C / C++ Programming

Lab # 07
Memory Management & Pointers

Name	*Muddassir Ali Siddiqui*
Instructor	*Miss Hira Sohail*
Date	*17th July 2025*

1. In-Lab Tasks: (*Write your lab task & screenshots here*)

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

[cc@ncdc-0063 labs]$ cd lab7/Lab_7_files_memory_management/
[cc@ncdc-0063 Lab_7_files_memory_management]$ ls
linked_list.c  linked_list.h  Makefile  test_linked_list.c  test_vector.c  vector.c  vector.h
[cc@ncdc-0063 Lab_7_files_memory_management]$ make
gcc -Wall -std=c99 -g -c linked_list.c
gcc -Wall -std=c99 -g -c test_linked_list.c
gcc -o linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab_7_files_memory_management]$
```

```
[cc@ncdc-0063 Lab_7_files_memory_management]$ make
make: Nothing to be done for 'all'.
[cc@ncdc-0063 Lab_7_files_memory_management]$
```

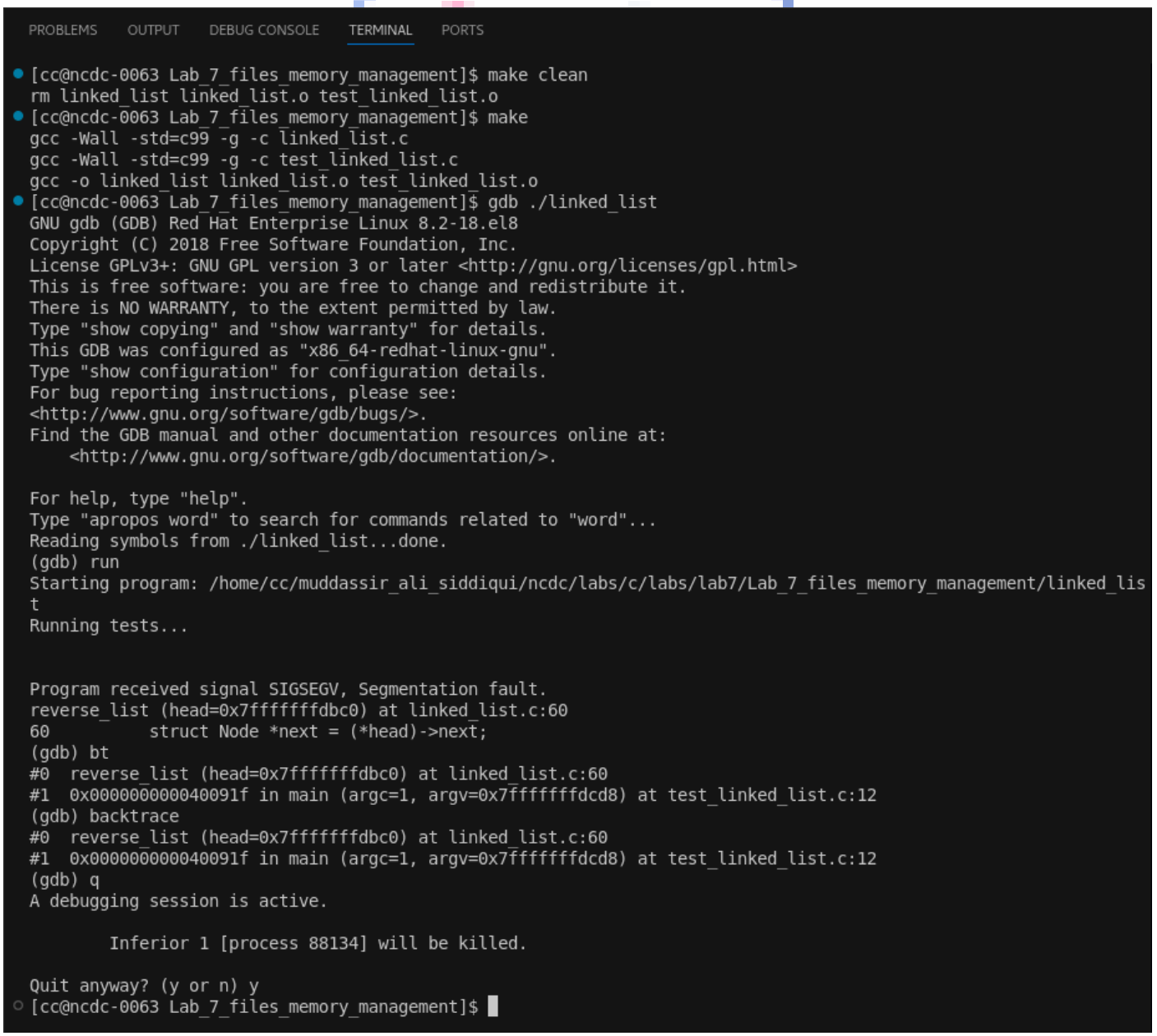
```
[cc@ncdc-0063 Lab_7_files_memory_management]$ make clean
rm linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab 7 files memory management]$
```

i. Task 1:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
[cc@ncdc-0063 Lab_7_files_memory_management]$ make
gcc -Wall -std=c99 -g -c linked_list.c
gcc -Wall -std=c99 -g -c test_linked_list.c
gcc -o linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab_7_files_memory_management]$ valgrind ./linked_list
==87442== Memcheck, a memory error detector
==87442== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==87442== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==87442== Command: ./linked_list
