

Embedded Debugging Strategies

**Advanced Embedded Linux
Development**
with **Dan Walkes**



University of Colorado **Boulder**

Learning objectives:

Overview of GDB command line

Using Remote GDB

Debugging

- Use -g argument to GCC for debug symbols
- Use -O0 to disable optimizations if needed
 - Improves single stepping experience

Native Debugging

```
ecen5013@ecen5013-VirtualBox:~/segfaulters$ gdb ./segfaulters
GNU gdb (Ubuntu 8.1-0ubuntu3) 8.1.0.20180409-git
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./segfaulters...(no debugging symbols found)...done.
(gdb) r
Starting program: /home/ecen5013/segfaulters/segfaulters
The value of i is 0

Program received signal SIGSEGV, Segmentation fault.
__strlen_avx2 () at ../sysdeps/x86_64/multiarch/strlen-avx2.S:62
62      ../sysdeps/x86_64/multiarch/strlen-avx2.S: No such file or directory.
(gdb) bt
#0  __strlen_avx2 () at ../sysdeps/x86_64/multiarch/strlen-avx2.S:62
#1  0x00005555555546a2 in segfaulters ()
#2  0x00005555555546f0 in main ()
(gdb) |
```

GDB Commands

- r <args> - run
- b - set breakpoint
- p - print variable name
- n - step over next statement
- finish - step out of function
- info b - list breakpoints
- delete <x> - delete breakpoints
- bt - Backtrace

Remote GDB

- What about when we have a problem which exists on the target only?
 - BR2_PACKAGE_HOST_GDB
 - BR2_PACKAGE_GDB
 - BR2_PACKAGE_GDB_SERVER
 - Build image with required GDB packages and host tools

Remote GDB

```
# gdbserver :10000 ./hello-world  
Process hello-world created; pid = 103  
Listening on port 10000
```

- On the target, pick an unused port, start gdbserver and the application

```
$ arm-poky-linux-gnueabi-gdb hello-world
```

```
(gdb) target remote 192.168.1.101:10000
```

- On buildroot, start cross gdb
- Setup for remote debug on same port specified to gdbserver

Remote GDB with Buildroot

- The cross debugger is in `buildroot/output/host/bin/`

Then, on the host, you should start the cross gdb using the following command line:

```
<buildroot>/output/host/bin/<tuple>-gdb -x <buildroot>/output/staging/usr/share/buildroot/gdbinit foo
```


Assignment 5 Part 1 - Native Socket Server

**Advanced Embedded Linux
Development
with Dan Walkes**



University of Colorado **Boulder**

Assignment 5 Tips

**Advanced Embedded Linux
Development
with Dan Walkes**



University of Colorado **Boulder**

-Wall and -Werror

- Coding Guidelines:

Every source file should compile with zero warnings (Use -Werror/-Wall for solving bugs) or justification be provided for each warning.

Examples:

```
$ gcc -c main.o main.c -Wall -Werror
```

```
CFLAGS = -g -Wall -Werror
```

```
#include <stdio.h>
|
int main(void)
{
    int i=0;
    printf("The value of i is %d\n",i);
    segfault();
    return 0;
}
```

```
#include <string.h>
#include <stdio.h>

void segfaulter( char * somestring )
{
    size_t len=strlen(somestring);
    printf("The length of the passed string is %zu",len);
}
~
```

```
ecen5013@ecen5013-VirtualBox:~/segfaulter$ ./segfaulter
The value of i is 0
Segmentation fault (core dumped)
```

-Wall and -Werror

```
CFLAGS += -Wall -Werror
```

```
main.c: In function 'main':
```

```
main.c:7:2: error: implicit declaration of function 'segfaultter'; did you mean 'setbuffer'? [-Werror=implicit-function-declaration]
```

```
    segfaultter();
```

```
    ~~~~~
```

```
    setbuffer
```

```
cc1: all warnings being treated as errors
```

Assignment 5 hints/suggestions

- Remember to close file descriptors (from both `socket()` and `accept()`) to avoid memory leaks.
- Consider using `shutdown()` in your signal handler strategy.
- To avoid `bind()` failing with “Address Already In Use” use `SO_REUSEADDR`

<https://stackoverflow.com/questions/18173111/is-it-necessary-to-close-an-accept-return-file-descriptor>

<https://man7.org/linux/man-pages/man2/shutdown.2.html>

<https://stackoverflow.com/a/19239058/1446624>

<https://beej.us/guide/bgnet/html/#bind>

Assignment 5 Memory Leak Check



- Use Valgrind to check your program for memory leaks

- Pronounced “val grinned”

- Example execution for our aesdsocket:

```
valgrind --leak-check=full --show-leak-kinds=all --track-origins=yes --verbose --log-file=/tmp/valgrind-out.txt  
./aesdsocket
```

Assignment 5 Memory Leak Check



```
cat /tmp/valgrind-out.txt
```

```
==1634==  
==1634== HEAP SUMMARY:  
==1634==      in use at exit: 0 bytes in 0 blocks  
==1634==    total heap usage: 74 allocs, 74 frees, 134,659 bytes allocated  
==1634==  
==1634== All heap blocks were freed -- no leaks are possible  
==1634==  
==1634== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)  
==1634== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```