GDB

Starting

```
gdb <executable>
gdb --args <executable> arg1 arg2 arg3
gdb -x gdbcommands --init-commands=gdb-openocd.cfg
```

Symbols and paths

Add symbols:

```
symbol-file /mnt/nfs/binary-with-symbols.elf
dir /mnt/nfs/src
```

Using

getting info

set print pretty on

- p[rint] <name> - print value
  - p * <name> - print what is pointed by
  - p/x <name> - print in hex format
  - p <name> @<n> print <n> values starting at <name>
  - p <chars> <TAB> - List all variables starting with <chars>
  - p ((sturct dummy*)variable)[0] - Dump structure members
- ptype var
- x /100bx m - eXamine memory located by variable m, print 100 bytes in hex format
- bt (backtrace)
- t (threads)

call stack

- bt (backtrace)
- frame 4 - switch to frame #4
- up
- down

breakpoints

- b[reak] malloc
- b mall <TAB> - List all starting with mall
- clear malloc
- info break
    - i[nfo] b
- dis[able] 1 - disable brakpoint 1
- en[able] 1
- d[elete] 1
- d 1 2
- d - delete all breakpoints
- cond[ition] 1 <expr> - stop at breakpoint 1 only if <expr> is true
- cond 1 - Remove condition from breakpoint 1

stepping

- s [tep] - step to next **source** line
- c [ontinue] <num_to_ignore>
- n [ext] - step (do not enter to subroutines)
- fin [ish] - execute until stack frame returns
- u [ntil] <line number> - execute to line (to avoid loops)

infos

- show debug-file-directory

threads

- info threads - show info about threads
- thread 4 - switch to thread 4
- print mutex
- info line - print current code line
- frame - print current execution position and code

OpenOCD

Disable Cortex M0 interrupts

- mon cortex_m maskisr on

```
set remote hardware-breakpoint-limit 4
set remote hardware-watchpoint-limit 2
define hook-step
mon cortex_m maskisr on
end
define hookpost-step
```

```
mon cortex_m maskisr off
end
```

gdbcommands file

gdbcommands

```
set verbose on
set auto-load safe-path /
set debug auto-load on

set sysroot /home/user/prj/buildroot/output/target

source ../../interactive/src

set substitute-path /usr/sbin/app /apps/app/app

add-symbol-file /home/user/prj/stm32/main/bootloader/build/bin/main.elf
0x08005000

dir ../../out/build/app-undefined
dir ../../out/build/app-undefined/apps/app
dir ../../out/build/app-undefined/apps/app/app

set substitute-path /usr/lib ../../out/target/usr/lib

set args -m 0 -c /etc/conf.d/conf.xml

#sharedlibrary

#target extended-remote 192.168.1.62:12345

break myFunction
```

gdb-openocd.cfg

```
target extended-remote localhost:2331
#monitor reset
monitor halt

# Setup GDB FOR FASTER DOWNLOADS
#set remote memory-write-packet-size 1024
#set remote memory-write-packet-size fixed
```

```
# Add GDB access to mem range where VTOR is located
mem 0xE0000000 0xE00FFFFF
```

```
define bootapp
  monitor reset halt
  # Adjust VTOR (Vector table offset register)
  set {int}0xE000ED08 = &exception_table
  # Set SP/PC to the values from the actual vector table
  set $sp = *(int*)&exception_table
  set $pc = *((int*)(&exception_table)+1)
end
```

symbols

- set symbol-reloading on
- add-symbol-file ~/mymodule.o 0xd8be4000
- add-symbol-file ~/myanother.o 0x12341234

- set debug-file-directory <directory> - set directory contain symbol file
- show debug-file-directory

file paths and libraries

- set verbose on
- set auto-load safe-path /
- info shared
- set sysroot /home/niziak/n/3/out/host/usr/arm-buildroot-linux-uclibcgnueabi/sysroot

source paths

Search paths (prefixes):

- dir /path/to/src1
- dir /path/to/lib/src2

Translate beginning of paths:

- set substitute-path /usr/lib ../../out/target/usr/lib

Kernel OOPS

Reading

```
PC is at RTMPCheckEtherType+0x90/0x4d4 [mt7601Uapsta]
LR is at RTMPCheckEtherType+0x34/0x4d4 [mt7601Uapsta]

Code: e59f3418 e0256593 e2859d43 e289902c (e5d939ba)
```

Oops occurs at offset 0x90 from RTMPCheckEtherType. 0x4d4 is length. “Code” line shows last instruction. Instruction in bracket is problematic instruction (at RTMPCheckEtherType+0x90)

Tracing

Disassembly kernel binary or module binary: objdump -dS vmlinux > /tmp/kernel.s and look into generated code.

Python

```
apt-get install gdb python2.7-dbg
```

```
gdb python <pid>
```

DRAFTS

call raise(kernel-thread-id, signo) or call pthread_kill(pthread-thread-id, signo).