| Security Commands Dead D | GDB QUICK REFERENCE GDB Version 4 | Breakpoints a | Breakpoints and Watchpoints | Execution Control | ntrol |
|--|---|--|--|---|--|
| break [file:] func set breakpoint at func [in file] a step [count] break + ajfset set break at ajfset lines from current stop break + ajfset set break at ajfset lines from current stop break + ajfset set breakpoint at address address address address address and incoming a we conditionally on nonzero ergrence on a learner and an increase arguer and a learner and an increase arguer and a learner and an increase arguer and a line cryen and increase arguer and a line cryen and increase arguer and a line crien and a line and a | Essential Commands | break [file:]line b [file:]line | set breakpoint at line number [in file] eg: break main.c:37 | continue [count] c [count] | continue running; if count specified, ignore this breakpoint next count times |
| break -offset breakpoint at address addr step [cound] break -offset set breakpoint at address addr set breakpoint at address addr break -offset set breakpoint at address addr set breakpoint at address address address address address cond in [capp] and be breakpoint at once in a make unconditional if no capt in [cound] no capt capt set a watchold for expression on breakpoint in [cound] no catch event the interiors make unconditional if no capt in [cound] no catch event the interiors make unconditional in the capt in [cound] no catch event the interior cact, total, or treature [capp] no catch event the interior cacter, which may be catch, tinish not address breakpoints at next instruction set variety [cound] into vatch and delete breakpoints at next instruction set variety [capp] and delete breakpoints at next instruction [signal num cache breakpoints of breakpoint in] delete breakpoints [cap breakpoint in] delete breakpoints [cap breakpoint in] and each breakpoints [cap breakpoint in] [capp breakpoint in] and each breakpoints [cap breakpoint in] [capp | debug program [using coredump core] set breakpoint at function [in file] | break [file:]func break +offset | set breakpoint at func [in file] set break at offset lines from current stop | $\mathtt{step} \hspace{0.1cm} [count] \\ \mathtt{s} \hspace{0.1cm} [count]$ | execute until another line reached; repeat count times if specified |
| coneas If easy pressk conditional expression on breakpoint the correct cone of [easy] thread and the conditional expression on breakpoint theread and the conditional expression on breakpoint to care from the care of an available land the care of a subspicious at a care of | start your program [with arylist] backtrace: display program stack display the value of an expression | break -offset break *addr break | | stepi [count] si [count] | step by machine instructions rather than source lines |
| threak roger temporary break disable when reached notificated roger vetch event break as a watchoolnt for expression expr | tinue running your program t line, stepping over function calls t line stemping into function calls | $cond \ n \ [expr]$ | | next [count] n [count] | execute next line, including any function calls |
| are the event service and whethous to expression carporated carporated carporated carporated carbon backs at a watenpoint or expression carporated carbon into watch carbon breakpoints at cart instruction and carbon carb | o mie, svepping moo rancoon caus | tbreak | temporary break; disable when reached break on all functions matching regex | nexti [count] ni [count] | next machine instruction rather than source line |
| info break show defined breakpoints info watch show defined breakpoints at next instruction info watch show defined breakpoints at next instruction is signal num clear [dite:] fun delete breakpoints at next instruction jump *undress in man info watch delete breakpoints at entry to fun() set var=capr clear [fite:] fine delete breakpoints on source line delete breakpoints or breakpoint n] disable breakpoints [or breakpoint n] enable [n] enable breakpoints [or breakpoint n] enable del [n] delete breakpoints [or breakpoint n] disable again when reached enable del [n] delete breakpoints [or breakpoint n] delete breakpoints [or breakpoint n] enable del [n] delete breakpoints [or breakpoint n] delete when reached enable del [n] delete breakpoints [or breakpoint n] delete when reached enable del [n] delete breakpoints [or breakpoint n] delete when reached enable del [n] delete breakpoints [or breakpoint n] delete when reached enable del [n] delete when reached enable del [n] delete when reached enable del [n] delete breakpoints [or breakpoint n] delete when reached enable del [n] enable breakpoints [or frame at address frame [n] end of command-tist non select frame n frames up end of command-tist non select frame n frames down into frame [nddi] delete when reached frame into all-reg [n] frame: all-reg mel memory of frame at address display n display n display n display n display n display n end delete when reached geating point display n | t GDB, with no debugging files | watch event | set a watchpoint for expression expression to break at event, which may be catch, throw ever fork refort load or | until [location] | run until next instruction (or location) |
