**GDB Quick Reference**

**Startup**

**.**gdbinit gdb init file

% gdb -help print startup help, show switches

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

% gdb object pid attach to running process

focus src/cmd switch focus to src/cmd window

source file load gdb commands/breakpoints from *file*

set env VAR=value set environment variable

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

**Breakpoints**

break main / n / file:10 set a breakpoint on a *function*/*line*/*file:line*

break class::member set breakpoint on class member

info breakpoints show breakpoints

delete (n) del all breakpoints (breakpoint *n*)

clear (function / line) del break pts at current *line*/*function*/*line*

enable/disable n turn breakpoint *n* on/off

tbreak function / line set a temporary breakpoint at *function*/*line*

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

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

break x:120 if strcmp(y, "hh") == 0 break at line ***120*** only if string ***y*** is “hh”

condition 2 arg == 20 break on breakpoint 2 if ***arg*** equals 20

watch expression set software watchpoint on variable

info watchpoints show current watchpoints

save breakpoints file save breakpoints to *file*

rbreak *regexpr* for break pts on overloaded member name

**Running the program**

attach process-id / detach attach to / detach from running program

kill kill current executing 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 / c continue the program

step / s (n) step (*n*) the program; step into functions

next / n (n) step (*n*) the program; step over functions

finish finish current function's execution

until *line* run until *line*, useful in a loop

ctrl-p / n previous / next command

ctrl-c stop execution of current program

jump *line*/+*offset*/-*offset*/-function *function*/-label *label* jump to location

**Stack back trace**

bt print stack backtrace

frame (*n*) show current execution position (frame *n*)

up / down move up / down stack trace

catch *catch*/*throw* catch a *catching*/*throwing* pt of exception

**Signal control**

info signals print signal setup

handle *signo* actions set debugger actions for signal

handle INT print/noprint, stop/nostop, pass/nopass

signal *signo* / *0* continue and send signal / no signal

**Thread**

info threads inquire about existing threads

thread *thread-id* switch to thread *thread\_id*

thread apply [tid list] [all] *args* apply a command to a list of threads

**Source browsing**

list *n* / *n*, *m* / *(file:)func* list 10 lines around line *n*, from *n* to *m*, etc.

list - list previous 10 lines

list \*0x22e4 list source at *address*

list *globalvar*/*class*/*struct* list *global vars* / *class* / *structure*

list *class::member* list member in class

cd dir change current directory to *dir*

pwd print working directory

search *regexpr* forward search 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

**Data browsing**

print (/x) *expression* print *expression* (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 \*this print contents of this pointer

ptype *type/class/var* print type/class/var members

display *expr/var* | updisplay display at stop / undisplay

info args print function parameters

info locals print local variables only

info functions *regexp* print function names

info variables *regexp* print global variable names

info history print value history (gdb 3.5)

info display show displays

help *x* show formats for *x*

whatis *expression* print type of *expression*

set var *variable = expression* assign value to variable

show values print value history (>= gdb 4.0)

**History display**

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

show/set history …/logging … history / logging

set history expansion on turn on c-shell like history

**Miscellaneous**

define command ... end define user command

shell command args execute shell command

**% GLOG\_logtostderr=true GLOG\_v=8 gdb --args <cmd/obj/bin> <args>**

**(gdb) tty /dev/pts/2**

**(gdb) break main**

**(gdb) run**

**(gdb) source breakpoints.txt**

**(gdb) continue**