==87442==
Running tests...
==87442== Invalid read of size 8
==87442== at 0x400840: reverse_list (linked_list.c:60)
==87442== by 0x40091E: main (test_linked_list.c:12)
==87442== Address 0x8 is not stack'd, malloc'd or (recently) free'd
==87442==
==87442==
==87442== Process terminating with default action of signal 11 (SIGSEGV)
==87442== Access not within mapped region at address 0x8
==87442== at 0x400840: reverse_list (linked_list.c:60)
==87442== by 0x40091E: main (test_linked_list.c:12)
==87442== If you believe this happened as a result of a stack
==87442== overflow in your program's main thread (unlikely but
==87442== possible), you can try to increase the size of the
==87442== main thread stack using the --main-stacksize= flag.
==87442== The main thread stack size used in this run was 8388608.
==87442==
==87442== HEAP SUMMARY:
==87442== in use at exit: 1,024 bytes in 1 blocks
==87442== total heap usage: 1 allocs, 0 frees, 1,024 bytes allocated
==87442==
==87442== LEAK SUMMARY:
==87442== definitely lost: 0 bytes in 0 blocks
==87442== indirectly lost: 0 bytes in 0 blocks
==87442== possibly lost: 0 bytes in 0 blocks
==87442== still reachable: 1,024 bytes in 1 blocks
==87442== suppressed: 0 bytes in 0 blocks
==87442== Rerun with --leak-check=full to see details of leaked memory
==87442==
==87442== For lists of detected and suppressed errors, rerun with: -s
==87442== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
Segmentation fault (core dumped)
[cc@ncdc-0063 Lab_7_files_memory_management]$

