GDB Linux Kernel Debugging Cheatsheet

Attaching to the kernel

```
target remote localhost:PORT
```

Attach to GDB kernel stub

```
add-symbol-file /path/to/mod.ko 0x...
```

Load the symbols file at the specified address

```
set substitute-path /old/path /new/path
```

Path substitution rule for finding source files. Note: useful for debug symbols with absolute paths

```
directory /kernel/source/path
```

Similar to above: search the specified directory for source files

```
detach
```

Detach the debugger without killing the VM

Breakpoints/watchpoints

```
b *0x...
```

Set a breakpoint at the specified memory address

```
en|dis [num]
```

Enable|disable a single breakpoint *num*

```
watch|rwatch|awatch *0x...
```

Set a watchpoint that's triggered on *writes*|*reads*|*both reads and writes* to the specified memory location

```
b *0x...  if cond
```

Break at the specified memory address if *cond* is true

```
command [num]
```

Specify commands every time you hit breakpoint number *num*

Stepping

```
si
```

Step one machine instruction

```
ni
```

Similar to above but steps over function calls

```
finish
```

Continue execution until the current function returns

```
return [val]
```

Terminate the exec path and return *val*

Examining the stack

```
bt|where
```

Show the call stack

```
frame [num]
```

Select the stack frame

Information

```
i b/d/r
```

Show breakpoints/displays/registers

```
show directories
```

Show source code directories

```
whatis var
```

Print type of the specified variable

```
i locals
```

Print local variables for the current stack frame

```
ptype struct name
```

Print the struct definition

Examining vars and memory

```
x/nfu 0x...
```

Print memory at the specified address

`n` - number of units to print
`f` - format (similar to printf)
`u` - unit (`g/w/h/b` - 64-bit val/32-bit/16-bit/single byte)

```
x/10i 0x...
```

Disassemble 10 instructions at the specified address

```
p *&array[0]@N
```

Print first *N* elements of the *array*. Move the array index to get specific elements

```
display [var|addr]
```

Similar to *print* but print *var/addr* after each stepping instruction

```
undisplay/en display [num]/dis
display [num]
```

Remove all/enable or disable a single display

Misc

```
layout split
```

Show both source code and machine instructions. Use `Ctrl-x o` to switch active window

```
set disassembly-flavor [att|intel]
```

Set disassembly style to AT&T or Intel

VMware x64 default GDB stub port `8864`
Qemu (-s) default GDB stub port `1234`

Author: Vitaly Nikolenko
http://duasynt.com/pub/gdb.pdf

Searching memory

```
find 0xdeadbeef,+0x1000,’t’,(char)0x65,’s’,’t’
```

Search starting from `0xdeadbeef` to `0xdeadbeef+0x1000` for the sequence of bytes ’test’

```
find /w1 0xdeadbeef,+0x1000,0x74736574
```

Equivalent to the above but searches for a single occurrence of ’test’. Other format attributes are similar to the `x` command: `b` - byte, `h` - half word, etc.