Operating Systems

L T JayPrakash

jtl@iiitb.ac.in

Aug 2, 2016

Introduction to the Course

- Introducing each other – Backgrounds
- Goals, Course Delivery and Outline
- Learning Management System (LMS)
- Syllabus and Pedagogy
- Class and Homework Assignments
- Why study Operating Systems?
- Changes in Course Delivery

Course Goals

- Introduce you to operating system concepts
  - Hard to use a computer without interacting with OS
  - Understanding the OS makes you a more effective programmer
- Cover important systems concepts in general
  - Caching, concurrency, memory management, I/O, protection
- Teach you to understand system internals (in Lab)
  - Programming assignments through different offerings
- Prepare you to take second level OS classes
  - Real-time systems, Advanced Operating Systems

Course Delivery and Exam Schedule

- Every Tuesday @ 2 pm, Wednesday @ 915 am.
- Lecture Dates – Aug (2,3,9,10,16,17,23,24,30,31), Sep (6,7,13,14,20,21,27,28), Oct (18,19,25,26), Nov(**1** 2,8,9,15,16,22,23,29,30) – 31 sessions.
- Mid Exam Date – Week of Oct 3, 2016.
- Final Exam Date – Week of Dec 5, 2016.
- For course related queries – After the lecture, or meet me in my office (room # 110).

Course Outline

- Core OS topics.
- Credit policy
  - Homework, quizzes, class tests and assignments – 10% (maximum)
  - Paper Reading and Presentation – 10% (maximum)
  - Mid-term exam – 40% (minimum) (syllabus – till mid-term)
  - End-term exams – 40% (minimum) (syllabus - pre + post mid-term)
- Programming Credits
  - Half of the credit will be based on individual effort, and
  - Half on the success—or lack thereof–of the overall team effort, in case it is a team assignment.
- There may be a small number of surprise quizzes at various points in the semester.

Syllabus

- There are numerous good sources of information for study, but no single textbook may be sufficient due to the evolving nature of the subject.
- I do not mandate a particular textbook on operating systems -- there are many possible alternatives, all of which are reasonable, none of which may cover all.
- I recommend coming to class and be a part of class discussions.
- Read the book in the background (e.g., after we cover a topic in class), both to fill in your knowledge, as well as to get a different perspective on operating systems.
- Any of these books are fine:
  - Andrew Tanenbaum's "Modern Operating Systems", ISBN: 0-13-031358-0
  - Silberschatz, Galvin "Operating System Concepts", ISBN 0-471-41743-2
- Others may be fine too -- just ask.

Topics on processes, memory management, scheduling, deadlocks, I/O, and file systems.

Details of Course

Pedagogy

- Discussion based on theoretical concepts.
- Problem solving exercises.
- Programming assignments to get a hands-on experience of working of an OS. Most of them through the weekly lab session.
- Pattern of Exam QP
  - Less theory, more problem solving.
  - Multiple choice questions, a possibility.
  - True / False with reasoning (mandatory).
  - Comparison type, definitions, short questions etc.
  - Long theory questions like short notes will **NOT** be part of any exam.

Homework / Assignment

- The assignments will be posted on LMS.
- Doing well in assignments is indicative of your doing well in the course overall.
- The performance in assignment may be considered as a factor in deciding the score for individual performance in the group exercises.
- The assignments could be any of the following types – finding internal details, programming, problem solving, paper reading etc.
- They may have a submission deadline, or in case no deadline is given, the default deadline will be Dec 1, 2016 (firm).
- Submission : LMS.
- Sharing and collaboration are great, but must respect rules! Team work is always encouraged.
- Use of LATEX is preferred for all documentation.

Why study Operating Systems?

- Operating systems are a maturing field
  - Most people use a handful of mature OSes
  - Hard to get people to switch operating systems
  - Hard to have impact with a new OS
- High-performance servers are an OS issue
  - Face many of the same issues as OSes
- Resource consumption is an OS issue
  - Battery life, radio spectrum, etc.
- Security is an OS issue
  - Hard to achieve security without a solid foundation
- New “smart” devices need new OSes
- Web browsers increasingly face OS issues

Big Data, Networks, large-scale software systems, whatever --- Key to its success is Operating System Design Principles, its working and optimal use of its policies.

