] show value of expr [or last value $] according to format :
x hexadecimal
d signed decimal
u unsigned decimal
b binary
a address, absolute and relative
c character
f floating point
call [/] [expr] like print but does not display void
x [NUM] [expr] examine memory at address expr; optional format spec follows shash

N count of how many units to display
w unit size, one of
b individual bytes
h halfwords (two bytes)
w words (four bytes)
g giant words (eight bytes)

/i printing format. Any print format, or
/s non-terminated string
/i machine instructions

disassem [addr] display memory as machine instructions

Automatic Display

display [/] [expr] show value of expr each time program stops according to format
/display display all enabled expressions on list
/display n remove number(n) from list of automatically displayed expressions

/display n enable display for expression(s) number n
/display n numbered list of display expressions


Expressions

expr
an expression in C, C++, or Modula-2
(including function calls), or:
add@len
an array of len elements beginning at addr
file : nm
a variable or function nm defined in file
{ type } addr
read memory at addr as specified type
$n
n th displayed value
$$
displayed value previous to $n
$-
value at address $-
$varg
convenience variable; assign any value
show values [ n ]
show last 10 values or surrounding $n$
show conv
display all convenience variables

Symbol Table

info address s
show where symbol is stored
info func [ regx ]
show names, types of defined functions
(all, or matching regx)
info var [ regx ]
show names, types of global variables (all,
or matching regx)
what is [ expr ]
show data type of expr or $ without
ewaluting; ptype gives more detail
ptype [ expr ]
describe type, struct, union, or enum

GDB Scripts

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

Signals

handle signal act
specify GDB actions for signal
print
announce signal
no print
be silent for signal
stop
halt execution on signal
nostop
do not halt execution
pass
allow your program to see signal
no pass
do not allow your program to see signal
info signals
show table of signals, GDB action for each

Debugging Targets

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

Controlling GDB

set param value
set one of GDB’s internal parameters
show param
display current setting of parameter
Parameters understood by set and show:
complaint
kml
number of messages on unusual symbols
confirm on/off
enable or disable cautionary queries
editing on/off
control readline command-line editing
height
number of lines before pause in display
language lang
Language for GDB expressions (auto, c or
modula2)
listsize n
number of lines shown by list
prompt str
use str as GDB prompt
radix base
octal, decimal, or hex number
representation
verbose on/off
control messages when loading symbols
width cpl
number of characters before line folded
write on/off
Allow or forbid patching binary, core files
(when reopened with exec or core)
history ...
groups with the following options:
h ...
h exp on/off
disable/enable readline history expansion
h file filename
file for recording GDB command-history
n size size
number of commands kept in history list
h save on/off
control use of external file for command
history,
print ...
groups with the following options:
p...
p address on/off
print memory addresses in stacks, values
p array on/off
compact or attractive format for arrays
p demang on/off
source (demangled) or internal form for
C++ symbols
p asm-dem on/off
demangle C++ symbols in machine-
instruction output
p elements limit
number of array elements to display
p object on/off
print C++ derived types for objects
p pretty on/off
structure display: compact or indented
p union on/off
display union members
p vtbl on/off
display of C++ virtual function tables
show commands
show last 10 commands
show commands n
show commands around number n
show commands +
show next 10 commands

Working Files

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

Source Files

dir names
add directory names to front of source
path
dir
clear source path
dir
show current source path
list
show next ten lines of source
list -
show previous ten lines
list lines
display source surrounding lines, specified
as
file: num
line number in named file
file: function
beginning of function in named file
off
off lines after last printed
off
off lines previous to last printed
address
line containing address
list . in from line to line
info line num
show starting, ending addresses of
compiled code for source line num
info source
show name of current source file
info sources
list all source files in use
forw regex
search following source lines for regex
rew regex
search preceding source lines for regex

GDB under GNU Emacs

K-x gdb
run GDB under Emacs
C-h m
describe GDB mode
R-n
step one line (step)
R-n
next line (next)
C-c C-f
finish current stack frame (finish)
C-c C-c
continue (cont)
C-u
up arg frames (up)
C-d
down arg frames (down)
C-x &
copy number from point, insert at end
(C in source file) set break at point

GDB License

show copying
Display GNU General Public License
show warranty
There is NO WARRANTY for GDB.
Display full no-warranty statement.

Roland Pesch (pesch@cygnus.com)
The author assumes no responsibility for any errors on this card.
This card may be freely distributed under the terms of the GNU
General Public License.
Please contribute to development of this card by annotating it.
GDB itself is free software; you are welcome to distribute copies of
it under the terms of the GNU General Public License. There is
absolutely no warranty for GDB.