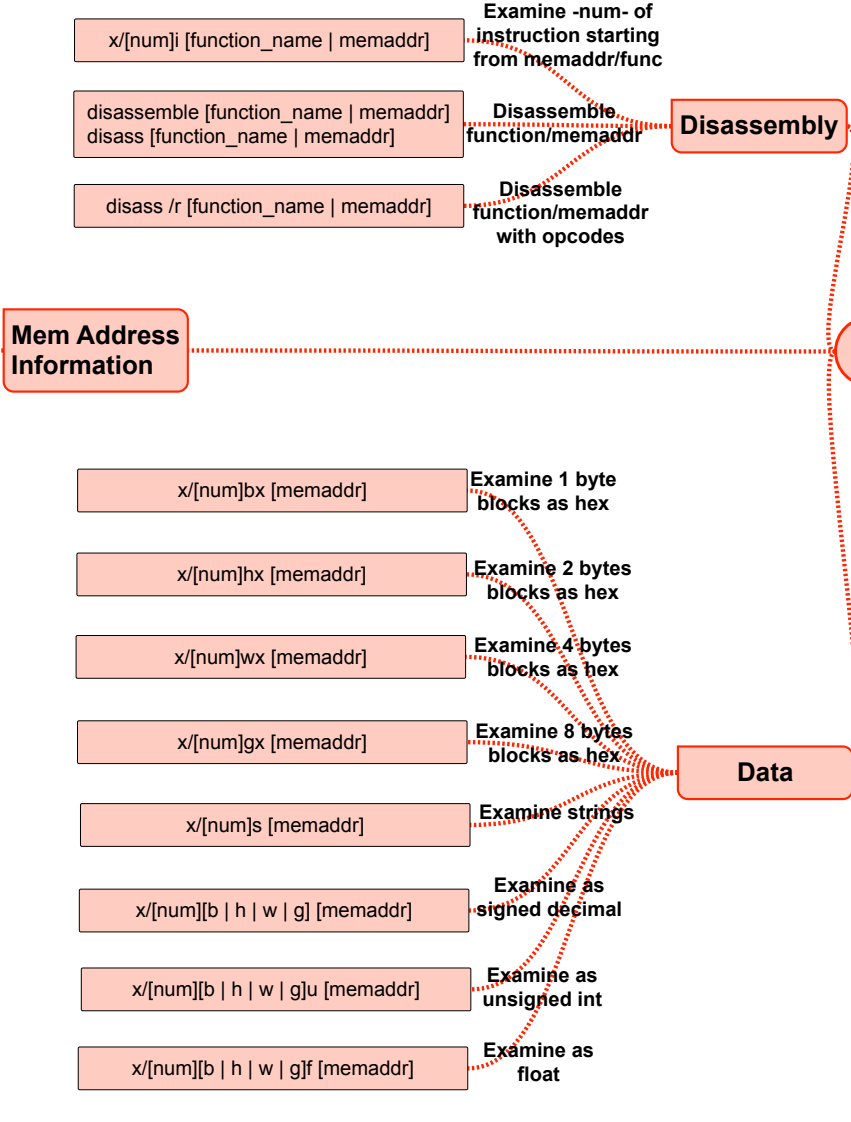
