Linked List

Yung-Hsiang Lu

Memory Management

How much memory is needed? When is this known?

1. know when a program is written: allocate fixed amount, for example

```
int a;
int b[100];       /* b can exactly100 elements */
```

2. know sometime during execution and allocate **only once**

```
int * c;
…
c = malloc(numElement * sizeof(int));
```

   allocate as more memory is needed, released (by calling free) when no longer needed

3. allocate more as needed but do not release existing memory

   $\Rightarrow$ called **dynamic structure**

Why Dynamic Structures?

- In many (actually most) applications, data are added and removed frequently.
- What's wrong with method 2?
  - allocate
  - allocate more for new data
  - copy old data
  - release old memory

  ⇒ allocating memory + copying contents + releasing memory take too much time, especially when ptr1 already has many elements.
- How do dynamic structures work?

  ⇒ Each element has a **pointer** to the next element.

```
ptr1 = malloc(size1);
if (size1 is too small)
{
    ptr2 = malloc(size2);
    copy contents from ptr1 to ptr2;
    free(ptr1);
}
```

Linked List