CS343: Operating System

Introduction to Operating System

Lect02 : 31th July 2023

Dr. A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

Outline

- What do we study in this course?
- Why should this be studied?
- What is *"Operating System"* ?
- How is the course structured?

OS as a Resource Manager

- Allow multiple programs to run at the same time
- Manage and protect memory, I/O devices, and other resources
- Includes multiplexing (sharing) resources in two different ways:
  - In time
  - In space

Abstraction

- Delving into the depths reveals more information

- An abstraction omits unneeded detail, helps us cope with complexity

Software Abstraction

```
int sum(int x, int y)
{
  int t = x+y;
  return t;
}
```

C

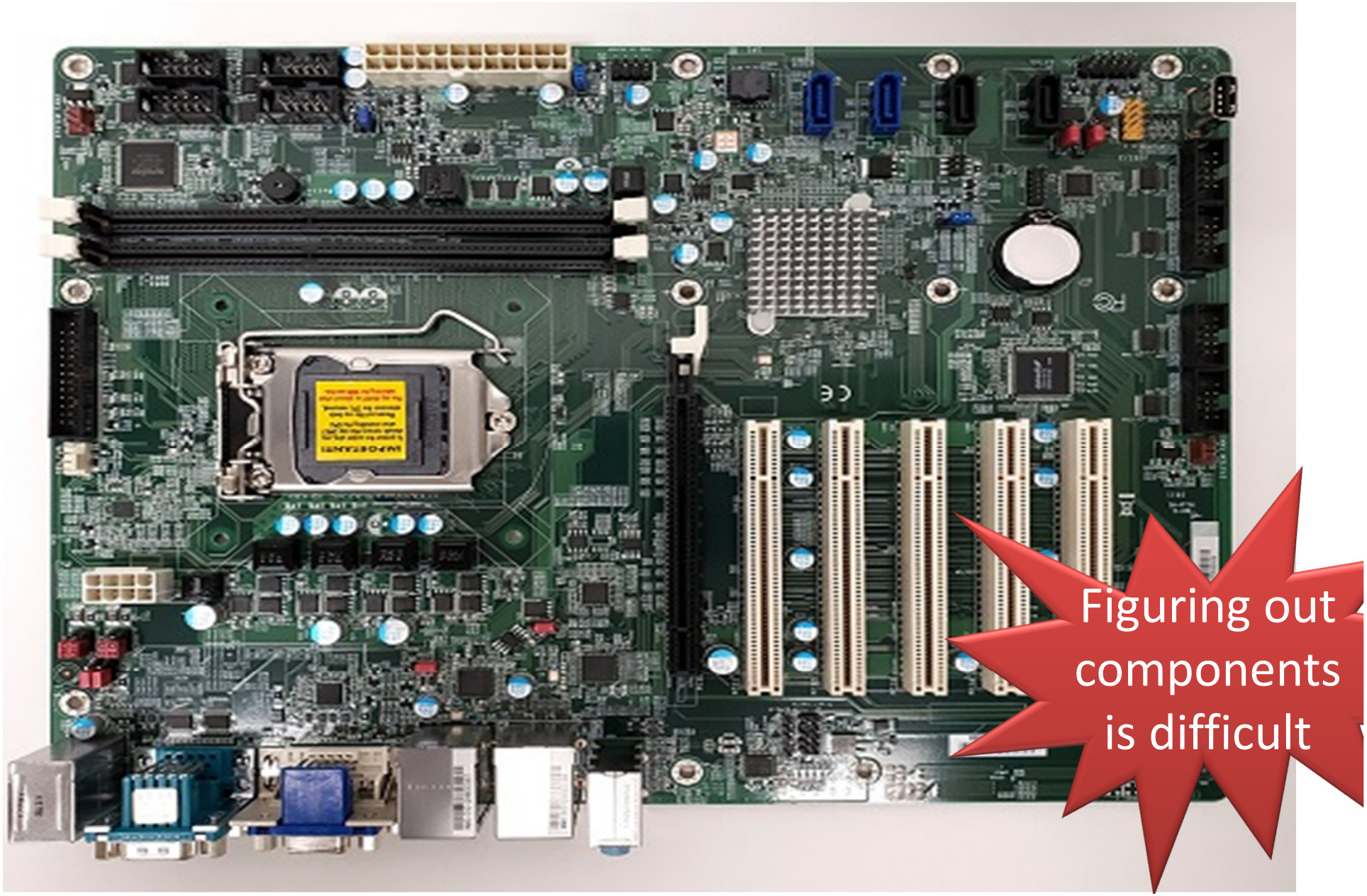
```
_sum:
      pushl %ebp
      movl %esp,%ebp
      movl 12(%ebp),%eax
      addl 8(%ebp),%eax
      movl %ebp,%esp
      popl %ebp
      ret
```

