

## Module 4 – Assessment

### Code Debugging Tools (GDB and Valgrind)

- 1) Using Valgrind identify memleaks in the given program. Explore optional flags in Valgrind.
  - i. Install valgrind by typing “sudo apt install valgrind” in the terminal
  - ii. Then compile the program using the following command “gcc -g -o program.out program.c”
  - iii. Then type “valgrind program.out” to find mem leaks

```
ramana@ramana-VirtualBox: ~/Desktop/Programs
ramana@ramana-VirtualBox:~/Desktop/Programs$ sudo valgrind ./buggypgm.out
==3593== Memcheck, a memory error detector
==3593== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==3593== Using Valgrind 3.22.0 and LibVEX; rerun with -h for copyright info
==3593== Command: ./buggypgm.out
==3593==
==3593== Invalid read of size 4
==3593== at 0x109238: test1 (buggypgm.c:23)
==3593== by 0x109560: main (buggypgm.c:76)
==3593== Address 0x4a81068 is 40 bytes inside a block of size 400 free'd
==3593== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3593== by 0x109232: test1 (buggypgm.c:22)
==3593== by 0x109560: main (buggypgm.c:76)
==3593== Block was alloc'd at
==3593== at 0x4846828: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3593== by 0x10910f: test1 (buggypgm.c:14)
==3593== by 0x109560: main (buggypgm.c:76)
==3593==
Value of *ptr: 10
String: Good day to you!
==3593== Invalid write of size 4
==3593== at 0x109318: test3 (buggypgm.c:43)
==3593== by 0x109574: main (buggypgm.c:78)
==3593== Address 0x4a81704 is 4 bytes inside a block of size 200 free'd
==3593== at 0x109328: test3 (buggypgm.c:44)
==3593== by 0x109574: main (buggypgm.c:78)
==3593== Block was alloc'd at
==3593== at 0x4846828: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3593== by 0x10920c: test3 (buggypgm.c:37)
==3593== by 0x109574: main (buggypgm.c:78)
==3593==
==3593== Invalid free() / delete / delete[] / realloc()
==3593== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3593== by 0x109328: test3 (buggypgm.c:44)
==3593== by 0x109574: main (buggypgm.c:78)
==3593== Address 0x4a81700 is 0 bytes inside a block of size 200 free'd
==3593== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3593== by 0x109328: test3 (buggypgm.c:44)
```

```
ramana@ramana-VirtualBox: ~/Desktop/Programs
==3593== by 0x109328: test3 (buggypgm.c:44)
==3593== by 0x109574: main (buggypgm.c:78)
==3593== Block was alloc'd at
==3593== at 0x4846828: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3593== by 0x10920c: test3 (buggypgm.c:37)
==3593== by 0x109574: main (buggypgm.c:78)
==3593==
==3593== Invalid read of size 4
==3593== at 0x109382: test4 (buggypgm.c:59)
==3593== by 0x10957e: main (buggypgm.c:79)
==3593== Address 0x0 is not stack'd, malloc'd or (recently) free'd
==3593==
==3593== Process terminating with default action of signal 11 (SIGSEGV)
==3593== Access not within mapped region at address 0x0
==3593== at 0x109382: test4 (buggypgm.c:59)
==3593== by 0x10957e: main (buggypgm.c:79)
==3593==
==3593== If you believe this happened as a result of a stack
==3593== overflow in your program's main thread (unlikely but
==3593== possible), you can try to increase the size of the
==3593== main thread stack using the --main-stacksize= flag.
==3593== The main thread stack size used in this run was 8388608.
==3593==
==3593== HEAP SUMMARY:
==3593==   in use at exit: 1,124 bytes in 2 blocks
==3593== total heap usage: 5 allocs, 4 frees, 1,764 bytes allocated
==3593==
==3593== LEAK SUMMARY:
==3593==   definitely lost: 100 bytes in 1 blocks
==3593==   indirectly lost: 0 bytes in 0 blocks
==3593==   possibly lost: 0 bytes in 0 blocks
==3593==   still reachable: 1,024 bytes in 1 blocks
==3593==   suppressed: 0 bytes in 0 blocks
==3593==
==3593== Rerun with --leak-check=full to see details of leaked memory
==3593==
==3593== For lists of detected and suppressed errors, rerun with: -s
==3593== ERROR SUMMARY: 4 errors from 4 contexts (suppressed: 0 from 0)
Segmentation fault
ramana@ramana-VirtualBox:~/Desktop/Programs$
```