```
[cc@ncdc-0063 Lab_7_files_memory_management]$ valgrind --leak-check=full ./linked_list
==87798== Memcheck, a memory error detector
==87798== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==87798== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==87798== Command: ./linked_list
==87798==
Running tests...

==87798== Invalid read of size 8
==87798==    at 0x400840: reverse_list (linked_list.c:60)
==87798==    by 0x40091E: main (test_linked_list.c:12)
==87798==  Address 0x8 is not stack'd, malloc'd or (recently) free'd
==87798==
==87798==
==87798== Process terminating with default action of signal 11 (SIGSEGV)
==87798==  Access not within mapped region at address 0x8
==87798==    at 0x400840: reverse_list (linked_list.c:60)
==87798==    by 0x40091E: main (test_linked_list.c:12)
==87798==  If you believe this happened as a result of a stack
==87798==  overflow in your program's main thread (unlikely but
==87798==  possible), you can try to increase the size of the
==87798==  main thread stack using the --main-stacksize= flag.
==87798==  The main thread stack size used in this run was 8388608.
==87798==
==87798== HEAP SUMMARY:
==87798==     in use at exit: 1,024 bytes in 1 blocks
==87798==   total heap usage: 1 allocs, 0 frees, 1,024 bytes allocated
==87798==
==87798== LEAK SUMMARY:
==87798==    definitely lost: 0 bytes in 0 blocks
==87798==    indirectly lost: 0 bytes in 0 blocks
==87798==      possibly lost: 0 bytes in 0 blocks
==87798==    still reachable: 1,024 bytes in 1 blocks
==87798==         suppressed: 0 bytes in 0 blocks
==87798== Reachable blocks (those to which a pointer was found) are not shown.
==87798== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==87798==
==87798== For lists of detected and suppressed errors, rerun with: -s
==87798== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
Segmentation fault (core dumped)
[cc@ncdc-0063 Lab_7_files_memory_management]$
```


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
[cc@ncdc-0063 Lab_7_files_memory_management]$ make clean
rm linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab_7_files_memory_management]$ make
gcc -Wall -std=c99 -g -c linked_list.c
gcc -Wall -std=c99 -g -c test_linked_list.c
gcc -o linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab_7_files_memory_management]$ gdb ./linked_list
GNU gdb (GDB) Red Hat Enterprise Linux 8.2-18.el8
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-redhat-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 ./linked_list...done.
(gdb) run
Starting program: /home/cc/muddassir_ali_siddiqui/ncdc/labs/c/labs/lab7/Lab_7_files_memory_management/linked_lis
t
Running tests...
Program received signal SIGSEGV, Segmentation fault.
reverse_list (head=0x7fffffffdbc0) at linked_list.c:60
60 struct Node *next = (*head)->next;
(gdb) bt
#0 reverse_list (head=0x7fffffffdbc0) at linked_list.c:60
#1 0x000000000040091f in main (argc=1, argv=0x7fffffffdcd8) at test_linked_list.c:12
(gdb) backtrace
#0 reverse_list (head=0x7fffffffdbc0) at linked_list.c:60
#1 0x000000000040091f in main (argc=1, argv=0x7fffffffdcd8) at test_linked_list.c:12
(gdb) q
A debugging session is active.
Inferior 1 [process 88134] will be killed.
Quit anyway? (y or n) y
[cc@ncdc-0063 Lab_7_files_memory_management]$

```
[cc@ncdc-0063 Lab_7_files_memory_management]$ make clean
rm linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab_7_files_memory_management]$ make
gcc -Wall -std=c99 -g -c linked_list.c
gcc -Wall -std=c99 -g -c test_linked_list.c
gcc -o linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab_7_files_memory_management]$ valgrind ./linked_list
==88583== Memcheck, a memory error detector
==88583== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==88583== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==88583== Command: ./linked_list
==88583==
Running tests...

Congrats! You have passed the reverse_list test!

