Chapter 13 - Dynamic Data Structures

At a Glance

Lesson Contents

- Overview
- Objectives
- Quick Quizzes
- Additional Resources
- Key Terms

Chapter Notes

Overview

Chapter 13 provides an introduction to linked lists and dynamic memory allocation in C. You learn how to use and create linked lists, stacks and queues. You also learn how to create dynamically linked lists. Finally, you learn about some common programming and compiler errors, and how to avoid them.

Objectives

- Introduction to linked lists
- Dynamic memory allocation
- Stacks
- Queues
- Dynamically linked lists
- Common programming and compiler errors

Introduction to Linked Lists

Topic Tip	You may want to use an animation to help visualize how a linked list works. For example, see **www.cs.stir.ac.uk/~mew/dissertation/simulation.htm**.

Topic Tip	Variants of linked lists include doubly-linked lists and circularly-linked lists. For more information, see **http://en.wikipedia.org/wiki/Linked_list**.

Quick Quiz 1

1. What is a linked list?

2. What is a self-referencing structure?

3. All programming languages that support pointers provide a special pointer value, known as both NULL and____________________, which acts as a sentinel or flag to indicate when the last structure has been processed.

4. The expression `t1.nextaddr->name` can, of course, be replaced by the equivalent expression____________________, which explicitly uses the indirection operator.

Dynamic Memory Allocation

Topic Tip	Make sure you read and understand why it is very important to check return values when making `malloc()` and `realloc()` function calls (see the Programming Note on page 616).

Stacks

Topic Tip	You may use an animation to helps visualize how a stack works. For example, see **www.cs.usask.ca/resources/tutorials/csconcepts/1998_5/stacks/java/** or **www.cs.hope.edu/~alganim/jvall/applet/stack.html**.

Topic Tip	Reverse Polish Notation (RPN) (postfix algebra) can be easily implemented using stacks. For more information, read the Historical Note on page 611.

Quick Quiz 2

1. What functions are available in C for the dynamic allocation and release of memory space?

2. How does `malloc()` work?

3. A(n)_____________________is a special type of linked list in which objects can only be added to and removed from the top of the list.

4. The operation of placing a new structure on the top of a stack is called a PUSH, and removing a structure from a stack is called a(n) _____________________.

Queues

Topic Tip	Stacks and queues are two special forms of a more general data object called a deque (pronounced "deck"). The term "deque" stands for "double-ended queue." For more information, see the Historical Note on page 620.

Topic Tip	You may use an animation to help visualize how a stack works. For example, see **www.cs.odu.edu/~zeil/cs361/Demos/replays/queuelist.html**.

Quick Quiz 3

1. What is a queue?

2. What are the names of the operations used to add and remove items to/from a queue?

3. In a(n)_____________________, elements can be added and removed from anywhere within the list.

4. The operation of adding a new structure to a dynamically linked list is called a(n) _____________________.

Additional Resources

1. Linked List:
http://en.wikipedia.org/wiki/Linked_list

2. Stack:
http://en.wikipedia.org/wiki/Stack_%28data_structure%29

3. Queue:
http://en.wikipedia.org/wiki/Queue

4. Deque:

http://en.wikipedia.org/wiki/deque

Key Terms

- **Dynamic memory allocation动态内存分配** makes it unnecessary to reserve a fixed amount of memory for a scalar, array or structure variable in advance.
- Placing a new item on top of the queue is formally referred to as **enqueueing入队**.
- The **heap堆** consists of unallocated memory that can be allocated to a program as requested, while the program is executing.
- The field on which a list is ordered is referred to as the **key field关键字域**, and insertions and deletions are always made to preserve the ordering of this field.
- A **linked list链表** is a set of structures in which each structure contains at least one member whose value is the address of the next logically ordered structure in the list.
- Items are removed from a **queue队列** in the order in which they were entered.
- Dynamic memory allocation is also known as **run-time allocation运行时分配**.
- Structures that are "linked" together by including the address of the next structure in the structure immediately preceding it are known as **self-referencing structures自引用结构**.
- The operation of removing an item from a queue is formally referred to as **serving服务**.