EE205: Data structures and Algorithms for Electrical Engineering

Project 2: Receiving messages from network packets

Project 2a: Linked lists implementation

1. Singly linked list:

- Since there is only one pointer “next->”, it is pretty time-consuming to traverse the list from head to the element right next to our “elem” passed as argument.

- **NOTE:** please fill in all of the “throw an exception” which handle the abnormal cases (how to handle depends on your choice ).

- Since HEAD is just an element of the list (HEAD = FRONT), when the list is empty, HEAD is NULL

2. Doubly linked list:

- I suppose there is no need to check if "before" actually belongs to the list?

- **NOTE:** please fill in all of the “throw an exception” which handle the abnormal cases (how to handle depends on your choice ).

Update: 10/07 10:23pm

- Done with skeleton code, please compile and check accuracy tomorrow (the most painful process T\_\_T)

- There are some comments, please pay some attention and do some necessary modifications.

- Don’t forget to collect the time and describe the time difference between two receives (if possible) in the README file

NOTE: 2017/10/08 : change all of the code

->next to the list\_get\_next

->head to the list\_get\_head

Because “head” and “next” are private and must be accessed via the interfaces provided only T\_\_T

2 files: single\_receiver.cpp, double\_receiver.cpp

2017/10/08 ~ 4:06 pm, project 2a checked (need to check for warning in lab machines)

GUILDELINES:

- First, test in local machine: test via Visual Studio Code, any mistakes must be corrected in the DEV files (in file Folder of Windows) first, then copy paste to Visual Studio Code (for coherence of the progress).

10/08: 4:52 pm SEGMENTATION FAULT in receiver!!! FIX please!!!

10/09: 3am Part 2 of single\_receiver is SO WEIRD! Fix please!!!!

10/09: 2pm remaining tasks to do

**+ check double\_receiver**

**+ collect time**

+ print to output file

+ submit

NOTE: In the Readme file, please describe briefly how you design your message receiver