

## Static Linked Lists

2-22-2002

---

---

---

---

---

---

---

---

## Opening Discussion

- What did we talk about last class? Do you have questions about the assignment that is due today?
- Minute essay comments
  - A core dump is an image of the memory the program was using when it crashed.
  - I don't know of any language that does inheritance quite like C++.

---

---

---

---

---

---

---

---

## Dynamic Memory (when to delete)

- If you have a linked list that keeps pointers to the objects you don't want to delete those objects until you are through with them and they are out of the list.
- Thinking back to the minute essay from last class you should picture in your head the difference in memory between a list of objects and a list of pointers to objects. You want the later so single objects persist through the entire execution.

---

---

---

---

---

---

---

---

## What is a Pointer?

- You all know the answer to this, it is an address in memory. All that really is is an integer.
- In some ways you can use integers like pointers for keeping track of where a given object is in an array. The real difference in doing this is that a real pointer has type information. It points to a specific type. An index for an "address" into an array does not.

---

---

---

---

---

---

---

---

## Linked Lists without Pointers?

- You can create entities called static linked lists by using index integers like pointers.
- A static linked list is implemented with an array of elements. For each element we keep a next integer instead of a next pointer. You walk this type of list with an integer as well.
- ```
for(int rover=0; rover>-1;
    rover=array[rover].next)
```

---

---

---

---

---

---

---

---

## Advantages

- So why would you ever want to do this? It turns out there are two compelling reasons.
- Writing to disk
  - Writing a real pointer to disk is pointless and doesn't help you figure out in the file where the "next" element is.
- Speed
  - Memory allocation is slow. If you know how many elements you need this is faster.

---

---

---

---

---

---

---

---

## Drawbacks

- Of course, there are problems as well. Otherwise we would only cover static linked lists.
- Type information
  - The integers don't carry any type information so the compiler can't find as many bugs for you.
- Flexibility
  - You typically lose inclusion polymorphism (unless it is an array of pointers). It is also much harder to get more nodes if you need them.

---

---

---

---

---

---

---

---

## Code

- Let's write a little code to handle a static singly linked list. One thing that you will notice is that we have to do our own "memory management". We have to keep track of which elements are being used with a "free list".

---

---

---

---

---

---

---

---

## Minute Essay

- Could you use a static linked list for your assignment? What restrictions would it impose or what things would you have to do to get around those restrictions?
- Remember that assignment #2 is due today. I intend to play basketball until about 2:00. I have a Java session from 2:30-3:30. Then an algorithms meeting starting at 4:00.

---

---

---

---

---

---

---

---