Memory leaks and errors practice session:

Typographical conventions

We use the following conventions in this guide:

emacs The name of a specific command or file

file You should replace file with a specific name

Exit abc Output that you see on the screen

valgrind command

valgrind --tool=memcheck –-leak-check=yes ./sample1

valgrind –-leak-check=yes ./sample1

1. Login into the Linux server with your login Ids

2. Create a new directory called mem_errors in your home directory <home>

mkdir mem_errors

3. Go inside the directory you have created in (2) /<home>/mem_errors

cd mem_errors

4. Copy the following files from the path as mentioned by the trainer:

a. sample1.c

b. sample2.c

c. sample3.c

d. sample4.c

e. sample5.c

5. Take a look at the example programs and observe if you can find any errors.

Compilation

6. Compile the files

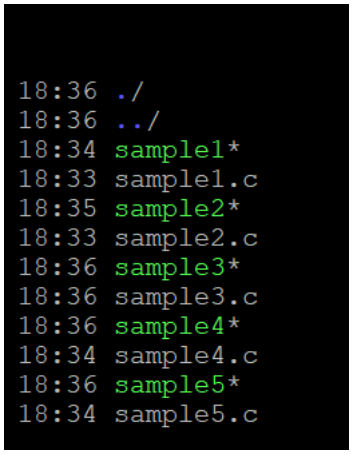
gcc –o sample1 –g sample1.c

gcc –o sample2 –g sample2.c

gcc –o sample3 –g sample3.c

gcc –o sample4 –g sample4.c

gcc –o sample5 –g sample5.c



Execution

7. Execute the file sample1

valgrind –-leak-check=yes sample1

After quitting the program Valgrind displays a list of memory errors in the program. Analyze these errors.

```
==9391== Memcheck, a memory error detector
==9391== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9391== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9391== Command: ./sample1
==9391==
Please enter your user name: pooja
data2 :pooja:
Please enter your user name: hello
data2 :hello:
Please enter your user name: 123
data2 :123:
Please enter your user name: ^C==9391==
==9391== Process terminating with default action of signal 2 (SIGINT)
==9391==    at 0x4F4E031: read (read.c:27)
==9391==    by 0x4ECB0F7: _IO_file_underflow@@GLIBC_2.2.5 (fileops.c:531)
==9391==    by 0x4ECC3A1: _IO_default_uflow (genops.c:380)
==9391==    by 0x4EA9B99: _IO_vfscanf (vfscanf.c:630)
==9391==    by 0x4EB9F87: __isoc99_scanf (isoc99_scanf.c:37)
==9391==    by 0x1087D9: main (sample1.c:23)
==9391==
==9391== HEAP SUMMARY:
==9391==     in use at exit: 112 bytes in 7 blocks
==9391==   total heap usage: 9 allocs, 2 frees, 2,160 bytes allocated
==9391==
==9391== 32 bytes in 2 blocks are definitely lost in loss record 3 of 4
==9391==    at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9391==    by 0x1087FA: main (sample1.c:30)
==9391==
==9391== 48 bytes in 3 blocks are definitely lost in loss record 4 of 4
==9391==    at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9391==    by 0x10879B: main (sample1.c:19)
==9391==
==9391== LEAK SUMMARY:
==9391==    definitely lost: 80 bytes in 5 blocks
==9391==    indirectly lost: 0 bytes in 0 blocks
==9391==      possibly lost: 0 bytes in 0 blocks
==9391==    still reachable: 32 bytes in 2 blocks
==9391==         suppressed: 0 bytes in 0 blocks
==9391== Reachable blocks (those to which a pointer was found) are not shown.
==9391== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==9391==
==9391== For counts of detected and suppressed errors, rerun with: -v
==9391== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```

8. Execute the file sample2

valgrind –-leak-check=yes ./sample2

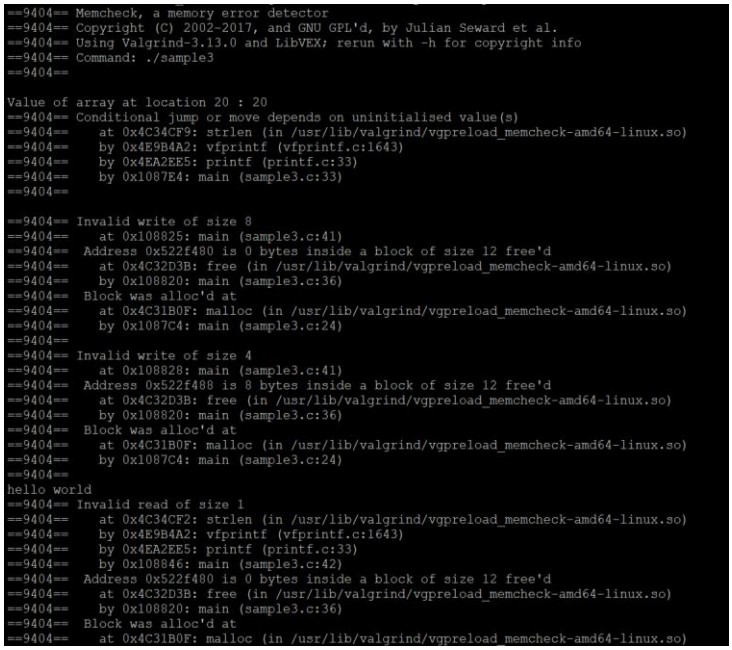
After quitting the program Valgrind displays a list of memory errors in the program. Analyze these errors.