| clear clear show defined watchpoints show defined watchpoints at next instruction Jump fines clear clear fluc; fund delete breakpoints at entry to fun() set var=crpr clear fluc; fund delete breakpoints at entry to fun() set var=crpr delete breakpoints on source line Display disable [n] delete breakpoints [or breakpoint n] Display enable once [n] delete breakpoints [or breakpoint n] print [n] [arph] enable once [n] enable breakpoints [or breakpoint n]; at a commands n exceute CDB commondies every time commands n exceute CDB commondies every time commands n exceute CDB commondies every time end commondies appreases default display] by n end of commondies at not contract time and commondies every time by [n] frame—innernoss if not, outernoss if not, outernoss if not, outernoss if not frame [nd] frame infants in stack; or of n her [n] end print time of all frames in stack; or of n her [n] end print time of all frames in stack; or of n her [n] end commondies every time and door n select frame number n or frames down doon n select frame n frames down doon n electron a frames of selected frame into a select frame n frames down doon n electron a frames down doon n electron a frames down doon n electron a frames down doon n into locals a select frame n frames down display [n] contraction into a value frame into a select frame n frames down doon n into locals a select frame n frames down doon n into locals a select frame n frames down doon n into locals a select frame n frames down doon n into locals a select frame n frames down doon n doon n select frame n frames down doon n doon n select frame n frames down doon n doon n select frame n frames down doon n doon n select frame n frames down doon n | ug coredump core produced by | of contract of con | unload | return [expr] | pop selected stack frame without |
| clear (ledet breakpoints at next instruction jump *auditess clear [lite:] fun delete breakpoints at entry to fun() set var=expr clear [lite:] fun delete breakpoints on source line delete breakpoints on source line delete breakpoints on source line enable in enable breakpoints [or breakpoint n] print [lite] [sep] and enable breakpoints [or breakpoint n] by [lit] [sep] and enable deal [n] enable breakpoints [or breakpoint n]. Enable deal [n] enable breakpoint n]. Examinated [n] execute GDB communated set every time of factors breakpoint n]. Enable deal [n] enable breakpoint n]. Enable breakpoint n]. Enable deal [n] enable breakpoint n]. Enable breakpoint n]. Enable deal [n] enable breakpoint n]. | cribe command line options | info watch | snow defined breakpoints show defined watchpoints | signal num | executing [secting return value] resume execution with signal s (none if 0) resume execution at enecified line number |
| clear [his] has delete breakpoints at early to jun() clear [his] line delete breakpoints on source line datase [n] disable breakpoints [or breakpoint n] enable [n] enable breakpoints [or breakpoint n] enable del [n] enable breakpoint s [or breakpoint n] enable del [n] enable breakpoint s [or breakpoint n]; disable egaint when reached ignore n count increated preakpoint n]; commands n execute GDB commond-list every time [silear] breakpoint n is reached. [sileart a [N] breakpoint n is reached. [sileart a [N] breakpoint n is reached. [sileart a [N] command-list program Stack trame [n] program Stack trame [n] program select frame number n or frame at address program select frame number n or frame at address program [nd arguments of select frame or frames of select frame number of select frame or frames of select frame number of select frame or frames of select frame number of select frame or frames of select frame number of select frame number of select frame number of select frame or frames of select frame number of select frame or frames of select frame number of select frame or frames of select frame number | | clear | delete breakpoints at next instruction | jump *address | or address |
| delete [n] delete breakpoints [or breakpoint n] Display disable [n] disable breakpoints [or breakpoint n] p [t] [exp] enable once [n] enable breakpoints [or breakpoint n] p [t] [exp] enable once [n] enable breakpoints [or breakpoint n]; disable again when reached ignore n count ignore breakpoint n, count times [silent] breakpoint n is reached. [silent a call [t]] expr commands n execute CDB commend-list every time [silent] breakpoint n is reached. [silent a call [t]] expr end Drogram Stack be [n] command-list need all frames in stack; or of n by [n] n command-list no frame of all frames in stack; or of n by [n] select frame number n or frame at address frame [n] n color all frames of select frame n frames up a select frame n frames of selected frame info and select frame n frames of selected frame info all-reg [n] frame; all-reg includes floating point display n disable disp n ounts ounts disable breakpoint n] frame; all-reg includes floating point display n disable disp n ounts | GDB; also q or EOF (eg C-d) C-c) terminate current command, or | clear [file:] line | delete breakpoints at entry to fun() delete breakpoints on source line | ser var-enpr | evaluate expr. without displaying it; use for altering program variables |