Changes in Course Delivery

- Theory necessary for the foundation, however, practical approach preferred.
  - Problem-solving and programming exercises
- Supported by lots of research papers, possibly original articles to get the depth of topics.
  - We may not be able to discuss every paper in the class, but pointers will be provided and interested students may like to read the original paper along with related book chapters.

- What's next?
  - May be the class is over for today?
  - May be a surprise test?
  - ……..

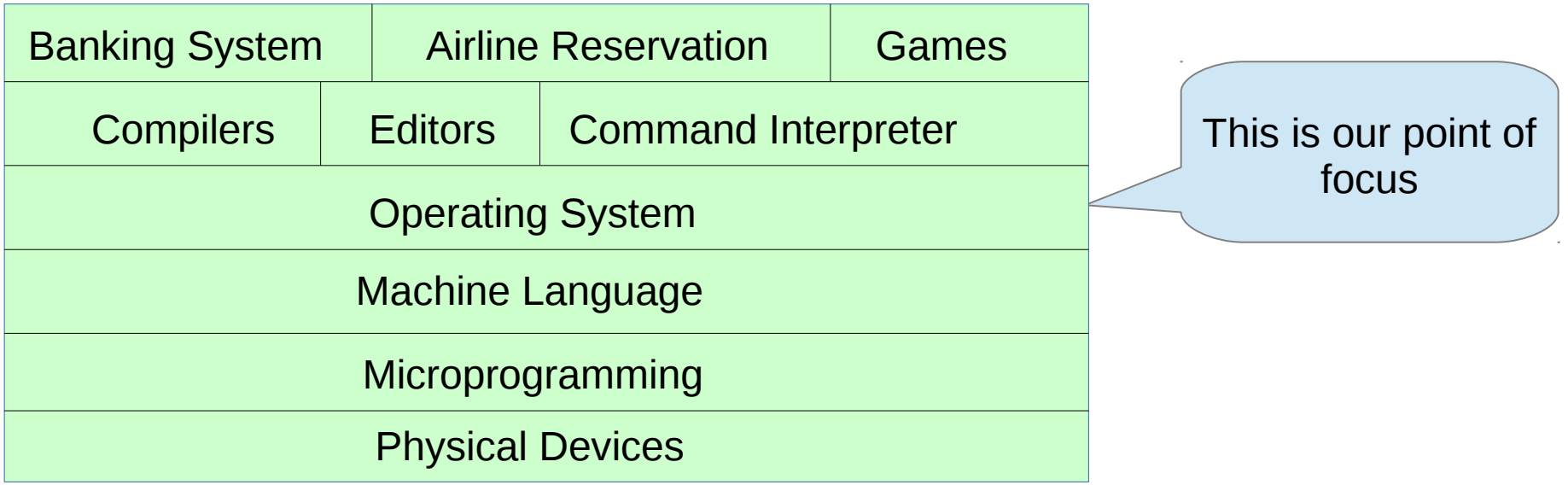
Submit a One Page Writeup

- What you know about OS?
  - 2 sentences
- What you want to know about OS?
  - 2 sentences
- Your experiences with using an OS?
  - 2 sentences each on your using Windows, Linux, and Android.
- Writing Roll No. And Name on the writeup is not mandatory.

What is an OS?

- Tool to make programmer's job easy
- Resource allocator
  - Must be fair; not partial to any process, specially for process in the same class
  - Must discriminate between different class of jobs with different service requirements
  - Do the above efficiently
    - Within the constraints of fairness and efficiency, an OS should attempt to maximize throughput, minimize response time, and accommodate as many users as possible
- Control program
- Tool to facilitate efficient operation of a computer system
- Virtual machine that is easier to understand and program
  - Encapsulates the complexities of hardware

