CSE31 : Lab #4 – C

This lab contains two parts. To ensure you get full credit, make sure you read this lab carefully and follow the instructions precisely.

Overview

This lab will look at strings in C and linked-list. The idea to investigate C's implementation of underlying data structures.

(Exercise) Out of Bounds

Consider the following code implemented in hello.c :

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
  char hello[] = "hello ", world[] = "world!\n", *s;
  s = strcat(hello,world);
  printf(s);
  return 0;
}
```

Q1. When you run the code, it fails to give the expected result, why?

Q2. What's one way to fix this bug?

Q3. How do you run a program in gdb?Why doesn't C provide the same kind of bounds-checking on arrays that Java does?

Q4. What is a pro and a con for implementing strings as dumb arrays rather than smart objects like in Java?

(Exercise) Cyclic Linked list

In cyclic_ll.c, complete the function has_cycle() to implement the following algorithm for checking if a singly-linked list has a cycle. Recall that if p is a pointer to a struct, p->member refers to a member variable in the struct, and is equivalent to (*p).member.

1) Start with two pointers at the head of the list

2) On each iteration, increment the first pointer by one node and the second pointer by two nodes. If it is not possible to do one or both of these because of a null pointer, then we know there is an end to the list and there is therefore no cycle.

3) We know there is a cycle if

   a) The second pointer is the same as the first pointer

   b) The next node of the second pointer is pointed to by the first pointer

The reason we know there is a cycle with the two conditions in 3) is that second pointer has wrapped around to the first one in the circle and we have detected it. After you have correctly implemented has_cycle, the program you get when you compile cyclic_ll.c will tell you that has_cycle agrees with what the program expected it to output.

What to hand in

When you are done with this lab assignment, you are ready to submit your work. Make sure you have done the following ***before*** you press Submit:

- Answers for each line of mem.c using **Q1-Q5**.
- Attach filled in cyclic_ll.c
- List of collaborators