9/25/12
Cs61 Scribe Notes
Gabriel Trevino
Norman Zhu

Lots of code being looked at today in lecture available at:

SAES Code repository cs61 -> cs61-lectures

When entering code into Piazza, be sure to hit preformatted button to retain indentation and nice formatting.

Static (globals)

Advantages of these:

- Simplest way to allocate storage of variables
- Managed by compiler
- Cheap
- No memory errors since they live only as long as program runs. (Lifetime of a global is program execution).

Disadvantages:

- Fixed size determined at compile time, not dependent on input

Automatic (locals / stack)

Advantages:

- Simple
- Managed by compiler
- Cheap

Slightly better but still kind of bad:

- # of stack frames dependent on input

Disadvantages

- Any individual frame has a fixed size
- Lifetime is bounded by the function. Once a function returns its variables, they disappear. Makes it difficult to return values from function to another.

Dynamic (heap)

Advantage:

- Super flexible! Lifetime is determined by [your] code
- Size can be totally input dependent.

Disadvantage:

- COMPLICATED
- Memory errors common : the way in which they are accessed (pointers) are fussy
- Nameless objects that require pointers

Dynamic variables: so powerful that they are used by all modern languages (including C) all the time.

They attempt to compensate for disadvantages by introducing implicit memory management.

Explicit dynamic storage duration: free must be called

Implicit dynamic storage duration: Garbage collection!
Advantages of garbage collection:

- Language calls free for you, so no memory leaks possible.
- No double frees

How does garbage collection work?

0. Track every dynamic object
1. Final all accessible objects (This is the tough part).
2. Free all inaccessible objects.

Accessible objects are those which have pointers to them.
Let A be all allocated active objects
Let R be a set of roots .
Find some A, where R is a subset of A', where A' is a subset of A,
Such that A' is closed under pointer following.

Dictionary code example : cs61-lectures/l07v

Takeaway Questions:
Which dictionary contains each word?
How many words are within a dictionary, given the number of allocations?
Code which accurately tracks allocations and tells us how many of them there are.