```
==9399== Memcheck, a memory error detector
==9399== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9399== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9399== Command: ./sample2
==9399==
Please input your username: pooja
data2 :pooja:
==9399== Invalid read of size 1
==9399==    at 0x4C34CF2: strlen (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9399==    by 0x4E9B4A2: vfprintf (vfprintf.c:1643)
==9399==    by 0x4EA2EE5: printf (printf.c:33)
==9399==    by 0x108868: main (sample2.c:34)
==9399==  Address 0x522f050 is 0 bytes after a block of size 16 alloc'd
==9399==    at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9399==    by 0x1087A2: main (sample2.c:16)
==9399==
tmp ::
==9399==
==9399== HEAP SUMMARY:
==9399==     in use at exit: 0 bytes in 0 blocks
==9399==   total heap usage: 4 allocs, 4 frees, 2,080 bytes allocated
==9399==
==9399== All heap blocks were freed -- no leaks are possible
==9399==
==9399== For counts of detected and suppressed errors, rerun with: -v
==9399== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

9. Execute the file sample3

Valgrind –-leak-check=yes ./sample3

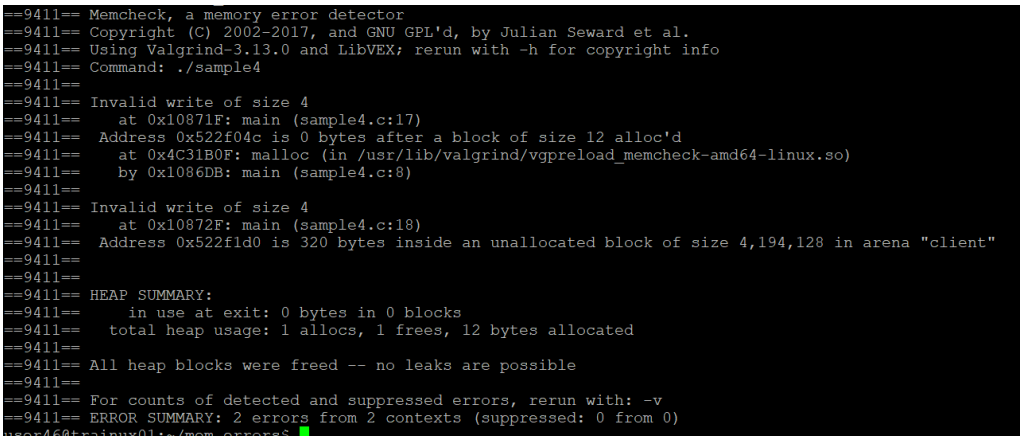
After quitting the program Valgrind displays a list of memory errors in the program. Analyze these errors.



10. Execute the file sample4

Valgrind –-leak-check=yes ./sample4

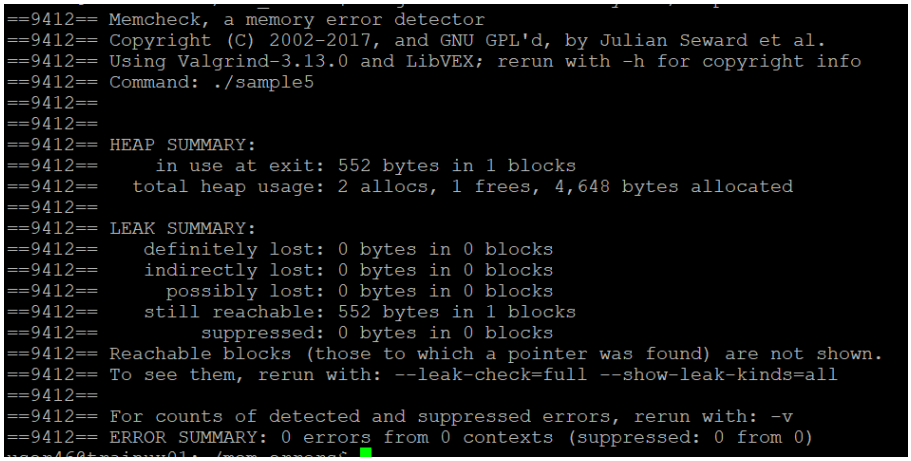
After quitting the program Valgrind displays a list of memory errors in the program. Analyze these errors.



11. Execute the file sample5

Valgrind –-leak-check=yes ./sample5

After quitting the program Valgrind displays a list of memory errors in the program. Analyze these errors.



12. Compile and execute the program in queue_linked_list_memory_leak

valgrind –leak-check=yes ./queue_llist

After quitting the program Valgrind displays a list of memory errors in the program. Analyze these errors.