assembly

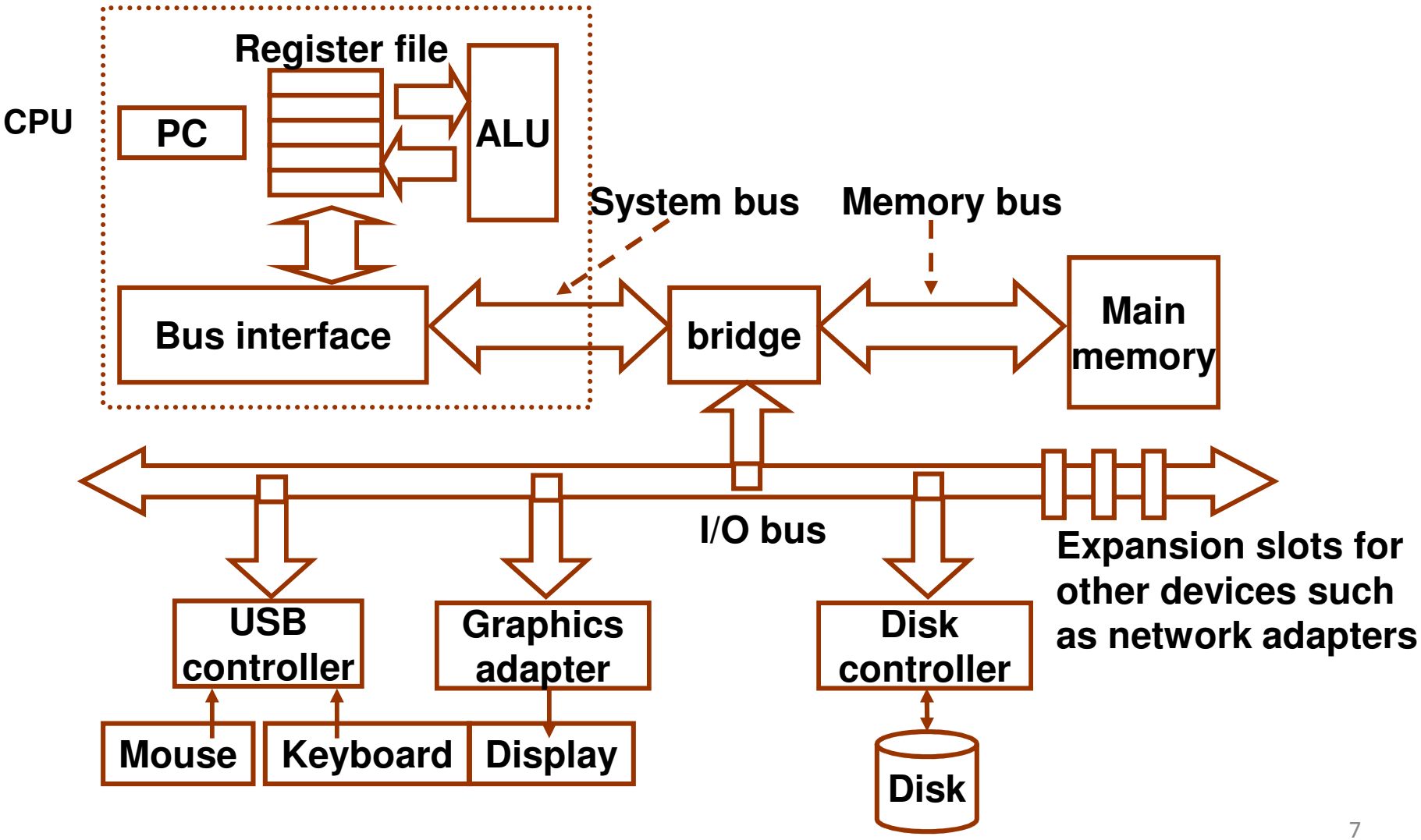
```
0x401040 <sum>:      0x55
                     0x89
                     0xe5
                     0x8b
                     0x45
                     0x0c
                     0x03
                     0x45
                     0x08
                     0x89
                     0xec
                     0x5d
                     0xc3
```

machine code

Example: Hardware Abstraction



Hardware Abstraction



Operating System

- Bridge the gap between hardware and software
- Establish a foundation for building higher-level programs
  - How to optimize programs?
  - How to debug large systems
  - How to deal with complexity

Operating System

- Provides a virtual execution environment on top of hardware
  - That is more convenient than the raw hardware interface
- "All of the code you did not write"
- More Simple, More reliable, More secure, More portable, More efficient……

What do OSes do?

