Winston Shine

Operating Systems

Lab 4

4/23/2024

1. segfault occurs, this is expected because we are tying to access memory that has not been allocated
2. we get more information from gdb because we compiled the program with the -g flag, such as function name, memory address, and even code snippet where the error occured.
3.

```
==10290== Use of uninitialised value of size 8
==10290==    at 0x401139: main (null.c:5)
==10290==
==10290== Invalid read of size 4
==10290==    at 0x401139: main (null.c:5)
==10290==  Address 0x0 is not stack'd, malloc'd or (recently) free'd
==10290==
==10290==
==10290== Process terminating with default action of signal 11 (SIGSEGV): dumping core
==10290==  Access not within mapped region at address 0x0
==10290==    at 0x401139: main (null.c:5)
==10290==  If you believe this happened as a result of a stack
==10290==  overflow in your program's main thread (unlikely but
==10290==  possible), you can try to increase the size of the
==10290==  main thread stack using the --main-stacksize= flag.
==10290==  The main thread stack size used in this run was 8388608.
==10290==
==10290== HEAP SUMMARY:
==10290==     in use at exit: 0 bytes in 0 blocks
==10290==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
```

we got additional information suck as: size of uninitialized value, size of of the invalid read, the current stack size, and a summary of the heap memory,

4.

```
==14156== HEAP SUMMARY:
==14156==     in use at exit: 1,024 bytes in 1 blocks
==14156==   total heap usage: 1 allocs, 0 frees, 1,024 bytes allocated
==14156==
==14156== 1,024 bytes in 1 blocks are definitely lost in loss record 1 of 1
==14156==    at 0x484376B: malloc (in /nix/store/78zxvxfy6klvwfc2s95y71y0b284fd6v-valgri
==14156==    by 0x40113E: main (null.c:3)
==14156==
```

```
==14156== LEAK SUMMARY:
==14156==    definitely lost: 1,024 bytes in 1 blocks
==14156==    indirectly lost: 0 bytes in 0 blocks
==14156==      possibly lost: 0 bytes in 0 blocks
==14156==    still reachable: 0 bytes in 0 blocks
==14156==         suppressed: 0 bytes in 0 blocks
```

gdb found no problems, the process ran successuflly valgrind shows the heap allocated memory and the numbers of allocs and frees valgrind also shows us the amount of memory lost.

5. it seems to run fine, and valgrind detects no errors. I think this is incorrect. I tested this by assigning values to some of the indexes of the array and printing them right at the end of the program. it still ran fine, and the numbers were correct but valgrind reports that the memory was freed when I clearly did not. I get that something is getting overwritten by the out of bounds indexing but it's unclear to me why valgrind detects this as a free.

lab questions

3. valgrind can tell us information about the amount of memory allocations and frees, and sometimes tell us if there is a memory leak.