

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

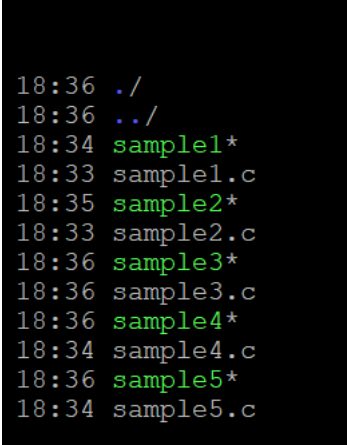
```
gcc -o sample1 -g sample1.c
```

```
gcc -o sample2 -g sample2.c
```

```
gcc -o sample3 -g sample3.c
```

```
gcc -o sample4 -g sample4.c
```

```
gcc -o sample5 -g sample5.c
```

A terminal window with a black background and white text. It shows a directory listing of files named sample1 through sample5. The files are listed in two columns. The first column shows the time (18:36) and the file name with an asterisk (sample1*, sample2*, sample3*, sample4*, sample5*). The second column shows the file name without an asterisk (sample1.c, sample2.c, sample3.c, sample4.c, sample5.c).

```
18:36 ./
18:36 ../
18:34 sample1*
18:33 sample1.c
18:35 sample2*
18:33 sample2.c
18:36 sample3*
18:36 sample3.c
18:36 sample4*
18:34 sample4.c
18:36 sample5*
18:34 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.

```

==9404== Memcheck, a memory error detector
==9404== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9404== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9404== Command: ./sample3
==9404==

Value of array at location 20 : 20
==9404== Conditional jump or move depends on uninitialised value(s)
==9404== at 0x4C34CF9: strlen (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9404== by 0x4E9B4A2: vfprintf (vfprintf.c:1643)
==9404== by 0x4EA2EE5: printf (printf.c:33)
==9404== by 0x1087E4: main (sample3.c:33)
==9404==

==9404== Invalid write of size 8
==9404== at 0x108825: main (sample3.c:41)
==9404== Address 0x522f480 is 0 bytes inside a block of size 12 free'd
==9404== at 0x4C32D3B: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9404== by 0x108820: main (sample3.c:36)
==9404== Block was alloc'd at
==9404== at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9404== by 0x1087C4: main (sample3.c:24)
==9404==

==9404== Invalid write of size 4
==9404== at 0x108828: main (sample3.c:41)
==9404== Address 0x522f488 is 8 bytes inside a block of size 12 free'd
==9404== at 0x4C32D3B: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9404== by 0x108820: main (sample3.c:36)
==9404== Block was alloc'd at
==9404== at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9404== by 0x1087C4: main (sample3.c:24)
==9404==

hello world
==9404== Invalid read of size 1
==9404== at 0x4C34CF2: strlen (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9404== by 0x4E9B4A2: vfprintf (vfprintf.c:1643)
==9404== by 0x4EA2EE5: printf (printf.c:33)
==9404== by 0x108846: main (sample3.c:42)
==9404== Address 0x522f480 is 0 bytes inside a block of size 12 free'd
==9404== at 0x4C32D3B: free (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9404== by 0x108820: main (sample3.c:36)
==9404== Block was alloc'd at
==9404== at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)

```

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.

```

==9411== Memcheck, a memory error detector
==9411== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9411== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9411== Command: ./sample4
==9411==

==9411== Invalid write of size 4
==9411== at 0x10871F: main (sample4.c:17)
==9411== Address 0x522f04c is 0 bytes after a block of size 12 alloc'd
==9411== at 0x4C31B0F: malloc (in /usr/lib/valgrind/vgpreload_memcheck-amd64-linux.so)
==9411== by 0x1086DB: main (sample4.c:8)
==9411==

==9411== Invalid write of size 4
==9411== at 0x10872F: main (sample4.c:18)
==9411== Address 0x522f1d0 is 320 bytes inside an unallocated block of size 4,194,128 in arena "client"
==9411==

==9411== HEAP SUMMARY:
==9411== in use at exit: 0 bytes in 0 blocks
==9411== total heap usage: 1 allocs, 1 frees, 12 bytes allocated
==9411==

==9411== All heap blocks were freed -- no leaks are possible
==9411==

==9411== For counts of detected and suppressed errors, rerun with: -v
==9411== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)

```

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.

```

==9412== Memcheck, a memory error detector
==9412== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==9412== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==9412== Command: ./sample5
==9412==
==9412==
==9412== HEAP SUMMARY:
==9412==   in use at exit: 552 bytes in 1 blocks
==9412==   total heap usage: 2 allocs, 1 frees, 4,648 bytes allocated
==9412==
==9412== LEAK SUMMARY:
==9412==   definitely lost: 0 bytes in 0 blocks
==9412==   indirectly lost: 0 bytes in 0 blocks
==9412==   possibly lost: 0 bytes in 0 blocks
==9412==   still reachable: 552 bytes in 1 blocks
==9412==   suppressed: 0 bytes in 0 blocks
==9412== Reachable blocks (those to which a pointer was found) are not shown.
==9412== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==9412==
==9412== For counts of detected and suppressed errors, rerun with: -v
==9412== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

```

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.