Garbage Collection

CS 520
Dept. of Computer Science
Univ. of New Hampshire
Garbage Collector

works with a memory allocator
identifies allocated blocks that are garbage

— okay to be re-claimed and re-allocated
memory allocator

mem Initialize ( heapSize )

mem Allocate ( blockSize, Finalizer )
use malloc to allocate the heap
block headers

Stores control information

1. is block allocated?
2. block length
3. if allocated, finalizer address
4. if allocated, is block reachable?

Use single bits for 1 and 4 and pack with the block length.
Finalizer address is its own word.

[Diagram of block structure]

* Does length include the header?
memInitialize

memInitialize (1000)

malloc return value

heap Begin

heap length
men Allocate (100, 0x80483c4)
aligning values on appropriate memory boundaries is important for correctness on some machines and for performance on all machines. A memory system is designed to quickly read aligned values.

Typically:

- A 4-byte value (int/float) should be on a 4-byte boundary.
  - Address evenly divisible by 4

- An 8-byte value (double) should be on an 8-byte boundary.
So allocator allocating In units of bytes would round-up user's requested length to be a multiple of 8. This, of course, ensures that header words will be aligned.
memFree?

Could require user to call free to explicitly deallocate a block

or when heap is full, run a garbage collector
garbage collection

identify allocated blocks that program can no longer reach

--- no pointers exist to it

mark and sweep!
Mark and sweep

1. Mark all reachable, allocated blocks

2. Sweep over the heap and re-claim the unmarked, allocated blocks
Re-claiming blocks

- Call the Finalizer function
- Mark the block as free
- Combine it with any adjacent free blocks
Which allocated blocks are reachable:

- pointed to by global variable
- or
- pointed to by the stack
  - i.e., local variable
  - parameter
  - temporary
  - saved register
- or
- pointed to by a reachable block
  - i.e., marking procedure is recursive
    - base of recursion: block is already marked
- or
- pointed to by a register
  - only need to worry about callee-saved ones
under Linux, linker inserts symbols that delimit the static data area

```c
extern int __data_start;
extern int __end; // just one underscore here!
```

Take the address of these to set the bounding addresses

---

Note: be careful about the heap's control variables
Stack

on Intel follow the %ebp chains

begin here

end here
Pointers inside reachable blocks

Consider these user data structures

 Globals

 Heap

 Stock

 What is garbage?

 Note: need to remember to clear mark bits
Java has run-time type info so it knows exactly where the pointers are.

What if you do not have run-time type info?

must assume that bits that look like a pointer into the heap are a pointer into the heap

i.e. the bits interpreted as a pointer might actually be an int or float.
This is conservative.

It might mark a block that does need to be marked.

OK to overmark — keep utilization issue

Not OK to undermark — correctness issue
advanced topics

concurrent garbage collection
incremental garbage collection
generational garbage collection

Resource: the Garbage Collection Page maintained by Richard Jones