- iv. Trying out optional flags

## i. -leak-check

```
root@ramana-VirtualBox: /home/ramana/Desktop/Programs# valgrind ./buggypgn.out --leak-check = yes
==3914== Memcheck, a memory error detector
==3914== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==3914== Using valgrind-3.22.0 and libVEX; rerun with -h for copyright info
==3914== Command: ./buggypgn.out --leak-check = yes
==3914==
==3914== Invalid read of size 4
==3914== at 0x189238: test1 (buggypgn.c:23)
==3914== by 0x189560: main (buggypgn.c:76)
==3914== Address 0x4a81068 is 40 bytes inside a block of size 400 free'd
==3914== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3914== by 0x189232: test1 (buggypgn.c:22)
==3914== by 0x189560: main (buggypgn.c:76)
==3914== Block was alloc'd at
==3914== at 0x4846328: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3914== by 0x18910e: test1 (buggypgn.c:14)
==3914== by 0x189560: main (buggypgn.c:76)
==3914==
Value of *ptr: 10
String: Good day to you!
==3914== Invalid write of size 4
==3914== at 0x18931b: test3 (buggypgn.c:43)
==3914== by 0x189574: main (buggypgn.c:78)
==3914== Address 0x4a81704 is 4 bytes inside a block of size 200 free'd
==3914== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3914== by 0x189328: test3 (buggypgn.c:44)
==3914== by 0x189574: main (buggypgn.c:78)
==3914== Block was alloc'd at
==3914== at 0x4846328: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3914== by 0x18920c: test3 (buggypgn.c:37)
==3914== by 0x189574: main (buggypgn.c:78)
==3914==
==3914== Invalid free() / delete / delete[] / realloc()
==3914== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3914== by 0x189328: test3 (buggypgn.c:44)
==3914== by 0x189574: main (buggypgn.c:78)
==3914== Address 0x4a81700 is 0 bytes inside a block of size 200 free'd
==3914== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3914== by 0x189328: test3 (buggypgn.c:44)
```

## ii. -errors-for-leak-kinds

```
root@ramana-VirtualBox: /home/ramana/Desktop/Programs# valgrind ./buggypgn.out --errors-for-leak-kinds = indirect
==3990== Memcheck, a memory error detector
==3990== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==3990== Using valgrind-3.22.0 and libVEX; rerun with -h for copyright info
==3990== Command: ./buggypgn.out --errors-for-leak-kinds = indirect
==3990==
==3990== Invalid read of size 4
==3990== at 0x189238: test1 (buggypgn.c:23)
==3990== by 0x189560: main (buggypgn.c:76)
==3990== Address 0x4a81068 is 40 bytes inside a block of size 400 free'd
==3990== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3990== by 0x189232: test1 (buggypgn.c:22)
==3990== by 0x189560: main (buggypgn.c:76)
==3990== Block was alloc'd at
==3990== at 0x4846328: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3990== by 0x18910e: test1 (buggypgn.c:14)
==3990== by 0x189560: main (buggypgn.c:76)
==3990==
Value of *ptr: 10
String: Good day to you!
==3990== Invalid write of size 4
==3990== at 0x18931b: test3 (buggypgn.c:43)
==3990== by 0x189574: main (buggypgn.c:78)
==3990== Address 0x4a81704 is 4 bytes inside a block of size 200 free'd
==3990== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3990== by 0x189328: test3 (buggypgn.c:44)
==3990== by 0x189574: main (buggypgn.c:78)
==3990== Block was alloc'd at
==3990== at 0x4846328: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3990== by 0x18920c: test3 (buggypgn.c:37)
==3990== by 0x189574: main (buggypgn.c:78)
==3990==
==3990== Invalid free() / delete / delete[] / realloc()
==3990== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3990== by 0x189328: test3 (buggypgn.c:44)
==3990== by 0x189574: main (buggypgn.c:78)
==3990== Address 0x4a81700 is 0 bytes inside a block of size 200 free'd
==3990== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==3990== by 0x189328: test3 (buggypgn.c:44)
```

### iii. -track-origins

```
root@ramana-VirtualBox: /home/ramana/Desktop/Programs# valgrind ./buggypgn.out --track-origins = yes
==4113== Memcheck, a memory error detector
==4113== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==4113== Using Valgrind 3.22.0 and LibVEX; rerun with -h for copyright info
==4113== Command: ./buggypgn.out --track-origins = yes
==4113==
==4113== Invalid read of size 4
==4113== at 0x109230: test1 (buggypgn.c:23)
==4113== by 0x109560: main (buggypgn.c:76)
==4113== Address 0x4a81068 is 40 bytes inside a block of size 400 free'd
==4113== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==4113== by 0x109232: test1 (buggypgn.c:22)
==4113== by 0x109560: main (buggypgn.c:76)
==4113== Block was alloc'd at
==4113== at 0x4846828: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==4113== by 0x10910E: test1 (buggypgn.c:14)
==4113== by 0x109560: main (buggypgn.c:76)
==4113==
Value of *ptr: 10
String: Good day to you!
==4113== Invalid write of size 4
==4113== at 0x109310: test3 (buggypgn.c:43)
==4113== by 0x109574: main (buggypgn.c:78)
==4113== Address 0x4a81704 is 4 bytes inside a block of size 200 free'd
==4113== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==4113== by 0x109328: test3 (buggypgn.c:44)
==4113== by 0x109574: main (buggypgn.c:78)
==4113== Block was alloc'd at
==4113== at 0x4846828: malloc (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==4113== by 0x10920C: test3 (buggypgn.c:37)
==4113== by 0x109574: main (buggypgn.c:78)
==4113==
==4113== Invalid free() / delete / delete[] / realloc()
==4113== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==4113== by 0x109328: test3 (buggypgn.c:44)
==4113== by 0x109574: main (buggypgn.c:78)
==4113== Address 0x4a81700 is 0 bytes inside a block of size 200 free'd
==4113== at 0x484988f: free (in /usr/libexec/valgrind/vgpreload_memcheck-and64-linux.so)
==4113== by 0x109328: test3 (buggypgn.c:44)
```

