**GDB Quick Refrence**

**Startup**

% gdb -help print startup help, show switches

% gdb --args object normal debug with input args

% gdb object core core debug (must specify core file)

% gdb object pid attach to running process

% gdb -tui same as Ctrl-X Ctrl-A

**Help**

help (running / run) list command classes

help info (line) list info commands (running program state)

help show (commands) list show commands (gdb state)

**Breakpoints**

break main set a breakpoint on a function

break 101 set a breakpoint on a line number

break basic.c:101 set breakpoint at file and line (or function)

info breakpoints show breakpoints

delete 1 delete a breakpoint by number

delete delete all breakpoints (prompted)

clear delete breakpoints at current line

clear function delete breakpoints at function

clear line delete breakpoints at line

enable/disable 2 turn a breakpoint on/off, but don't remove it

tbreak function|line set a temporary breakpoint

commands break-no ... end set gdb commands with breakpoint

ignore break-no count ignore bpt N-1 times before activation

condition break-no expression break only if condition is true

condition 2 i == 20 example: break on breakpoint 2 if i equals 20

watch expression set software watchpoint on variable

info watchpoints show current watchpoints

save breakpoints [filename] / source [filename]

**Running the program**

run run the program with current arguments

run args redirection run with args and redirection

set args args... set arguments for run

show args show current arguments to run

cont continue the program

step single step the program; step into functions

step count singlestep \fIcount\fR times

next step but step over functions

next count next \fIcount\fR times

CTRL-C actually SIGINT, stop execution of current program

attach process-id attach to running program

detach detach from running program

finish finish current function's execution

kill kill current executing program

until [n] useful in a loop

**Stack backtrace**

bt print stack backtrace

frame show current execution position

frame *n* show frame *n*

up / down move up / down stack trace (towards/away main)

info locals print automatic variables in frame

info args print function parameters

**Browsing source**

list 101 list 10 lines around line 101

list 1,10 list lines 1 to 10

list main list lines around function

list basic.c:main list from another file basic.c

list - list previous 10 lines

list \*0x22e4 list source at address

cd dir change current directory to \fIdir\fR

pwd print working directory

search regexpr forward current for regular expression

reverse-search regexpr backward search for regular expression

dir dirname add directory to source path

dir reset source path to nothing

show directories show source path

**Miscellaneous**

define command ... end define user command

RETURN repeat last command

shell command args execute shell command

source file load gdb commands from file

**Browsing Data**

print expression print expression, added to value history

print/x expression print in hex

print array[i]@count artificial array - print array range

print $ print last value

print \*$->next print thru list

print $1 print value 1 from value history

print ::gx force scope to be global

print 'basic.c'::gx global scope in named file (>=4.6)

print/x &main print address of function

x/countFormatSize address low-level examine command

x/x &gx print gx in hex

x/4wx &main print 4 longs at start of \fImain\fR in hex

x/gf &gd1 print double

help x show formats for x

info locals print local automatics only

info functions regexp print function names

info variables regexp print global variable names

ptype name print type definition

whatis expression print type of expression

set variable = expression assign value

display expression display expression result at stop

undisplay delete displays

info display show displays

show values print value history (>= gdb 4.0)

info history print value history (gdb 3.5)

**Object File manipulation**

file object load new file for debug (sym+exec)

file discard sym+exec file info

symbol-file object load only symbol table

exec-file object specify object to run (not sym-file)

core-file core post-mortem debugging

**Signal Control**

info signals print signal setup

handle signo actions set debugger actions for signal

handle INT print print message when signal occurs

handle INT noprint don't print message

handle INT stop stop program when signal occurs

handle INT nostop don't stop program

handle INT pass allow program to receive signal

handle INT nopass debugger catches signal; program doesn't

signal signo continue and send signal to program

signal 0 continue and send no signal to program

**Machine-level Debug**

info registers print registers sans floats

info all-registers print all registers

print/x $pc print one register

stepi single step at machine level

si single step at machine level

nexti single step (over functions) at machine level

ni single step (over functions) at machine level

display/i $pc print current instruction in display

x/x &gx print variable gx in hex

info line 22 print addresses for object code for line 22

info line \*0x2c4e print line number of object code at address

x/10i main disassemble first 10 instructions in \fImain\fR

disassemble addr dissassemble code for function around addr

**History Display**

show commands [ \_ / n / + ] print command history (>= gdb 4.0)

show/set history … history

set logging … logging

info editing print command history (gdb 3.5)

ESC-CTRL-J switch to vi edit mode from emacs edit mode

set history expansion on turn on c-shell like history

break class::member set breakpoint on class member. may get menu

list class::member list member in class

ptype class print class members

print \*this print contents of this pointer

rbreak regexpr useful for breakpoint on overloaded member name

.gdbinit gdb init file

tty /dev/pts/2 set gdb output to terminal window /dev/pts/2

quit quit gdb