Separate Compilation

Bryce Boe

2013/10/09

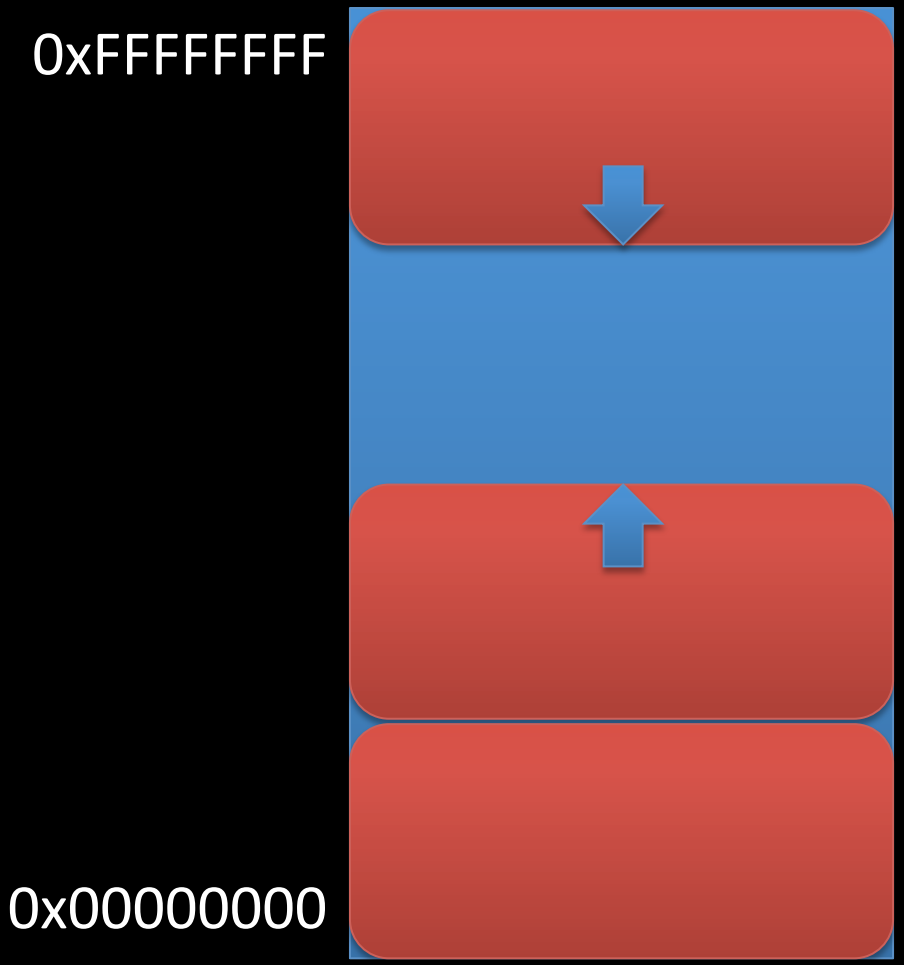
CS24, Fall 2013

Outline

- Monday Review + overflow example
- Lab 2 Solution
- Libraries and Separate Compilation

MONDAY REVIEW

What are the segments called, and what do they contain?



Memory Allocation Check

```
int *array_copy(int *values, int n) {
        int *copy = malloc(sizeof(int) * n);
        for (int i = 0; i < n; ++i)
                copy[i] = values[i];
        double pointless[8];
        return copy;
}
```

- What is the size of foo's simplified activation record?
- How much memory is allocated from the heap?

LIBRARIES AND SEPARATE COMPILATION

What?

- A **library** (also referred to as modules) is a collection of structures and functions that perform some function
  - stdio: Provides the FILE struct and input and output routines
  - list (project 1): Provides a List struct and associated operations

Example

- <In class example using the following files:>
  - pre_library.c
  - library_usage.c
  - cs24lib.c and cs24lib.h
  - cs24lib_ext.c and cs24lib_ext.h

Notes from the example

- In order to re-use functions they need to be in their own files
- Use MACRO conditionals (*#include guard*) to prevent #including the same *code* more than once
- Separate structure definitions and function declarations into .h files to support *separate compilation*

Library Components: Header File (.h)

- Provides the *interface* for the module
- Defines data structures (e.g., FILE, List, Node)
- Declares function prototypes
  - int get_at(struct List *list, int index);
- Uses macros (#define, #ifndef, #endif) to prevent duplicate declarations

Library Components: Implementation File (.c)

- Provides the *implementation* for the module
- Uses the #include macro to include the associated header
- Provides the function definition (i.e., the completed source code)

Questions

- Why should you never #include a ".c" file?
  - Doing so doesn't allow for *separate compilation*
- What is the purpose of the "#ifndef … #define … #endif" guard around the content of ".h" files?
  - Avoids structures and functions from being declared more than once

Another Question

- What is the primary purpose of separate compilation?
  - To reduce subsequent compilation time by reusing *object* files

For Next Monday

- Finish reading chapter 1 in the text book (if you haven't already)
- Begin reading chapter 3 (might want skim/ read chapter 2) as it's helpful for project 1
  - Note the book uses C++ so (for now) think about how to do similar in C