Operating Systems
EE463 – Spring 2023
Lab 4 Report

Khaled Dahhasi - 1935129
Thursday, June 8, 2023

In this lab we were tasked to fix a code after investigating the issues by debugging it.

Part 1)

We look for an error that causes memory leak. First lets look at the current output of the current main.c program provided in the lab.

```
Destroying Node, 3 are in existence right now
4
3
-417676544
2

Destroying Node, 2 are in existence right now
4
3
-417676608
3

Destroying Node, 1 are in existence right now
-417676608
4
Destroying Node, 0 are in existence right now
Destroying Node, -1 are in existence right now
Destroying Node, -2 are in existence right now
Destroying Node, -3 are in existence right now
Destroying Node, -4 are in existence right now
Destroying Node, -5 are in existence right now
Destroying Node, -6 are in existence right now
Destroying Node, -7 are in existence right now
Destroying Node, -8 are in existence right now
Destroying Node, -9 are in existence right now
free(): invalid pointer
Aborted
```

There are two bugs here, one is the memory leak where a memory address is printed, this can be fixed if we delete the correct node. As in the original code, the node deleted is wrong:

```
temp->next(marker->next());
delete temp;
temp = 0;
```

the node that should be deleted is marker not temp, as temp refers to the node that precedes the node we want to delete. Marker is what refers to the node we actually want to delete. Thus, the fix is:

```
temp->next (marker->next());
 delete marker; //fix
 temp = 0;
 return 0;
```

another bug is the fact that the deletion doesn't stop, this is because in the remove method, a new node is constantly being made. We need to fix that:

```
else {
 head_ = new Node(marker->value(), marker->next());
 delete marker;
 marker = 0;
}
```

The fix:

```
else {
 head_ = marker->next(); //fix
 delete marker;
```

```
 marker = 0;


}
```

The correct output:

```
KhaleDahhasi@lamp ~/lab4$ ./main
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Destroying Node, 3 are in existence right now
3
2
1

Destroying Node, 2 are in existence right now
3
2

Destroying Node, 1 are in existence right now
3

Destroying Node, 0 are in existence right now
```

Part two:

modified driver:

```
int main (int argc, char **argv) {


        LinkedList *list = new LinkedList ();

 list->insert (1);

 list->insert (2);

 list->insert (3);

 list->insert (4);

 printf("%s\n", "The fully created list is:");

 list->print ();

 printf("\n%s\n", "Now removing elements:");

 list->remove (2);

 list->print ();

 return 0;

}
```

output:

Khaled Dahhasi - 1935129
Thursday, June 8, 2023

```
KhaleDahhasi@lamp ~/lab4$ ./main
Creating Node, 1 are in existence right now
Creating Node, 2 are in existence right now
Creating Node, 3 are in existence right now
Creating Node, 4 are in existence right now
The fully created list is:
4
3
2
1

Now removing elements:
Destroying Node, 3 are in existence right now
4
3
1
```

1,2,3,4 are inserted then 2 is removed and the output is correct. So perhaps in the first part I solved both bugs 😊