talloc assignment (part 2)

Compilers / interpreters

(Note: class next Wed on Zoom)

We're building a linked list

malloc

free

Part 2: build a very rudimentary garbage collector so don't need to worry abut free

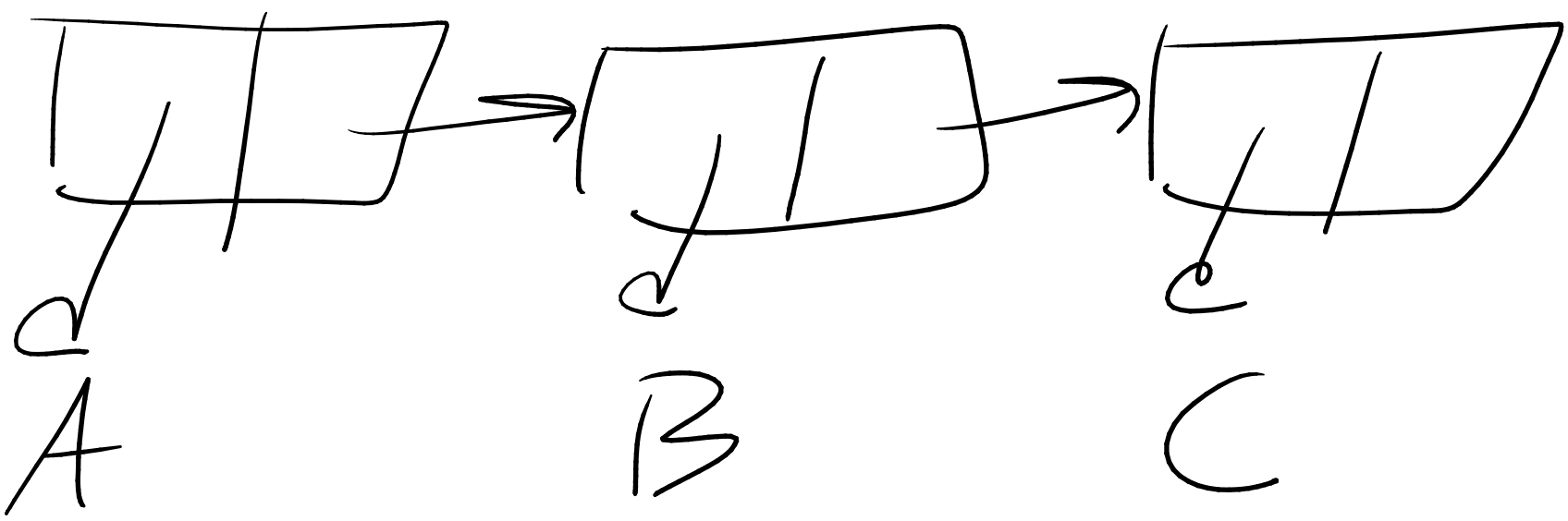
---

```
void *talloc(int size) {
    ... malloc memory ...
        remember it
        return a pointer to it
}
```

talloc(3) A

talloc(sizeof(int)) B

talloc(20) C




```
Void tfree() {
    loop ove list
      free memory
  } free list itself
```

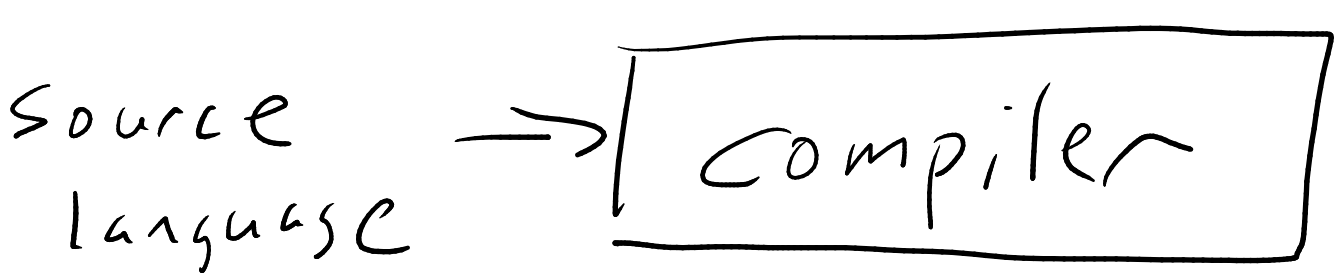
After done, in future

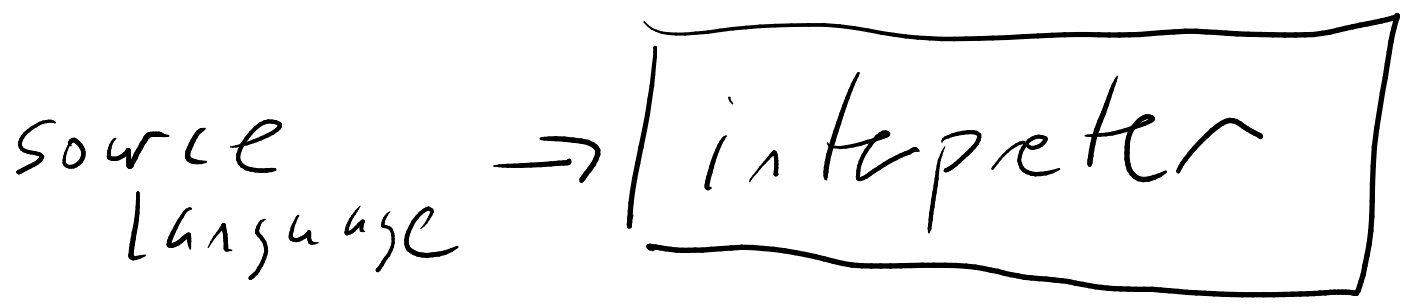
- talloc instead of malloc
- one tfree at end of main

Compiler - translate a program from one language to another

Interpreter - directly executes a program, "pretending to understand it"

We're going to build one of each right now!





Source language: Greet

Great program:

hello 3
bye 2
hello 1

Output:

hello
hello
hello
bye
bye
hello

Target language: bash

Implementation language
- the language we will write the compiler in: Python

Myth to ~~demystify~~ bust.

False fact some people believe:

"An interpeter translates a program line-by-line, executing as it goes."

This is not generally true, and is generally unhelpful.

Bust it:

- what is a line of Scheme?
- modify Greet. New command: suppress. If it appears at end of program, don't even print anything
- Python print(len(lst))

Maaser - keep everyone participating
Scribe - writes answers for group
Reader - reads out loud for group
Presenter - share w/ rest of class