

Announcements

- Midterm #2 was returned
- Program #4 on the web
- Reading
 - Notes (Today)
 - Notes (Next Class)

How to make heaps safer (from lect 18, but got left out)

- aux tag storage
 - create additional (redundant) info elsewhere in heap
- preamble/trailer with known patterns
 - allocate $2 * Y$ bytes extra
 - fill out first and last Y bytes with a known pattern (non-zero)
 - return ptr Y bytes into region
 - on free (or when user requests)
 - verify y bytes at start and end haven't changed
- On Linux,
 - `mcheck_pedantic(NULL)`
 - turns on heap checking.

Project #4 Discussion

- Really three parts to project:
 - implement generic table abstraction (table.c)
 - refactor intCount.c
 - intCount.c - changed to call table.c abstractions (for avl)
 - libavl.so - implementation of avl tree
 - implementation of chained hashTable
- Notes:
 - keyword and value in table are void*
 - can't assume they point to anything
 - someone could store ints there (i.e. intCount)

Garbage Collection

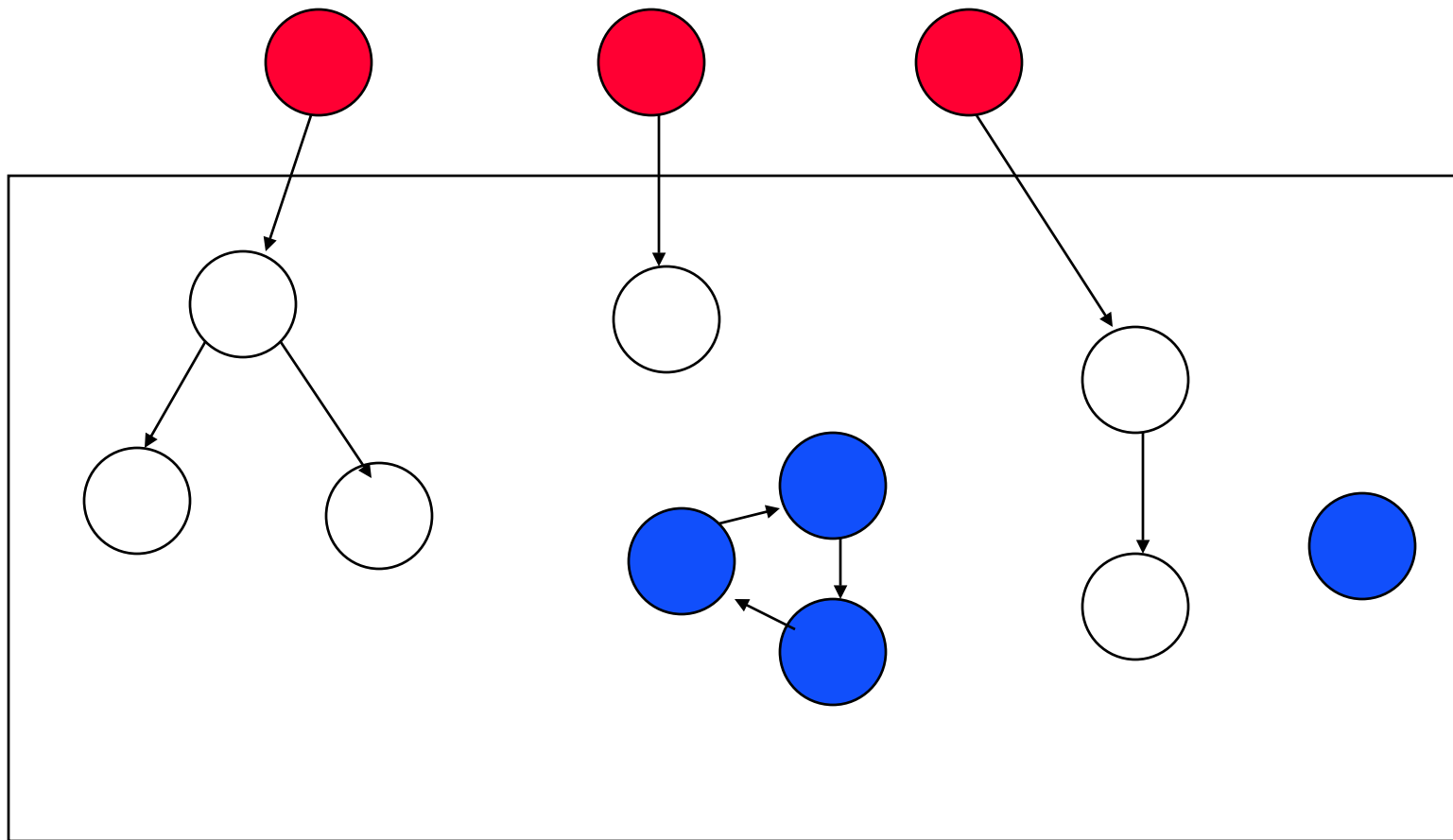
- What is Garbage?

- memory the program can't get to anymore
- Example:
 - `ptr = malloc(40);`
 - `ptr = NULL;`

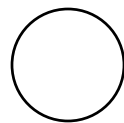
- How to detect Garbage?

- Need to know what is a pointer
 - know what everything is (Java)
 - guess anything that looks like a pointer is one (C)
 - any 32-bit quantity in globals, on stack, is assumed to be a pointer.
 - anything in a memory region pointed to
- Compute reachability graph
 - Follow all pointers

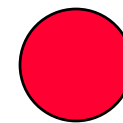
Reachability Graph



Not-reachable
(garbage)



Reachable



variables
(Inside the program)

Mark and Sweep Garbage Collection

- Mark Function

- each allocated block has a "marked" field
- isPtr returns header for ptr (if in heap) or null

```
void mark(ptr p) {  
    header * b;  
    b = isPtr(p);  
    if (!b) return;  
    if (b->marked) return;  
    b->marked = 1;  
    for (i=0; i < b->size; i++) {  
        mark(b + i + 1);    /* assumes heap header is 32 bits */  
    }  
}
```

Mark & Sweep

- Sweep

```
void sweep(header *curr) {  
    while (curr->size) {  
        if (curr->marked) {  
            curr->marked = 0;  
        } else if (curr->allocated) {  
            free(curr+1);  
        }  
        curr = curr->next;  
    }  
}
```

- Calling Mark and Sweep

- Call when the allocate function is about to call sbrk

Garbage Collection

- Writing isPtr
 - need to be able to find a block header given a ptr into it
- Conservative Garbage Collector
 - something may look like a pointer, but not be one

Midterm #2

- Overall results were very good
 - Mean: 79.6
 - StdDev: 9.1
 - Min: 57
 - Max: 94
- Re-grades
 - submit requests via web site
 - turn in exam booklet to me
 - do not mark on exam booklet if you want a re-grade