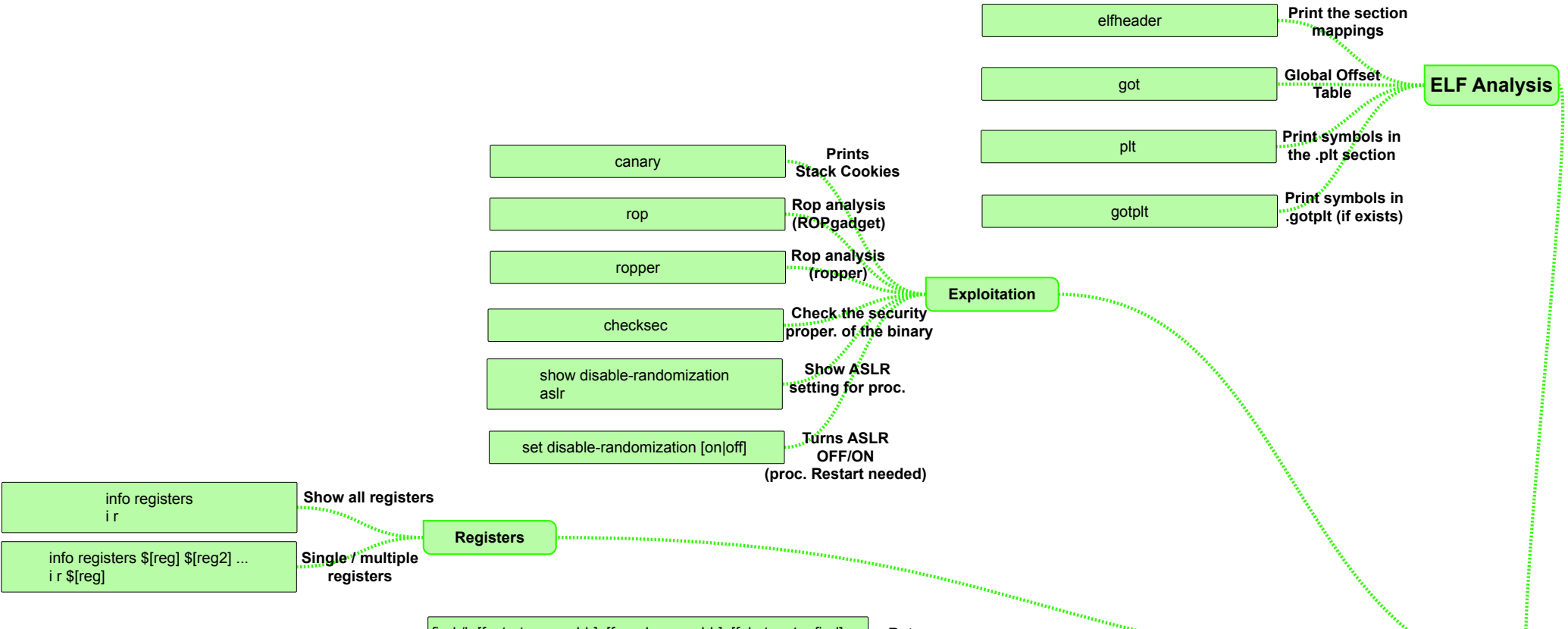


