

Lecture 23

Memory control

CS 241: Foundations of Sequential Programs
Fall 2014

Troy Vasiga et al
University of Waterloo

A few words on heap management

- ▶ WLP4, C, C++ have explicit memory management
- ▶ delete is a promise to not use memory again
- ▶ never trust a programmer:

```
int wain(int *a, int n) {  
    int* b = NULL;  
    b = new int[3];  
    *b = 7;  
    *(b+1) = 17;  
    *(b+2) = 345;  
    delete [] b;  
    println (*(b+1));  
    return 0;  
}
```



C++ does
not dislike
this

- ▶ Java doesn't trust programmers: it does implicit memory management. Programmers cannot directly access memory locations in Java.
- ▶ More to discuss over next several classes

NEW and DELETE

Recall the two CFG changes involving arrays:

factor → NEW INT LBRACK expr RBRACK

```
code(factor) = code(expr)
              add $1, $3, $0
              new($1)
```

statement → DELETE LBRACK RBRACK expr SEMI

⋮
delete(\$1)

Additionally, we would have to add something in the prologue code:

```
.import init
.import new
.import delete
```


Four dimensions of heap allocation

1. fixed vs. variable size blocks
2. implicit vs. explicit allocation
3. implicit vs. explicit reclamation
4. language/implementation
 - ▶ Can pointers be relocated?

An easy case

Fixed size, explicit allocation, explicit reclamation, no pointer reallocation

- ▶ Initialize

- ▶ Allocate

- ▶ Reclaim

Being slightly more clever

Reclaim:

Back to allocate, with this change to reclamation:

If free list \neq empty

Another way to implement this

Costs:

- ▶ allocation
- ▶ reclamation
- ▶ failure
- ▶ waste

Variable size blocks

Life only gets more difficult.

A picture:

Issues:

▶ Allocate:

▶ Reclaim:

Allocation strategies

- ▶ First-fit:

- ▶ Best-fit:

- ▶ Worst-fit

Deallocation

Extra information required to help with merging

Theorem: If area is twice as big as the amount to be allocated, the holes are on average twice as big as the allocated blocks.

Four cases of delete

Knuth 50% rule: At equilibrium,

Deferred reclamation

Using a dual-core processor

An even better idea

Comparison of copy/compaction

Implicit reclamation

Sometimes called *garbage collection*.

- ▶ Traditionally:

- ▶ Contemporarily:

Issues opened up

A few final words about C/C++

A few final words about CS241