GDB Cheatsheet

- **GDB Initialization**

Command	Description
`gdb ./myprogram`	Debug a specific program
`gdb --args ./myprogram arg1 arg2`	Debug with arguments
`gdb -p <process_id>`	Attach to a running process

- **Running Program**

Command	Description
`run <args>`	Start the program (r for short)
`start`	Start and stop at main()
`continue`	Continue execution (c for short)

- **Breakpoints**

Command	Description
`break main`	Set breakpoint at main() (b for short)
`break file.c:10`	Set breakpoint at line 10 of file.c
`break function_name`	Set breakpoint at a function
`info breakpoints`	List all breakpoints (i b for short)
`delete <breakpoint_num>`	Remove a breakpoint (d for short)

- **Step through code**

Command	Description
`next`	Execute next line, skip function calls (n for short)
`step`	Execute next line, enter function calls (s for short)
`finish`	Run until current function returns
`until <line>`	Run until specified line

- **Execution Control**

Command	Description
`quit`	Exit GDB (q for short)
`kill`	Stop the running program
`Ctrl+C`	Interrupt the running program

- **Examining Data**

Command	Description
`print variable`	Print value of variable (p for short)
`print *pointer`	Print value pointed to by pointer
`print array[5]@10`	Print 10 elements starting at array[5]
`display variable`	Automatically print variable each step
`info locals`	Show local variables
`info registers`	Show CPU registers

- **Examining Code**

Command	Description
`list`	Show source code around current position (l for short)

Command	Description
`list function`	Show source code of function
`list file.c:15`	Show source code around line 15 of file.c
`backtrace`	Show function call stack (bt for short)

- **Writing to Registers**

Command	Description
`set *(char *)($ebp-0x8) = 0x41`	Writing a single byte char('A' in Ascii)
`set *(short *)($ebp-0x8) = 0x1c`	Writing a 2-byte short integer
`set *(int *)($ebp-0x8) = 0xdeadbeef`	Writing a 4-byte long integer
`set *(long long *)($ebp-0x8) = 0xdeadbeefcafebabe`	Writes an 8-byte long long
`set {char [5]}($ebp-0x8) = "ABCD"`	Writes the 5-byte string "ABCD" (includes null terminator 0)

- **Useful Configuration**

```
set disassembly-flavor intel
set pagination off
```