# PwnDBG / gdb v0.1

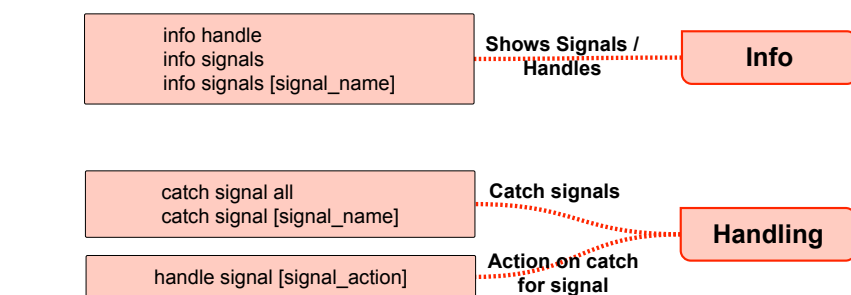
## Examination



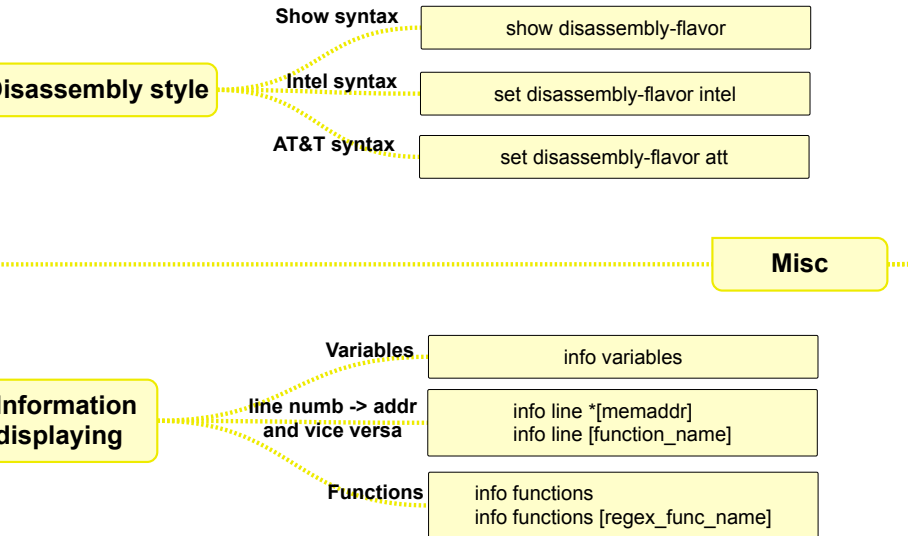
## Analysis



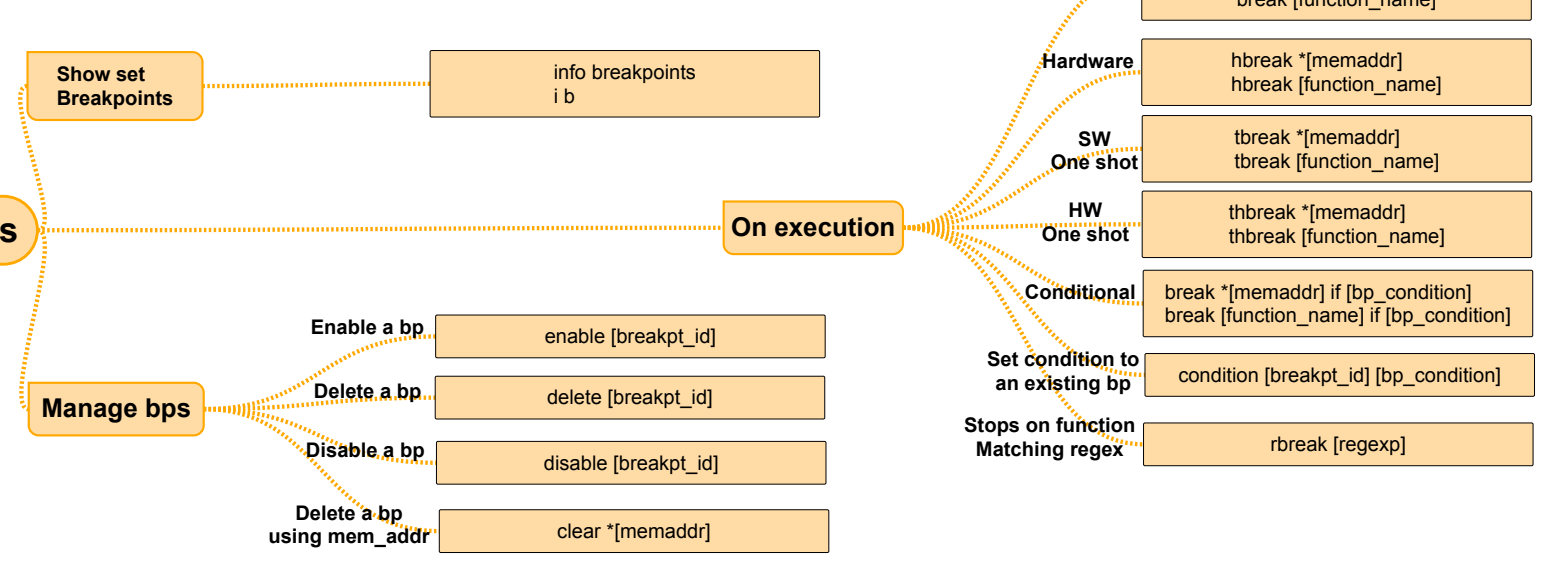
## Signals



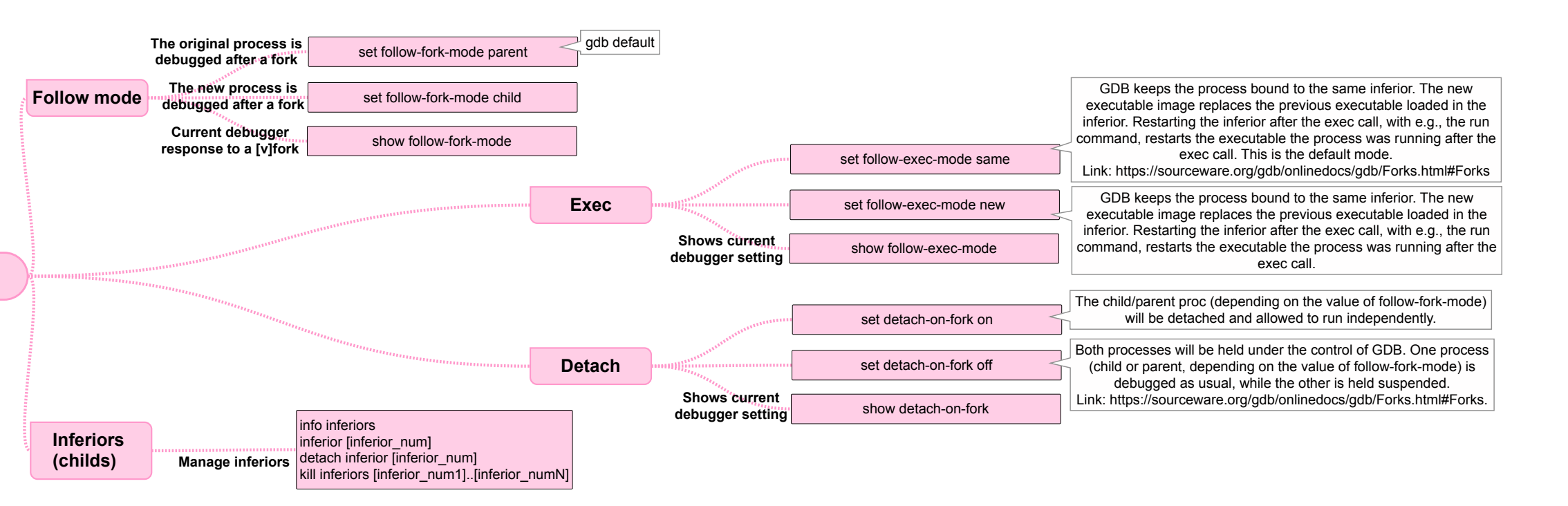
## Generic commands



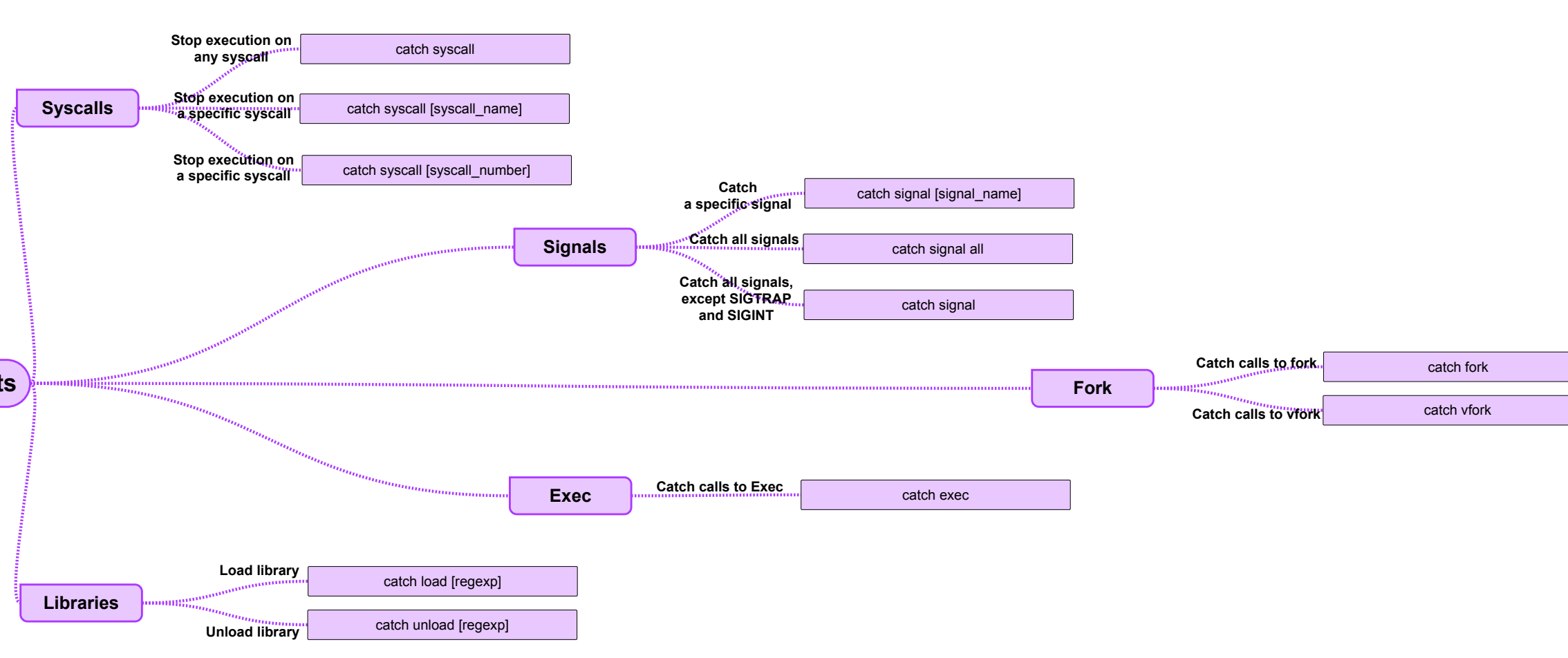
## Breakpoints



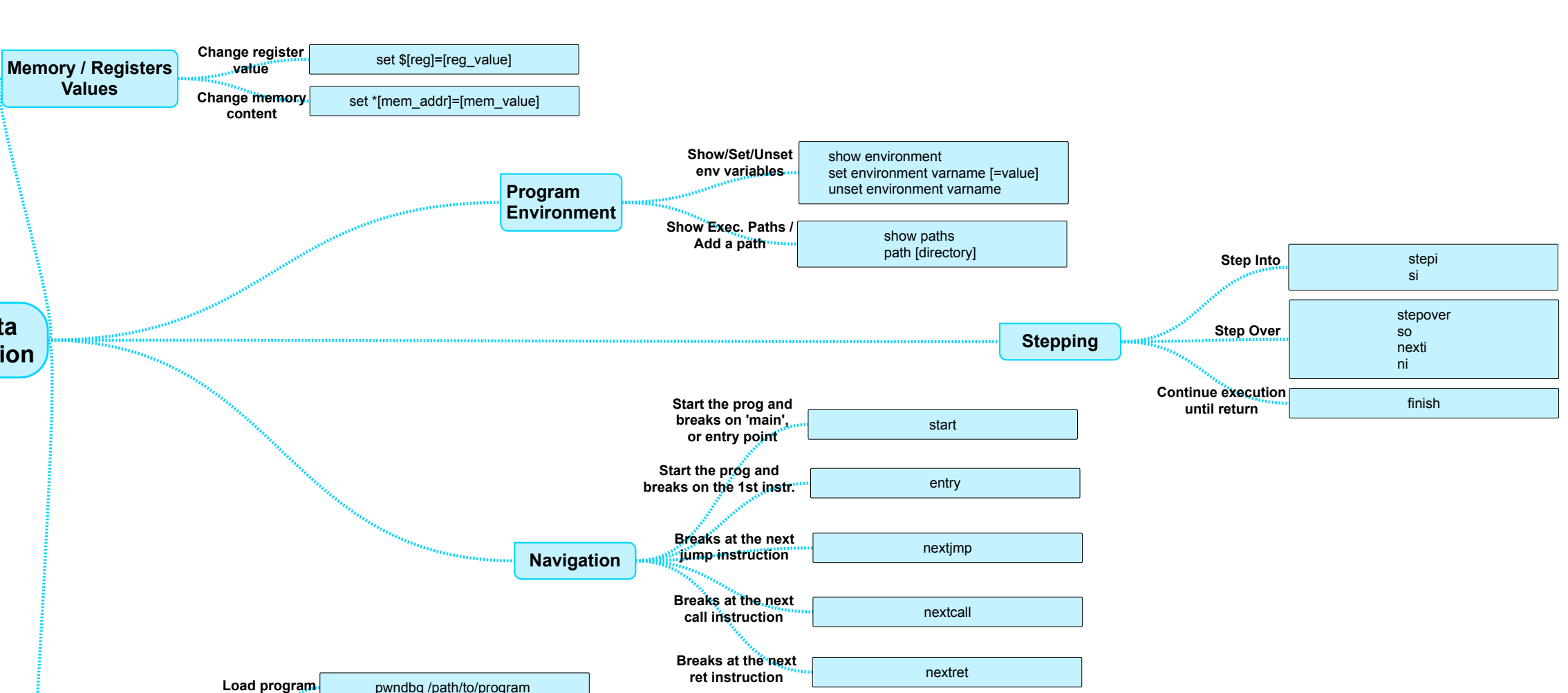
## Fork



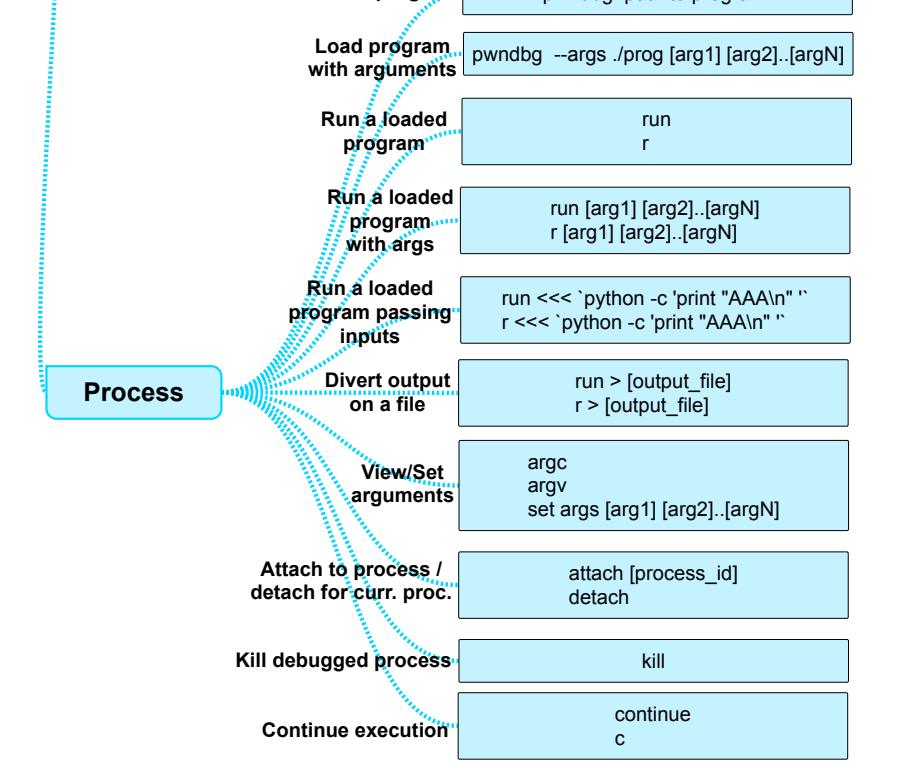
## Catchpoints



## Flow/Data Manipulation



## Process



### Values:

```
[arena_addr] - (hex) memory address of the arena in hex format (i.e. 0x08043000)
[b|h|w|g] - (char in list) 'b' to examine as byte, 'h' for half word (16 bits), 'w' for word (32 bits), 'g' for giant (64 bits)
[breakpt_id] - (integer) breakpoint id (i.e. 1, 2, 40, etc.) that you already set, returned by pwndbg
[bp_condition] - (string) condition to match (i.e. $eax == 0, 0x0804556c != 0, etc.)
