CS180 - Linked Lists

Note Title 2/7/2011

Announcements

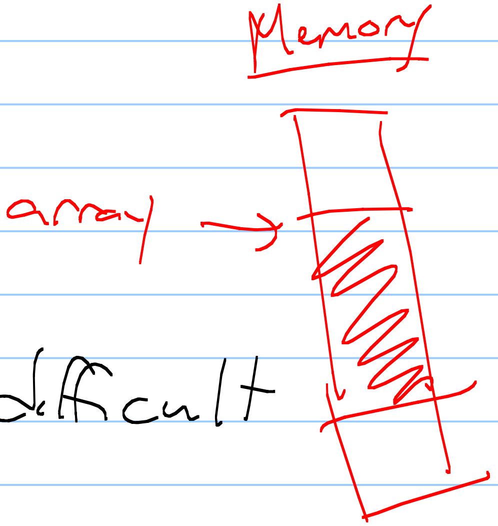
- Lab will be on Wednesday this week!
- HW due Tuesday by midnight
- Ch. 3 photocopies are at front of room
- Textbook is now available!

Recap of Arrays (Ch 3.1 of text)

Limits

not at all flexible

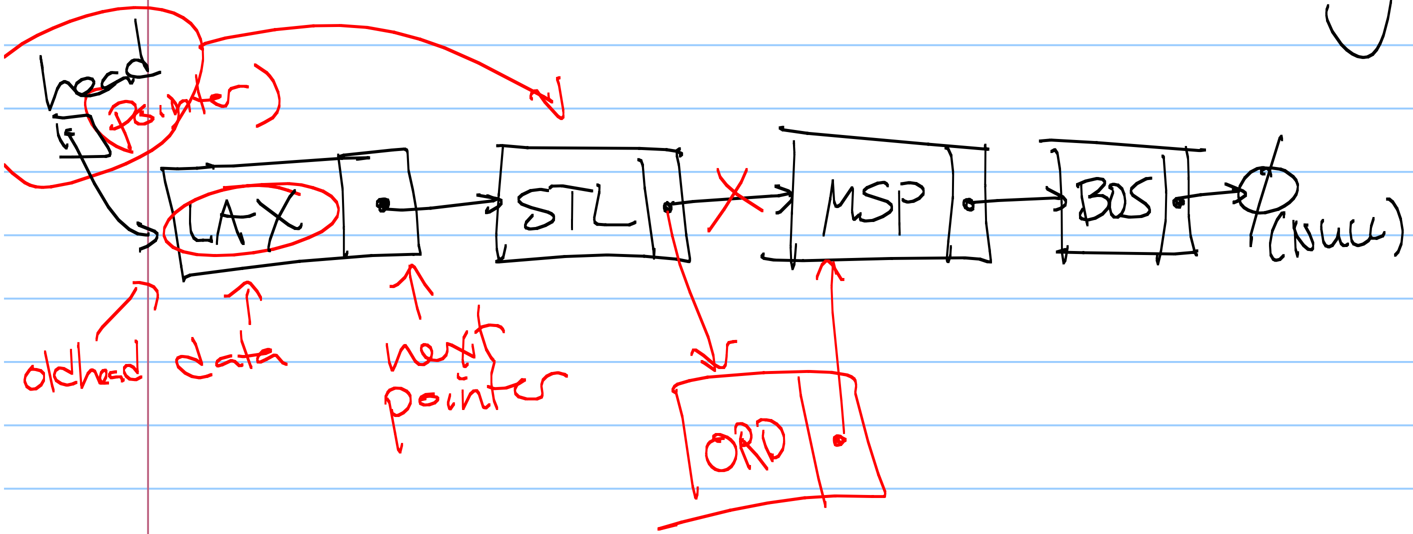
- size is fixed
- inserting + moving is difficult

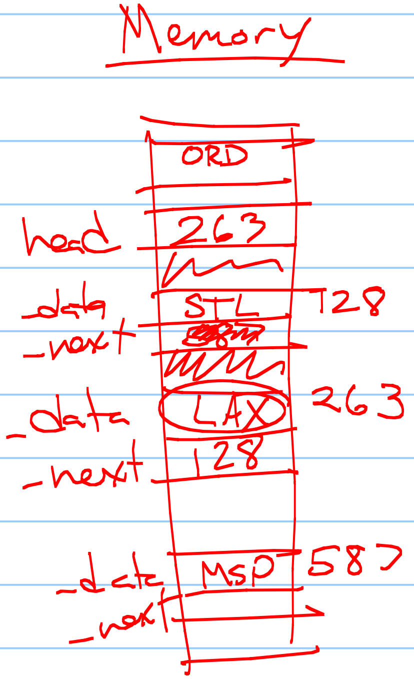


- 1 data type
- acces is unrestricted

Singly Linked Lists

A collection of nodes that together form a linear ordering.





```
head = head->next->next;
```