Typical OS Architecture Design



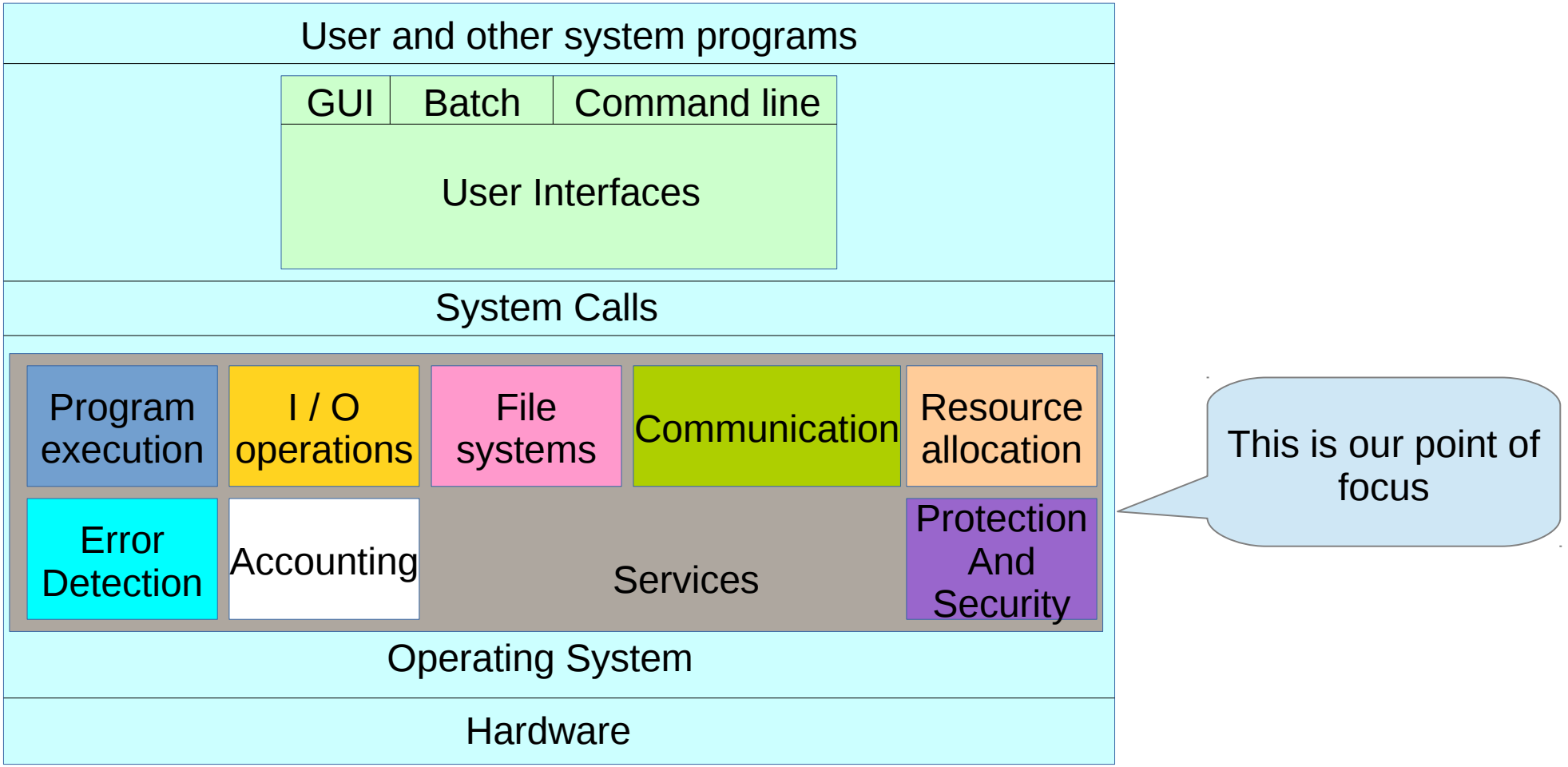
- Typical OS structure
  - Application environment - shell, mail, text processing package, sccs
  - Operating system - support programs for applications

A microinstruction program that controls the functions of a central processing unit or peripheral controller of a computer.

A set of microinstructions that defines the individual operations that a computer carries out in response to a machine-language instruction.

..... many more definitions published and available in different literature.

Zooming into the OS Layer



A view of Operating System services

Thank you

See you back
Aug 03, 2016 at 915 am.