[count_s] - (integer) Number of element of the stack to dump
[f_start-memaddr] [f_end-memaddr] - (hex) memory addresses to use as start and end for memory search in hex format (i.e. 0x08043000)
[f_bytes_to_find] - (hex) byte or sequence of bytes to search (comma separated). For example 0x11, 0x22, etc. See Examples
[f_halfword_to_find] - (hex) half word (16bits) or sequence of half words to search (comma separated). For example 0x1122
[f_word_to_find] - (hex) word (32bits) or sequence of words to search (comma separated). For example 0x11223344
[f_giant_to_find] - (hex) giant (64bits) or sequence of words to search (comma separated). For example 0x1122334455667788
[function_name] - (string) Function name (i.e. main, printf, etc.)
[function_name | memaddr] - (strings) Function name or Memory address (i.e. main, 0x08040404)
[level] - (integer) Stack Frame level (0 is the current one)
[memaddress] - (hex) memory address such as 0x08041b5c
[memaddress] - (hex) memory address in the format "0x08041b5c"
[mem_value] - (signed int | hex | char | string) Value for memory (i.e. 0x1337, 1, -1, 0xFFFFFFFF, 'A', "String", etc.)
[num] - (unsigned integer) Number of instructions (i.e. x/10i - 10 instructions), bytes, word, dword, etc.
[num_of_bytes] - (unsigned int) Number of bytes to dump
[offset_s] - (integer) Element offset from $esp
[reg] - (string) Regular Expression, value depends on expected function/library name, etc.
[reg] - (string) Name of a register (i.e. eax, ax, ah, al, ebx, etc.)
[reg_value] - (signed int | hex | char) Value for register (i.e. 0x1337, 1, -1, 0xFFFFFFFF, 'A', etc.)
[signal_action] - (string in list) 'nostop' | 'stop' | 'noprint' | 'pass' | 'ignore' | 'noignore'
[signal_name] - (string) signal name (i.e. SIGSEGV, SIGKILL, etc.)
