COMP1511 PROGRAMMING FUNDAMENTALS

LECTURE 18

Revision: Linked Lists

the

LAST TIME...

- Quick revision of pointers, strings and arrays of struct
- 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/~CS1511/22T3/LIVE/WEEK10/

“

COURSE FEEDBACK

Tell us about your experience and shape the future of education at UNSW.

Click the link in Moodle

Please be mindful of the UNSW Student Code of Conduct as you provide feedback. At UNSW we aim to provide a respectful community and ask you to be careful to avoid any language that is sexist, racist or likely to be hurtful. You should feel confident that you can provide both positive and negative feedback but please be considerate in how you communicate.

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:
    - Thursday 24th November 2-4pm
- FACE TO FACE in Sitar/Kora labs J17:
    - Friday 25th November 2-4pm

Register: https://www.eventbrite.com.au/e/week-11-revision-classes-tickets-467976057987

LINKED LISTS

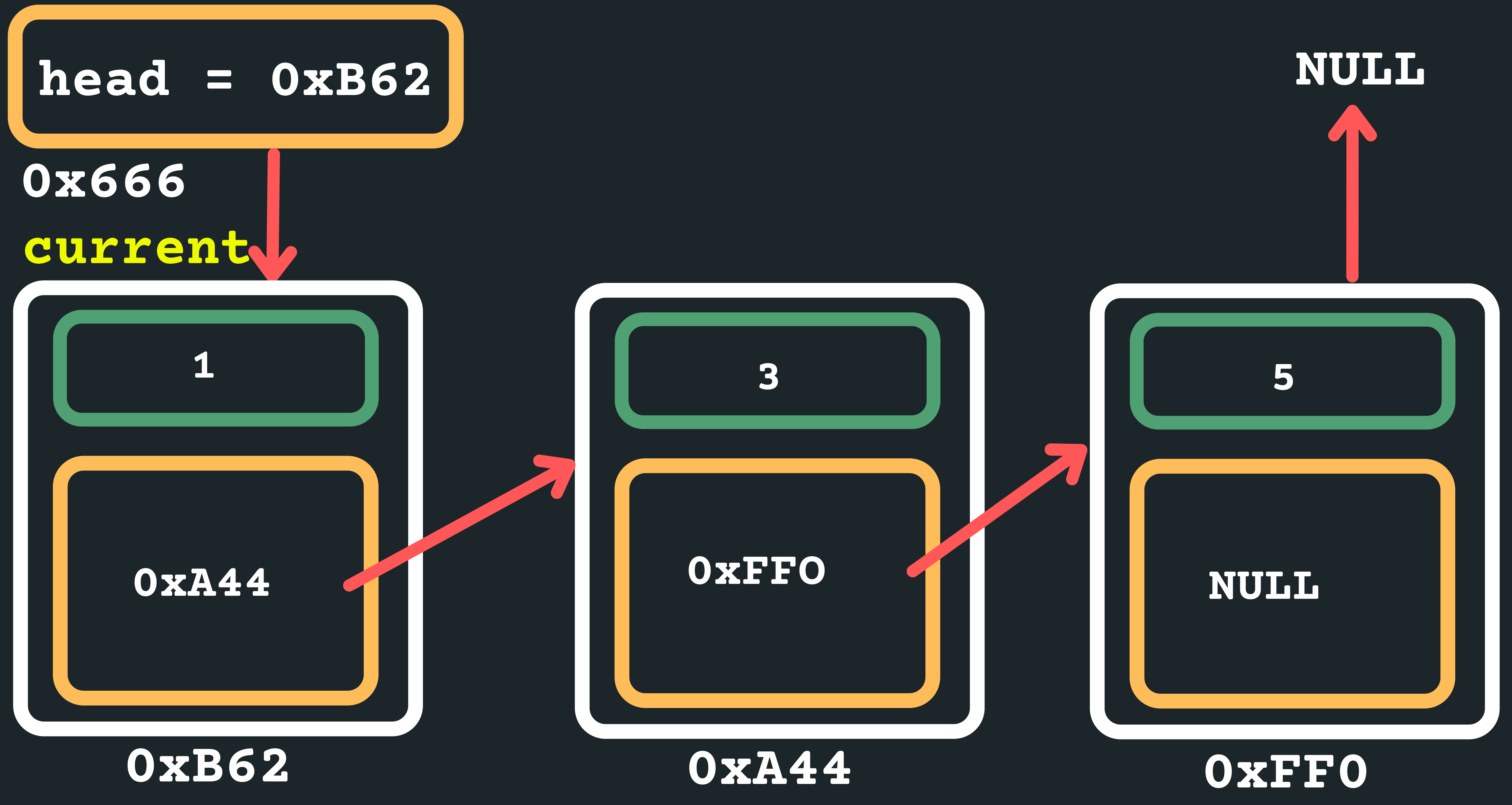
REVISION

- 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

(see the working files for the details spec)

EXAMPLE PROBLEM

●○○

Problem 2: Concatenate two linked lists (join one linked list to another)

EXAMPLE PROBLEM

Problem 3: Given two linked lists, return the difference in the number of items in the two lists.

EXAMPLE PROBLEM

Problem 4: 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.

BREAK TIME...

Did you enjoy your first taste of programming?

EXAMPLE PROBLEM

Problem 5: Count all the elements in the linked list that are divisible by 6 and output the count.

EXAMPLE PROBLEM

Problem 6: Now delete the first node in the list that is divisible by 6

THANK YOU

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

problem1.c

problem2.c

problem3.c

problem4.c

problem5.c

problem6.c

REACH OUT

CONTENT RELATED QUESTIONS

Check out the forum

ADMIN QUESTIONS

cs1511@unsw.edu.au