| disable [n] disable breakpoints [or breakpoint n] print [ll] [expr] enable once [n] enable breakpoints [or breakpoint n] p [ll] [expr] enable del [n] enable breakpoints [or breakpoint n]; disable again when reached ignore n count ignore breakpoints [or breakpoint n]; commands n execute CDB command-list every time [silent] breakpoint n is reached. [silent a breakpoint n] x [ll/l] expr command [sit suppresses driant display] and Program Stack be cit of or one and frames in stack; or of n by [n] recommand-list in trace of all frames in stack; or of n by [n] recommand-list in trace of all frames in stack; or of n by [n] select frame number n or frame at address trame [n] select frame n frames up select frame n frames up select frame n frames of selected frame into 10cals into args line all-reg [n] frame; all-reg includes floating point display n disable disp n onness disable disp n | nd to running process | $\mathtt{delete}\ [n]$ | delete breakpoints [or breakpoint n] | Display | |
| enable [n] enable breakpoints [or breakpoint n] and enable once [n] enable breakpoints [or breakpoint n]; enable del [n] enable breakpoints [or breakpoint n]; delete when reached is now cannot delete when reached is now commands a supersessed and now command lets appresses default display end command lets suppresses default display and Program Stack by [n] frames—innermost if n>0, outermost if now elsect frame number n or frame at address frame [n] select frame n frames up select frame n frames down down info organ [nd of gentle display number of select frame n frames of select frame n frames down down info organ [nd organic of select frame n frames of select frame of frames of select frame of frames of select frame and not | | disable [n] | disable breakpoints or breakpoint n | print $[/f]$ [expr] | show value of expr [or last value \$] |
| enable once [n] enable breakpoints [or breakpoint n]; enable del [n] enable breakpoints [or breakpoint n]; elable again when reached ignore n count commands n commands n breakpoint n, count times command-ist end of command-ist breakpoint n is reached, [silant x[Nud] expr end Program Stack backtrace [n] print trace of all frames in stack; or of n by [n] frames—innermost if no, outermost if trame [n] select frame number n or frame at address time aselect frame number n or frame at address inf on n, select frame number or of name at address inf on n, select frame number of select frame of select frame or frames of select frame or frames of select frame in frames of select frame info locals info all-rog [m] frame; all-rag includes floating point display n disable disp n enable disp n disable disp n disable disp n disable disp n enable disp n into display into display into display into display disable disp n enable disp n | classes of commands | enable $[n]$ | enable breakpoints [or breakpoint n] | p [/f] [expr] | according to format f: |
| enable del [n] enable breakpoints [or breakpoint n]; delter when reached ignore a recount ignore breakpoint n; count times commands n execute CDB command-list suppresses draint display] and Program Stack becktrace [n] print trace of all frames in stack; or of n trame [n] print trace of all frames in stack; or of n trame [n] eselect frame number n or frame at address frame [n] eselect frame n frames up select frame n frames up select frame n frames up a select frame n frames of select frame n frames of selected frame info args into locals info locals info all-reg [n] frames; all-reg includes floating point display n display n display n info all-reg [n] frames; all-reg includes floating point display n info all-reg [n] frames; all-reg includes floating point display n info d | the descriptions for commands in ss ss ribe command | enable once $[n]$ | enable breakpoints [or breakpoint n]; disable again when reached | × T : | hexadecimal signed decimal imsigned decimal |
| ignore n count ignore breakpoint n, count times c commands n commands n breakpoint n, count times c commands n breakpoint n, servedue (GDB commund-list every time c commands n breakpoint n is reached, [silent connum deliated commund-list command-list end of command-list end of command-list end of command-list not disable disp n display n disable disp n | ogram | enable del $\left[n\right]$ | enable breakpoints [or breakpoint n]; delete when reached | 4 0 1 | octal binary |
| commands n execute GDB command-list every time file approximated-list supersessed default display x [Ntrd] capr | your program with arglist | ignore n count | ignore breakpoint n, count times | ଟେଧ | address, absolute and relative character |
| command-list suppresses default display and Program Stack backtrace [n] print trace of all frames in stack; or of n by [n] frames—innermost if n00, outenmost if trame [n] select frame number n or frame at address u np n select frame number n or frame at address inf in on, adjointy current frame about n select frame n frames of an info arga info arga info arga info arga info arga info all-rog [rn] frame; all-rag includes floating point info display n disable disp n info display info display info display disable disp n | your program with current argument your program with input, output | commands n [silent] | execute GDB command-list every time breakpoint n is reached. [silent | f call $[/f]$ $expr$ | floating point like print but does not display void |