==88583== Use of uninitialised value of size 8
==88583==    at 0x4008EE: add_to_back (linked_list.c:82)
==88583==    by 0x4009F8: main (test_linked_list.c:30)
==88583==
==88583==
==88583== Process terminating with default action of signal 11 (SIGSEGV)
==88583==  Bad permissions for mapped region at address 0x400628
==88583==    at 0x4008EE: add_to_back (linked_list.c:82)
==88583==    by 0x4009F8: main (test_linked_list.c:30)
==88583==
==88583== HEAP SUMMARY:
==88583==     in use at exit: 1,040 bytes in 2 blocks
==88583==   total heap usage: 7 allocs, 5 frees, 1,120 bytes allocated
==88583==
==88583== LEAK SUMMARY:
==88583==    definitely lost: 0 bytes in 0 blocks
==88583==    indirectly lost: 0 bytes in 0 blocks
==88583==      possibly lost: 0 bytes in 0 blocks
==88583==    still reachable: 1,040 bytes in 2 blocks
==88583==         suppressed: 0 bytes in 0 blocks
==88583== Rerun with --leak-check=full to see details of leaked memory
==88583==
==88583== Use --track-origins=yes to see where uninitialised values come from
==88583== For lists of detected and suppressed errors, rerun with: -s
==88583== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
Segmentation fault (core dumped)
[cc@ncdc-0063 Lab_7_files_memory_management]$
```

ii. Task 2:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

[cc@ncdc-0063 labs]$ cd lab7/Lab_7_files_memory_management/
[cc@ncdc-0063 Lab_7_files_memory_management]$ ls
linked_list.c  linked_list.h  Makefile  test_linked_list.c  test_vector.c  vector.c  vector.h
[cc@ncdc-0063 Lab_7_files_memory_management]$ make
gcc -Wall -std=c99 -g -c linked_list.c
gcc -Wall -std=c99 -g -c test_linked_list.c
gcc -o linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab_7_files_memory_management]$
```

```
[cc@ncdc-0063 Lab_7_files_memory_management]$ gdb ./linked_list
GNU gdb (GDB) Red Hat Enterprise Linux 8.2-18.el8
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-redhat-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 ./linked_list...done.
(gdb) run
Starting program: /home/cc/muddassir_ali_siddiqui/ncdc/labs/c/labs/lab7/Lab_7_files_memory_management/linked_lis
t
Running tests...

Congrats! You have passed the reverse_list test!


Program received signal SIGSEGV, Segmentation fault.
0x00000000004008ee in add_to_back (head=0x7fffffffdb98, data=15) at linked_list.c:82
82          prev->next = new_node;
(gdb) bt
#0  0x00000000004008ee in add_to_back (head=0x7fffffffdb98, data=15) at linked_list.c:82
#1  0x00000000004009f9 in main (argc=1, argv=0x7fffffffdcd8) at test_linked_list.c:30
(gdb) q
A debugging session is active.

        Inferior 1 [process 88740] will be killed.

Quit anyway? (y or n) y
[cc@ncdc-0063 Lab_7_files_memory_management]$
```

```
[cc@ncdc-0063 Lab_7_files_memory_management]$ make clean
rm linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab_7_files_memory_management]$ make
gcc -Wall -std=c99 -g -c linked_list.c
gcc -Wall -std=c99 -g -c test_linked_list.c
gcc -o linked_list linked_list.o test_linked_list.o
[cc@ncdc-0063 Lab_7_files_memory_management]$ valgrind ./linked_list
==89664== Memcheck, a memory error detector
==89664== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==89664== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==89664== Command: ./linked_list
==89664==
Running tests...

Congrats! You have passed the reverse_list test!

Congrats! All of the test cases passed!
==89664==
==89664== HEAP SUMMARY:
==89664==     in use at exit: 0 bytes in 0 blocks
==89664==   total heap usage: 9 allocs, 9 frees, 1,152 bytes allocated
==89664==
==89664== All heap blocks were freed -- no leaks are possible
==89664==
==89664== For lists of detected and suppressed errors, rerun with: -s
==89664== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
[cc@ncdc-0063 Lab_7_files_memory_management]$
```

iii. Task 3:

```
[cc@ncdc-0063 vector]$ make clean
rm test_vector vector.o test_vector.o
[cc@ncdc-0063 vector]$ make
gcc -Wall -std=c99 -g -c vector.c
gcc -Wall -std=c99 -g -c test_vector.c
gcc -o test_vector vector.o test_vector.o
[cc@ncdc-0063 vector]$ valgrind --leak-check=full ./test_vector
==112971== Memcheck, a memory error detector
==112971== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==112971== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==112971== Command: ./test_vector
==112971==
Calling vector_new()
Calling vector_delete()
Calling vector_new() again

These should all return 0 (vector_get()): 0, 0, 0

Doing a bunch of vector_set()s

These should be equal:
98 = 98
15 = 15
65 = 65
-123 = -123
21 = 21
43 = 43
0 = 0
0 = 0
0 = 0
3 = 3

Congratulations!!! Test complete.

==112971==
==112971== HEAP SUMMARY:
==112971==     in use at exit: 0 bytes in 0 blocks
==112971==   total heap usage: 9 allocs, 9 frees, 3,280 bytes allocated
==112971==
==112971== All heap blocks were freed -- no leaks are possible
==112971==
==112971== For lists of detected and suppressed errors, rerun with: -s
==112971== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
[cc@ncdc-0063 vector]$
```

2. Critical Analysis: (*Write you critical analysis / conclusion here*)

In this lab we learn how to write the makefile. By making a makefile we can run multiple and particular tests one by one. Just hit the make and set the target you want to perform.e.g. make all: run all the tests. Make clean delete all the dependencies which create during make all.