[syscall_number] - (unsigned integer) number of a specific syscall (i.e. 0 for x86 Read, 1 for x86 Write, etc.)
[syscall_name] - (string) Name of a specific syscall (i.e. 'read', 'write', 'socket', 'fork', etc.)
```

## Examples

### Registers/Memory Analysis (Little Endian x86 ELF)

```
0xffffd4b4: 0x54 0xd5 0xff 0xff 0x5c 0xd5 0xff 0xff
pwndbg> x/4bx 0xffffd4b4
0xffffd4b4: 0x54 0xd5 0xff 0xff
pwndbg> x/bd 0xffffd4b4
0xffffd4b4: 84
pwndbg> x/gx 0xffffd4b4
0xffffd4b4: 0xffffd5cffffd54
pwndbg> info line main
No line number information available for address 0x0804854 <main>
pwndbg> !r $eax
eax 0x0804020 134520864
pwndbg> !r $ax
ax 0x020 -24544
pwndbg> !r $al
al 0x20
pwndbg> set $ax=0x1337
pwndbg> !r $eax
eax 0x041337 134484791
pwndbg> !r $ebx $ecx $esp
ebx 0x0
ecx 0x3356e5c4 861332932
esp 0xffffd4a8 0xffffd4a8
```

### Search Patterns in Memory (Little Endian x86 ELF)

```
0x08048000: 0x54 0xe8 0x69 0x73 0x20 0x69 0x73 0x20
0x08048008: 0x61 0xe6 0x20 0x65 0x78 0x61 0x6d 0x70
0x08048010: 0xc6 0x65 0x00 0x00
0x08048090: "This is an example"
pwndbg> find /b 0x08048000, 0x08048013, 'e','x','a','m'
0x0804800b
1 pattern found.
pwndbg> find /b 0x08048000, 0x08048013, 0xc6,0x65
0x08048010
1 pattern found.
pwndbg> find /w 0x08048000, 0x08048013, 0x0000656c
0x08048010
1 pattern found.
pwndbg> find /h 0x08048000, 0x08048013, 0x656c
0x08048010
1 pattern found.
pwndbg> find /g 0x08048000, 0x08048013, 0x706d61786520e661
0x08048008
1 pattern found.
pwndbg> find 0x08046000, 0x08048013, "example"
Pattern not found.
pwndbg> find 0x08048000, 0x08048013, "example"
0x0804800b
1 pattern found.
```

LINKS  
Pwndbg Docs: <https://browserpwndbg.readthedocs.io/en/docs/>  
GDB Reference: <https://sourceware.org/gdb/current/online/docs/gdb/index.html#Top>

Author: bdev  
Website: <https://reversingforfun.info>