| Program Stack backtrace [n] print trace of all frames in stuck: or of n by [n] round-innermost if n>0, outermost if frame [n] select frame number n or frame at address frame [n] in n if non adjently current frame up n select frame n frames up select frame n frames up select frame n frames own info args income [add] describe selected frame, or frame at addr info args income [add] describe selected frame, or frame at addr info args income [add] describe selected frame in a selected frame info args income at a selected frame info args in a guments of selected frame info argin in a selected info teag [n] frame; all-reg includes floating point display n disable disp n disable disp n info display in the display in t | irected | | suppresses default display] end of command-list | x [/Nuf] $expr$ | examine memory at address expr; optional format spec follows slash |
| becktrace [n] print trace of all frames in study; or of n frames—linearmost if no.6, outermost if no.6 frames—linearmost if no.6 frames—linearmost if no.6 frame number n or frame at address frame up n select frame n frames up down info frame [ndin] describe selected frame or frames of select frame or frames of select frame or frames of select frame and described described described info totals arguments of selected frame or frames of selected frame info reag [nd] described of selected frame info reag [nn] . register values for reggs mj in selected display [n] frame; all-reg [nn] frame; all-reg includes floating point display n disable disp n each described info display in the display not disable disp not display not | dev as stdin and stdout for next run | Program Stac | ب | ~ π | count of how many units to display unit size; one of |
| frame [n] select frame number nor frame at address n if no n, display current frame up n select frame n frames a up down n info frame [addr] info args info args info locals info locals info all-reg [m] info all-reg [m] info all-reg [m] info all-reg includes floating point display info all-reg includes floating point display n display n info display | fly arglist for next run fly empty argument list av argument list | backtrace [n] | print trace of all frames in stack; or of n frames—innermost if n>0, outermost if | | b individual bytes h halfwords (two bytes) w words (four bytes) |
| up n ref in on a righwy current frame of the reason and the reason | v all environment variables | frame $[n]$ | $n \le 0$ select frame number n or frame at address | 4-7 | g giant words (eight bytes) printing format. Any print format, or |
| down select frame n frames down info args info argination of selected frame info reg [rn] frame; all-reg includes floating point display n disable disp n disable disp n disable disp n info display info display info display info able disp n disable disp n display info display | r value of environment variable var environment variable var | u dn | n; if no n, display current frame select frame n frames up | | s null-terminated string i machine instructions |
| into args arguments of selected frame Automatic Disjuint locals local variables of selected frame display [//] exprint allocal [m] register values for regs m] in selected display [//] exprint allocal [m] frame; allocal floating point display windisplay n display | ove var from environment | down n info frame [addr] | select frame n frames down describe selected frame, or frame at $addr$ | ${\tt disassem} \; [addr]$ | display memory as machine instructions |
| undisplay n disable disp n enable disp n enable disp n into display n into display n | ge working directory to dir "make" ute arbitrary shell command string | info args info locals info reg $[m]$ info all-reg $[rm]$ | 87 5 | Automatic Di $_{ m display}[f]$ $_{expr}$ $_{ m display}$ | splay show value of expreach time program stops [according to format f] display all enabled expressions on list |
| enable disp n info display | ents show one or more arguments | | | undisplay n disable disp n | remove number(s) n from list of automatically displayed expressions disable display for expression(s) number n |
| | , , , , , , , , , , , , , , , , , , , | | | enable disp n | enable display for expression(s) number n numbered list of display expressions |

| Expressions | | Controlling GDB | DB | Source Files | |
|--------------------------|--|---------------------|--|---------------------------------|--|
| expr | an expression in C, C++, or Modula-2 | set param value | set one of GDB's internal parameters | dir names | add directory names to front of source |
| | (including function calls), or: | show param | display current setting of parameter | | path |
| addr@len | an array of ten elements beginning at | Parameters understo | Parameters understood by set and show: | dir | clear source path |
| Glo. r. some | addr o maight on function and defined in 610 | complaint limit | | show dir | show current source path |
| Jacc - 1910 | a variable of temporal rate defined in Just | confirm on/off | enable or disable cautionary queries | lio+ | about next ton lines of someon |
| $\{type\}addr$ | read memory at addr as specified type | fo/uo Suitipe | control readline command-line editing | list - | show next ten lines of source |
| 69 · | most recent displayed value | height lpp | number of lines before pause in display | list lines | display source surrounding lines, specified |
| £ 69 € | nth displayed value | language lang | Language for GDB expressions (auto, c or | | 388 |
| 99 6 | displayed value previous to \$ | | modula-2) | file: nam | line number [in named file] |
| 9 0 | hat address assume back from a | Tistsize n | number of lines shown by List | ; | |
| , e | value at address \$ | prompt str | use str as GDB prompt | frie: Jranction | beginning of function [in named file] |
