

# CS 2303

# Systems Programming

Bob Kinicki  
C08



# Introduction



**Systems Programming**

# Introduction

- Survey and TA/SA Introductions
- Pause to Look Backwards and Forwards
- Course Objectives
- Course Operation/Expectations
- Course Plan and **Fluid** Syllabus
- Systems Concepts
- 'Old' Development Environment
  - C and C++
- Higher Level Language History

# Look Backwards and Forwards

- Computers
- WPI CS Curriculum
- Instructor
- Students
  - Expected Background
  - Going Forward
  - Your Future

# CS2303 Course Objectives

- To expose students to some of the lower level systems interface 'grunge' only clearly visible via C.
- To learn to program in C++ by learning to program in C first.
- To further develop the ability to design programs with emphasis on the abstract view of data structures.
- To get experience with the low-level implementation of data structures in C.

# CS2303 Course Objectives

- To learn the advantages of programming in an object-oriented language such as C++.
- To experience programming in the

Large

# CS2303 Course Objectives

**Pointers!!**

# Course Operation/Expectation

- The course web page is an important asset.
- Required Labs
- 5 Programming Assignments
- 2 Exams

# Course Plan and Fluid Syllabus

- To cover the details of C briskly.
  - Assume an understanding of iteration and conditional constructs.
  - To use only C I/O {grunge as promised!} at first
- To introduce data structures in C by doing at least one program with 'structs' and call by value.
- To finish up with as much C++ as possible.
- {Note - reading of the textbook will require jumping around during the C portion of the course.}

# Systems Concepts

- The goal in this programming course is to try to expose the students to places where the software and hardware meet or where the application interfaces with the operating system (OS).
- A systems viewpoint includes resource management (CPU and memory), process scheduling, concurrency and performance.
- {Note - this is too much for this instance of the course!}

# Systems Concepts

- The assignments include simulation and a little computer networking that introduce two system performance concerns - **efficiency** and **fairness**.
- The other important approach to appreciate is the computer scientist technique of using **abstraction** to insulate outer interfaces from 'under-the-hood' details (e.g., virtual memory and loaders).

# User Memory Protection

# Virtual Memory

# C Program Development Environment

## Standard Steps

1. *Edit*
2. *Preprocess*
3. *Compile*
4. *Link*
5. *Load*
6. *Execute*

Deitel & Deitel

© 2007 Pearson Ed -All rights reserved.

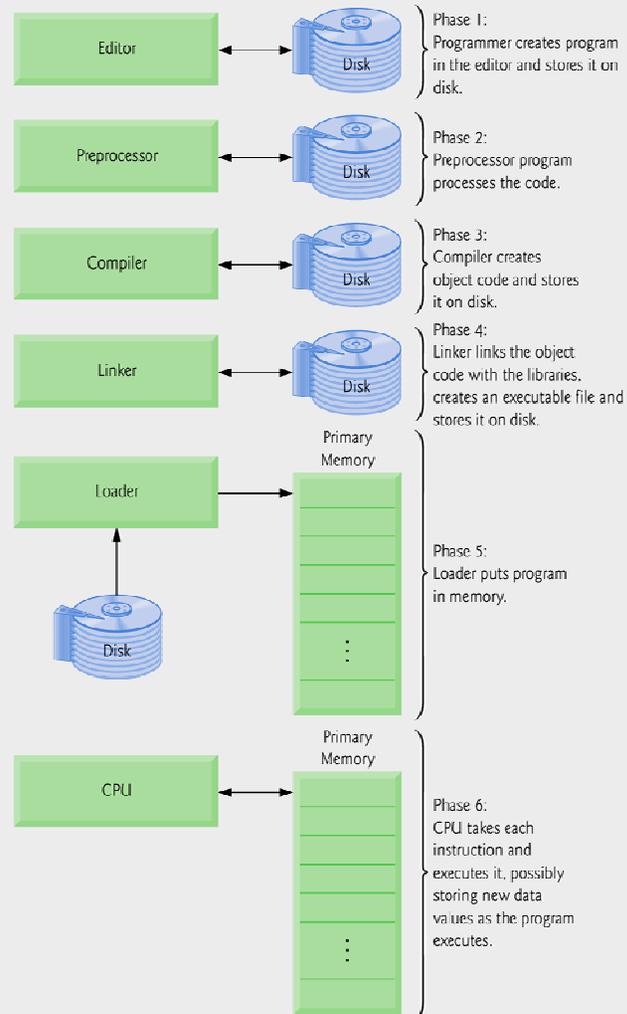


Fig. 1.1 | Typical C development environment.

# Higher Level Programming Languages History

Fortran <sub>w</sub>	1957	COBOL	1959	
Algol	1960	1968	Lisp	1959
PL1	1964	APL	1962	
Pascal <sub>w</sub>	1970	SNOBOL	1967	
C <sub>w</sub>	1972	Prolog	1972	
Basic	1975	Scheme <sub>w</sub>	1975	
C++ <sub>w</sub>	1986	ADA	1983	
Java <sub>w</sub>	1995			