- 2) With the same program, using GDB, set breakpoints, run the program, list the code, run from one breakpoint to another, print the value of variables while execution, check assemble code, disable breakpoints, check registers info, explore optional flags.

#### i. Set break points

```
root@ramana-VirtualBox: /home/ramana/Desktop/Programs# (gdb) run ./buggypgn.out
Starting program: ./buggypgn.out
No executable file specified.
Use the "file" or "exec-file" command.
(gdb) exit
root@ramana-VirtualBox: /home/ramana/Desktop/Programs# gdb ./buggypgn.out
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
Copyright (C) 2024 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 ./buggypgn.out...
(gdb) break 2
Breakpoint 1 at 0x10d: file buggypgn.c, line 14.
(gdb) break 4
Note: breakpoint 1 also set at pc 0x10d.
Breakpoint 2 at 0x10d: file buggypgn.c, line 14.
(gdb) break test1
Note: breakpoints 1 and 2 also set at pc 0x10d.
Breakpoint 3 at 0x10d: file buggypgn.c, line 14.
(gdb) break info
Function "info" not defined.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) info break
Num Type Disp Enb Address What
1 breakpoint keep y 0x00000000000010d in test1 at buggypgn.c:14
2 breakpoint keep y 0x00000000000010d in test1 at buggypgn.c:14
3 breakpoint keep y 0x00000000000010d in test1 at buggypgn.c:14
(gdb)
```



## v. Printing out the variables

```
root@ramana-VirtualBox:/home/ramana/Desktop/Programs
Starting program: /home/ramana/Desktop/Programs/buggyppn.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, test1 () at buggyppn.c:14
14      int ptr = malloc(sizeof(int) * 100);
(gdb) n
15      if (ptr == NULL) {
(gdb) n
19      for (int i = 0; i < 100; i++) {
(gdb) n

Breakpoint 2, test1 () at buggyppn.c:20
20      ptr[i] = i;
(gdb) print ptr
$1 = (int *) 0x555555502a0
(gdb) print *ptr
$2 = 0
(gdb) print i
$3 = 0
(gdb) n
19      for (int i = 0; i < 100; i++) {
(gdb) n

Breakpoint 2, test1 () at buggyppn.c:20
20      ptr[i] = i;
(gdb) n
19      for (int i = 0; i < 100; i++) {
(gdb) n

Breakpoint 2, test1 () at buggyppn.c:20
20      ptr[i] = i;
(gdb) print *ptr
$4 = 0
(gdb) print i
$5 = 2
(gdb) print *(ptr+i)
$6 = 1
(gdb)
```