| es es | convenience variable: assign any value | agno vinor | representation | ffo+ | off lines after last printed |
| | | Ho/ no osoquon | control monorous when leading combols | - on | off lines previous to last printed |
| show walnes | show last 10 values for surrounding \$n | width cal | control messages when loading symbols number of characters before line folded | 1:o+ f / | from line for line 1 |
| show conv | disnlay all convenience variables | write on/off | Allow or forbid patching binary, core files | info line num | show starting, ending addresses of |
| | O CONTRACTOR OF THE PARTY OF TH | | (when reopened with exec or core) | | compiled code for source line num |
| Symbol Table | | history | groups with the following options: | info source | show name of current source file |
| of months | | ъ | | info sources | list all source files in use |
| info address s | show where symbol s is stored | p exp off/on | disable/enable readline history expansion | forw regex | search following source lines for regex |
| info func [reger] | show names, types of defined functions | h file filename | file for recording GDB command history | rev regex | search preceding source lines for regex |
| | (all, or matching regex) | h size size | number of commands kept in nistory list | | |
| info var $[regex]$ | show names, types of global variables (all, | m save ollon | control use of external life for command history | GDB under GNU Emacs | NU Emacs |
| | or matching regex) | 41711 | | M-x gdb | run GDB under Emacs |
| whatis [expr] | show data type of expr [or \$] without | pr mr | groups with the following options: | C-b m | describe GDB mode |
| ptype expr | evaluating; ptype gives more detail | p oddwood a | contact and an artist of the contact | M-s | step one line (step) |
| pt.vpe tame | describe type struct union or enum | Ma/ma essamme d | p duates on of plun memory authores in stacks, values | M-n | next line (next) |
| 10. 10. 4 | | parray off/on | compact or attractive format for arrays | M-i | step one instruction (stepi) |
| | | b demangl on/off | p demangl on/off source (demangled) or internal form for | C-c C-f | finish current stack frame (finish) |
| GDB Scripts | | | C++ symbols | M-c | continue (cont) |
| source script | read, execute GDB commands from file | p asm-dem on/off | p asm-dem on/off demangle C++ symbols in machine- | M-u | up arg frames (up) |
| | script | 2 | instruction output | M-d | down arg frames (down) |
| dofine out | annate now ODD command and amounts | p elements umur | p elements is must number of array elements to display | G-x & | copy number from point, insert at end |
| commond-list | create new GDD command critic, execute | p object on/off | p object on/off print C++ derived types for objects | C-x SPC | (in source file) set break at point |
| and and | and of command-list | p pretty off/on | | | |
| document and | end of communication for now CDB | ffo/uo uoiun d | display of union members | CDD Linear | |
| heln-tent | command cmd | p vtbl off/on | display of C++ virtual function tables | acina ricelise | |
| end way | and of help-text | | | show copying | Display GNU General Public License |
| | | show commands | show last 10 commands | show warranty | There is NO WARRANTY for GDB. |
| Cionala | | show commands n | show 10 commands around number n | | Display full no-warranty statement. |
| onguens beedle in the | day | SHOW COMMUNICS + | show hext to commands | | |
| national sayman act | specify GDB actions for signar, | Working Files | | | |
| noprint | be silent for sienal | Sur Fermion | | | |
| ston | halt execution on signal | file fue | use file for both symbols and executable; | Convright (2)1991 | Convright Class '92 '93 '98 Free Software Foundation Inc |
| nostop | do not halt execution | | with no arg, discard both | on the same | Boland H. Pesch |
| pass | allow your program to handle signal | core [file] | read file as coredump; or discard | The author assume | The author assumes no responsibility for any errors on this card. |
| nopass | do not allow your program to see signal | exec [hle] | use file as executable only; or discard | This cord may be fa | This cood may he feedy distributed under the terms of the CMII |
| info signals | show table of signals, GDB action for each | symbol [file] | use symbol table from file; or discard | General Public License. | se. |
| | | load file | dynamically link file and add its symbols | | |
| Debugging Targets | rgets | add-sym file addr | read additional symbols from file, | Improvements can be | I'lease contribute to development of this card by annotating it. Improvements can be sent to bug-gdb@gnu.org. |
| target type param | target type param connect to target machine, process, or file | | dynamically loaded at addr | | 0 |
| help target | display available targets | info files | display working files and targets in use | GDB itself is free so | GDB itself is free software; you are welcome to distribute copies of |
| detach param | connect to another process release target from GDB control | patn dars | add dars to front of path searched for executable and symbol files | it under the terms of the GNU C | it under the terms of the GNU General Public License. There is shedutely no weaventy for GDR |
| | | show path | display executable and symbol file path | appointery no warran | 10 den: |
| | | info share | list names of shared libraries currently | | |
| | | | loaded | | |