



C

Systems Programming

CS2600 Systems Programming

Dr. Amar Raheja

Lecture 1

What is Systems Programming?

- Two kinds of Programming
 - Application Programming
 - Systems Programming
- Applications Programming
 - It aims to produce software which provides services to the user directly
- Systems Programming
 - aims to produce software and software platforms which provide services to other software, and are performance constrained
 - requires a great degree of hardware awareness
 - goal is to achieve efficient use of available resources because the software itself is performance critical

Attributes of Systems Programming

- The programmer exploits the properties of the OS and the hardware that the program runs on
- A low-level programming language is usually used
 - Programs can operate in resource-constrained environments
 - Programs written to be efficient with little runtime overhead, they may have a small runtime library, or none at all
 - Programs may use direct and "raw" control over memory access and control flow
 - The programmer may write parts of the program directly in assembly language

Foundations of Systems Programming

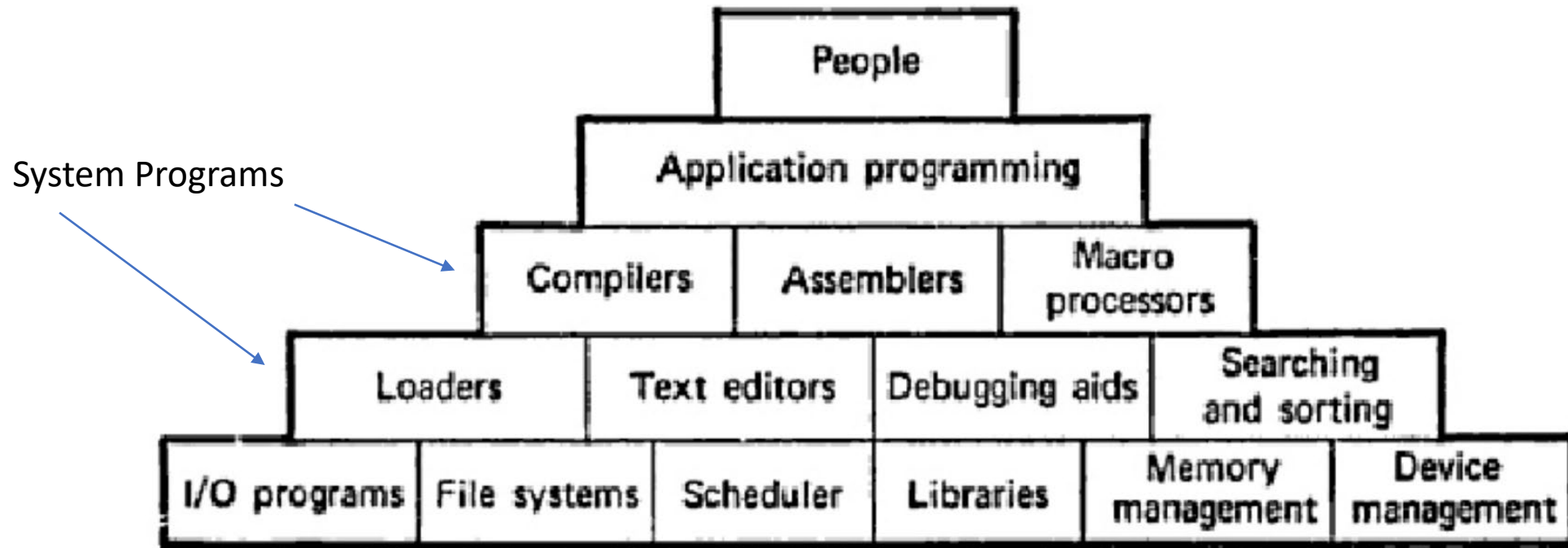


FIGURE 1.1 Foundations of systems programming

Systems Programming

- System programming leads to the development of computer system software that manages and controls the computer operations.
- The low-level codes are very close to the hardware level and deal with things such as registers and memory allocations.
- The system programs or system software coordinates data transfer across the various components and deals with the compiling, linking, starting and stopping of programs, reading from files as well as writing to files.
- Operating System + Programming
- UNIX Operating System + C Programming
- Why C?
 - Learning to program in C gives a set of low-level programming tools that are unmatched by any other programming language.
 - The power of C is its ability to express programming instructions using a combination of low level and high-level constructs.

Learning Systems Programming

- Understand the Basics of the UNIX Operating Systems
 - Logins, shell, environment, file system, file hierarchy, basic utilities, users and groups, root, special device files, system calls
- Understand the Basics of C Programming Language
 - Data types, I/O, Arrays, Functions, Pointers, Memory Management, Data Structures (linked lists using pointers)
- Systems Programming combines the two
 - separate program that is bundled with the kernel, that interfaces to it to achieve its functionality, and that provides higher level services to users.
 - Systems program uses system calls that are calls to a function that is a part of the UNIX kernel.

System Calls

- It is the set of functions provided to the other programs
- System calls are used to request access to the resources of the machine, to communicate with other currently running programs and to start new programs
- Over the years, standards have been developed to regularize at least a subset of system calls common to all systems, for example POSIX
- Many standard library functions for Programming Languages use system calls as part of their code
 - E.g. `malloc()` uses `mmap()` and `brk()`
- System calls grouped by families based on functions
 - Memory management, Time management, Process management

Course Details: Topics

- Basic C syntax (Variables and control structures)
- Functions and Program Structure
- I/O (Console and File)
- Pointer Data types and memory management
- Structures and Linked lists using pointers
- Introduction to UNIX
- UNIX File hierarchy
- Core utilities for file management and process control
- Jobs, I/O and redirection
- Regular expressions
- Advanced utilities, filters (grep, sed and awk)
- Vi and Emacs interactive editors
- UNIX shells – Bourne, Bash, C Shell, Korn Shell
- Shell Scripting
- Unix System Calls

Course Bookkeeping

- Programming Projects: Upload code and report (if needed) to blackboard
- Exams: Each exam tests a set of topics thoroughly. The weight of each exam is listed in the syllabus online
- Warning: Please do not copy projects in any way
 - If I determine that two projects are similar, then both students get an F in the class.
 - Submit an unfinished project and I will still give you at least 50% for your honest effort. Mention the errors or reasons for an incomplete project in your report.
- Make sure you are able to login to the Cal Poly Intranet by using the following command

`ssh username@login.cpp.edu`

- Any Questions or Comments ?

Syllabus: Assessment

Assignments	20%
Programming Projects	30%
Exams (1,2, 4 ,5)	32%(8% each)
Exam (3 and 6)	28%(12%+16%)

- How Do I get an A?
 - Submit work on time
 - Don't cheat
 - Keep on top of material
- Curving: very generous
 - (100-"highest weighted score") is added to your weighted total to determine your grade
 - You are being graded relative to your classmates



No Cheating

- Please don't cheat
- Copying on exams from online resources or course notes
- Using project scripts found online, or from friends in previous classes
- Seriously, please don't cheat
- If caught, you get an F in class and are reported to higher authorities



Inquiries and Suggestions

- Have you ever worked with any variant of UNIX or Linux before?
- Level of comfort?
- How many have Linux machine or a Mac as of now?
- Mac is UNIX backend, so you need not install Linux
- Windows users can install Linux by portioning hard drive or just remote login to the Cal Poly Intranet
- Also look into Windows Subsystems for Linux (WSL)
- UNIX Emulation on Windows: Cygwin (GPL) <http://www.cygwin.com>