## vi. Disassembling the program

```
root@ramana-VirtualBox:/home/ramana/Desktop/Programs
(gdb) disassemble
Dump of assembler code for function test1:
0x000055555551109 <@B>: endbr64
0x00005555555110a <@B>: push %rbp
0x00005555555110b <@B>: mov %rsp,%rbp
0x00005555555110c <@B>: sub $0x10,%rbp
0x00005555555110d <@B>: mov $0x100,%edi
0x00005555555110e <@B>: call 0x555555502a0 <malloc@plt>
0x00005555555110f <@B>: mov %rax,%rbx
0x000055555551110 <@B>: cmqp %rbx,%rbx
0x000055555551111 <@B>: jnz 0x55555551115 <test1+5b>
0x000055555551112 <@B>: lea 0x1(%rip),%rax <test1+8b> # 0x555555502a0
0x000055555551113 <@B>: mov %rax,%rdi
0x000055555551114 <@B>: call 0x555555502a0 <fprintf@plt>
0x000055555551115 <@B>: jmp 0x5555555111b <test1+13b>
0x000055555551116 <@B>: movl %ebx,%ecx
0x000055555551117 <@B>: jmp 0x5555555111b <test1+13b>
0x000055555551118 <@B>: mov %ecx,%eax
0x000055555551119 <@B>: cld
0x00005555555111a <@B>: lea 0x0(%rax,%rax,1),%rdx
0x00005555555111b <@B>: mov %rdi,%rbx
0x00005555555111c <@B>: add %rbx,%rdi
0x00005555555111d <@B>: mov %rdi,%eax
0x00005555555111e <@B>: addl %eax,%rdi
0x00005555555111f <@B>: cmpl %eax,%ecx
0x000055555551120 <@B>: jle 0x55555551124 <test1+59>
0x000055555551121 <@B>: mov %rdi,%rbx
0x000055555551122 <@B>: mov %rbx,%rdi
0x000055555551123 <@B>: call 0x555555502a0 <fprintf@plt>
0x000055555551124 <@B>: mov %rdi,%rbx
0x000055555551125 <@B>: addl %eax,%rdi
0x000055555551126 <@B>: mov (%rbx),%eax
0x000055555551127 <@B>: lea 0x1(%rip),%rax <test1+59> # 0x555555502a0
0x000055555551128 <@B>: mov %rax,%rdi
0x000055555551129 <@B>: mov %eax,%eax
0x00005555555112a <@B>: call 0x555555502a0 <printf@plt>
0x00005555555112b <@B>: leave
0x00005555555112c <@B>: retq
```

## vii. Deleting the breakpoints

```
root@ramana-VirtualBox:/home/ramana/Desktop/Programs
6.3      y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
(gdb) run
Starting program: /home/ramana/Desktop/Programs/buggyppn.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Value of *ptr: 10
String: Good day to you!
free(): double free detected in tcache 2

Breakpoint 4.2, __pthread_kill_implementation (no_tid=0, signo=6, threadid=140737353779688) at ./nptl/pthread_kill.c:31
warning: 31 ./nptl/pthread_kill.c: No such file or directory
(gdb) disable
(gdb) break info
Function "info" not defined.
Make breakpoint pending on future shared library load? (y or n) n
(gdb) info break
Num Type Disp Enb Address What
1 breakpoint keep n 0x000055555551109 in test1 at buggyppn.c:14
3 breakpoint keep n 0x00005555555110f in test3 at buggyppn.c:37
4 breakpoint keep n <MULTIPLE>
4 breakpoint already hit 1 time
4.1 y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
4.2 y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
4.3 y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
5 breakpoint keep n <MULTIPLE>
5 breakpoint already hit 1 time
5.1 y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
5.2 y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
5.3 y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
6 breakpoint keep n <MULTIPLE>
6 breakpoint already hit 1 time
6.1 y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
6.2 y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
6.3 y 0x00007ffff70d573c in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
(gdb) delete
Delete all breakpoints, watchpoints, tracepoints, and catchpoints? (y or n) y
(gdb) info break
No breakpoints, watchpoints, tracepoints, or catchpoints.
(gdb)
```

## viii. Viewing the register

```
root@ramana-VirtualBox: /home/ramana/Desktop/Programs
5.2 y- 0x00007ffffc9ea30 in __pthread_kill_implementation at ./nptl/pthread_kill.c:30
5.3 y- 0x00007ffffc9ea30 in __pthread_kill_implementation at ./nptl/pthread_kill.c:30
6 breakpoint keep n <MULTIPLE>
6 breakpoint already hit 1 time
6.1 y- 0x00007ffffc9ea30 in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
6.2 y- 0x00007ffffc9ea30 in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
6.3 y- 0x00007ffffc9ea30 in __pthread_kill_implementation at ./nptl/pthread_kill.c:28
(gdb) delete
Delete all breakpoints, watchpoints, tracepoints, and catchpoints? (y or n) y
(gdb) info break
No breakpoints, watchpoints, tracepoints, or catchpoints.
(gdb) info registers
rax      0x0          0
rbx      0xffffffffa740 140737353779008
rcx      0x0          0
rdx      0xffffffffe6 4294967270
rsi      0x6          6
rdi      0xffffffffa740 140737353779008
rbp      0xfffffffffd90 0x7fffffd90
rsp      0x7fffffd90 0x7fffffd90
r8       0x0          0
r9       0xfffffffffb40 140737354118208
r10      0x0          0
r11      0x246        582
r12      0x6          6
r13      0xffffffffe0c0 140737488347328
r14      0x16        22
r15      0x7fffffe0c0 140737488347328
rip      0xffffffffc9ea30 0x7ffffc9ea30 <_GI__pthread_kill+56>
eflags   0x202        [ SF IF ]
cs       0x33        51
ss       0x2b        43
ds       0x0          0
es       0x0          0
fs       0x0          0
gs       0x0          0
fs_base  0xffffffffa740 140737353779008
gs_base  0x0          0
(gdb)
```