COMP1511 PROGRAMMING FUNDAMENTALS

LECTURE 18

Revision: Linked Lists

LAST TIME...

- The Exam
- Quick revision of pointers
- Example Q1

TODAY...

- Linked Lists - some example problems revision

“

WHERE IS THE CODE?

Live lecture code can be found here:

HTTPS://CGI.CSE.UNSW.EDU.AU/~CS1511/22T1/LIVE/WEEK10/

COURSE FEEDBACK

my Experience surveys

http://myexperience.unsw.edu.au/

REVISION CLASSES

PLEASE BOOK NOW!

Come along and work on revision problems with the support of our lovely tutors:

- ONLINE:
  - Tuesday 26th April 3-5pm (Bridget and Enzo)
- FACE TO FACE in Brass Lab:
  - Wednesday 27th April 11-1pm (Ben and Michelle)
  - Thursday 28th April 3-5pm (Gab and Riley)

Book using Hale: https://unsw.to/hale

LINKED LISTS

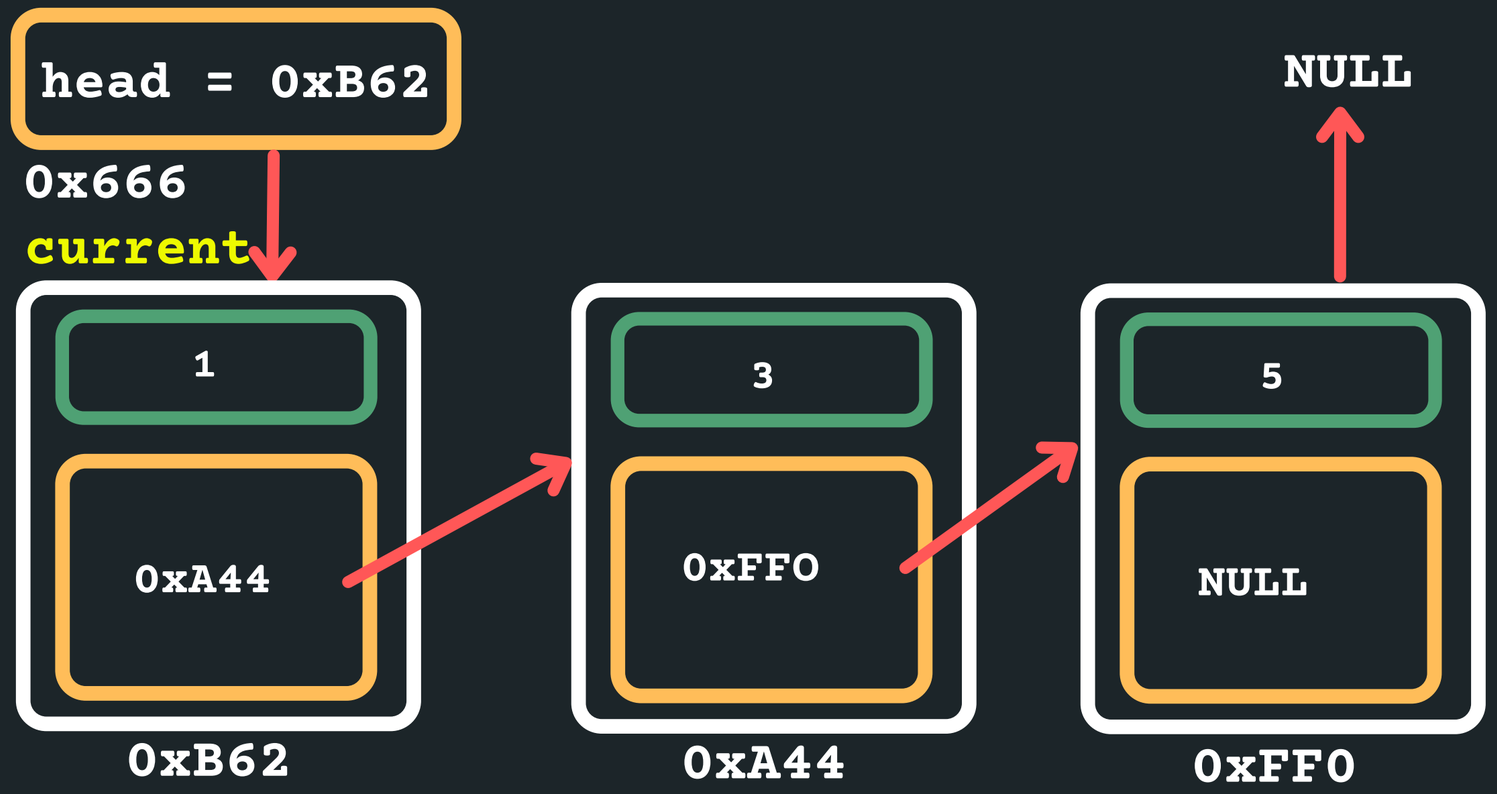
REVISION

- Can only access things sequentially by traversing the whole list
- Can add nodes in as needed (dynamic memory allocation) - by using malloc(sizeof(struct node))
- Can delete nodes as needed (by using free()
- Can check for memory leaks (has everything been freed?) by using: dcc --leakcheck

RECALL A LINKED LIST STRUCTURE

head is just a pointer (not a node!) that holds the address of the first node

End of the list reached when you hit NULL



LINKED LISTS

REVISION

- Some special boundary conditions that you need to consider when you manipulate lists:
  - Empty list
  - List with 1 element
  - Something happening at the beginning of the list
  - Something happening at the end of the list
  - Something will not occur, the item is not in the list (inserting after a number that doesn't exist etc)

EXAMPLE PROBLEM

Problem 1: Find the range (the difference between the biggest term and the smallest term) of a linked list

EXAMPLE PROBLEM

Problem 2: Insert a specified number into the middle of a linked list. Assume that there is always going to be an even number of numbers in the list before insertion

EXAMPLE PROBLEM

Problem 3: Concatenate two linked lists (join one linked list to another)

BREAK TIME...

Did you enjoy your first taste of programming?

EXAMPLE PROBLEM

Problem 4: Count all the elements in the linked list that are divisible by a specified number

EXAMPLE PROBLEM

Problem 5: Now delete all the nodes that are divisible by the specified number

Thank you all so much for tuning in, for learning, for engaging, and I hope that you had an enjoyable intro to programming. Don't forget that Rome wasn't built in a day, and becoming a better programmer entails lots of practice!

I really appreciate the engagement that you have shown throughout the lectures, and I wish you all well in the final exam.

Have a wonderful *short* break, I hope you all get some proper down time.

Good Luck in the exam and for your future courses, and I may see some of you again in your later courses :)

WHAT DID WE LEARN TODAY?

REVISION:
LINKED LISTS

linked_list.c

main.c

linked_list.h

REACH OUT

CONTENT RELATED QUESTIONS

Check out the forum

ADMIN QUESTIONS

cs1511@cse.unsw.edu.au