Functions (very stripped down version)

What might we want to do?

insert front

delete front

→ front - return a reference to _data

Constructor

destructor

empty

Implementation - Nodes

We want a node to store two things

- data
- pointer to next

template

Code: Need our node class

E (in textbook)

```
template <typename Object>
class SLinkedList {

    private:

        class SNode {
          private:
            Object _elem;
            SNode<Object>* _next;
        };
```

Implementation — List

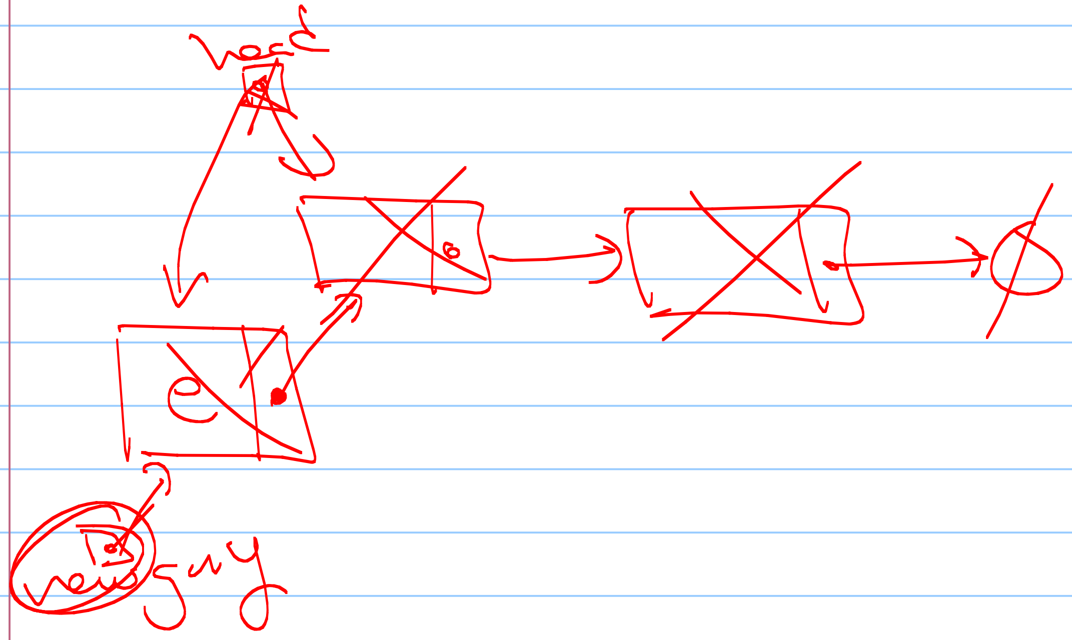
What private data do we need?

in private section:

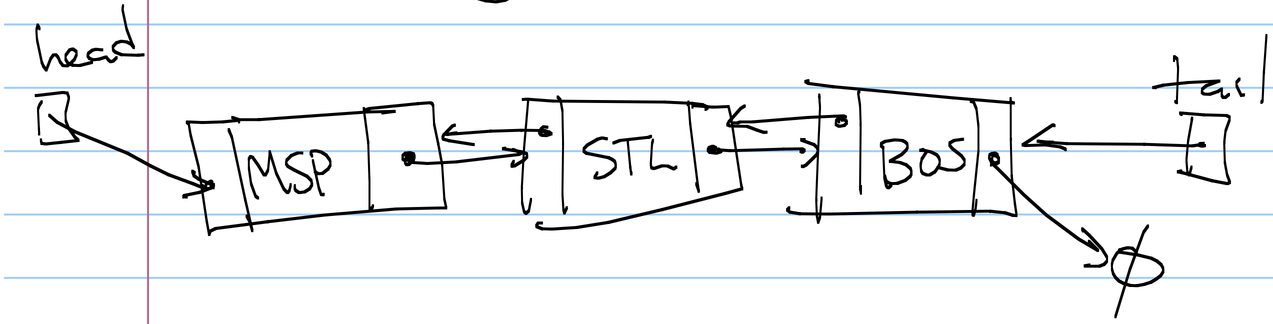
```
SNode<Object> * _head;
```

Code

(switch to SLinkedList.h
and SLinkedList.cpp)



Doubly Linked List



Implementation

What changes from SLinked List?