CS 2303
Systems Programming Concepts

Bob Kinicki

A08

Introduction

Introduction

- Survey and TA/SA Introductions
- Pause to Look Backwards and Forwards
- Course Objectives
- Course Operation/Expectations
- Course Plan and 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.

* **Student is responsible for information on web page!**

- 5 Required Labs
- 5 Programming Assignments
- 2 Exams

Course Plan and 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 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*

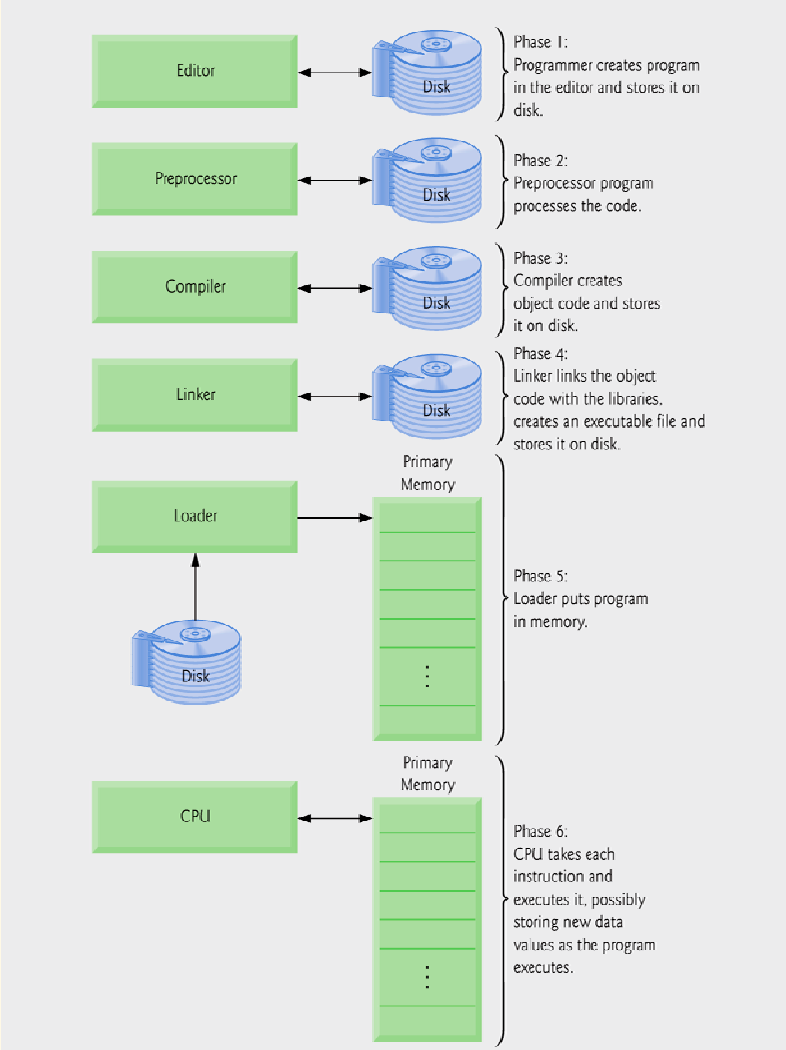

Fig. 1.1 | Typical C development environment.

Deitel & Deitel

Higher Level Programming Languages History

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

Review/Summary

- Course Objectives
- Course Operation/Expectations
- Course Plan and Syllabus
- Abstraction and Program Development Environment