

Project 1 Overview (List ADT), Array List

Bryce Boe

2013/10/14

CS24, Fall 2013

Outline

- Separate Compilation Quick Review
- Project 1 Overview (List ADT)
- Array List implementation
 - Realloc

SEPARATE COMPILATION REVIEW

Questions

- Why do we separate code into '.h' and '.c' files?
 - Separates declarations from definitions
 - Allows separate compilation
- What is the primary purpose of separate compilation?
 - To reduce subsequent compilation time by reusing *object* files

PROJECT 1: LIST ADT

What is an abstract data type?

- Some structure (class in C++) and associated functions
- Container ADTs allow you to store data and perform certain tests

Abstract Data Types

- A container for data
 - Container provides a set of operations
 - Essentially: insert, delete, contains, and iteration
 - Abstract in that the *customer* does not need to concern themselves with the implementation

Project 1 Purpose

- Implement the List ADT using two distinct storage models
- Understand the tradeoffs between the two implementations

List Operations

- Consult the project 1 description for the list of methods to implement

Simple Container Implementation

- See `simple.c`

PROJECT 1 ARRAY LIST

How do you resize the array?

- Allocated size should remain a power of 2
- Double the size whenever needing to expand the list
- Don't worry about shrinking the size of the list (though you can if you want)
- See example (realloc.c)

Array-implementation walk through

- `struct List* list_construct()`
- `void list_destruct(struct List *list)`
- `int list_size(struct List *list)`
- `int list_is_empty(struct List *list)`
- `char *list_at(struct List *list, int position)`
- `int *list_push_back(struct List *list, char *ite)`
- `char *list_remove_at(struct List *list, int pos)`

For Wednesday

- Continue 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
- Suggested:
 - Complete the array list implementation
 - Start the linked list implementation