- Manage physical resources
- Provide virtual resources
- Implement mechanisms and enforce policies for the control and use of resources
- Mediate the interaction of mutually distrusting applications

What Physical Resources Do OSes Control?

- CPU, Memory
- Storage Devices, Networks
- Input Devices (keyboard, mice, cameras)
- Output Devices (printers, displays, speakers)

- And many virtual resources

Issues In OS Design

- **Structure:** how is an OS organized?
- **Concurrency**: how are parallel activities created and controlled?
  - All hardware are run in parallel CPU, HDD, N/W
  - User want many application to run in parallel…
- **Sharing**: how are resources shared?
- **Naming**: how are resources named by users?

Issues In OS Design

- **Protection:** how are distrusting parties protected from each other?
- **Security:** how to authenticate, authorize and ensure privacy?
- **Performance:** why is it so slow?

More OS Issues

- **Reliability:** how do we deal with failures?
- **Extensibility:** how do we add new features?
- **Communication:** how do we exchange information?
- **Scale:** what happens as demands increase?
- **Persistence:** how do we make information outlast the processes that created it?
  - Don't start a game every time from the start..
- **Accounting:** who pays the bills and how do we control resource usage?

Why different Operating System?

- What is the difference between OSes desktops, laptops, mobile phones, washing machines etc.?
  - Performance /speed, Power consumption, Cost
  - General purpose /special purpose
  - Domain Specific: Network, DSP, Image. Crypto

Why different Operating System?

- **Different OS optimization parameter lead to different problem hence the different algorithm/heuristics**
- Is Android OS is same as Ubuntu ? === > No
  - **Power and memory consumption is very crucial in Phone**
  - **Battery level input need to monitor**

Why to Learn Operating Systems?

- Provides an understanding from the bottom up
- Even if few people build OSes, understanding how OSes work is crucial for building working systems
- Who knows..
  - You got places in Microsoft, Google, HP, Amazon..
  - And need to work in OS, device drive, multicore….

Why to Learn Operating Systems?

- This course will go far beyond OS design to cover all aspects of computer system, including
  - **Concurrency,**
  - **Synchronization,**
  - **Input/output, filesystems,**
  - **Networking, routing,**
  - **Distributed systems and** so forth
- Engineering pride alone requires full understanding

CS343 Course Objectives

To learn –

- Theory behind resource management
- Theory behind concurrency and related issues
- How to adopt to new hardware resources and standardization
- Issues affecting modern operating systems (Android, VxWork, Distributed OS,)

CS343 Course Objectives

To learn –

- Advanced Power Management Support
- Multi-core support in OS
- How ***device driver*** and how all the interface works
- How the system call works..

OS Motivation: Resource Management

- At the hardware level resources are typically dedicated.
- An OS provides versions of these resources that are
  - Either virtualized (each user gets the illusion of having a copy of the resource)
  - Or arbitrated (one user at a time, but with queuing handled by the OS).

OS Motivation: Resource Management

- Strategies used to give multiple users access to a dedicated physical resource
  - Also used in many user-level programs.
- By studying these issues explicitly
  - You will learn patterns that can be reused in many other contexts.

OS Motivation: Concurrency

- Writing concurrent code is not easy
  - Especially using threads with shared memory and locks.
  - A large number of current CS students will do this at some point in their careers.
- Now we can see 256-288 processors in Desktop System
  - **Intel KNL XeonPhi, AMD Thread ripper processor**
- There is a growing trend to address concurrency
  - Introduction to threads, races, deadlocks

OS Motivation: Concurrency

- The material is hard
  - **Actually the material is easy, but applying it is hard**
  - **Useful to see it several times before graduating.**
- A solid introduction to concurrent programming
  - A major benefit of taking an OS course

OS Motivation: Performance Analysis and Contention Resolution

- When resources are shared, contention typically follows.
- Contention can be resolved in many ways
  - Using queuing, fair sharing, or prioritization.
- In some cases, such as CPU scheduling
  - No single technique suffices and the best known solutions are bizarre hybrids.

OS Motivation: Performance Analysis and Contention Resolution

- Often, the most interesting problem
  - Figuring out what kind of contention is the root cause of some observable problem.
- An OS is a perfect context for learning these ideas, whose applicability is much broader than computer science
- ***Critical Section, Deadlock and Locking***
  - ***Theoretical Analysis and Proof of Algorithms***

OS Motivation: Interfaces and Hiding Complexity

- A well-designed interface is beautiful thing
  - It is even more beautiful to fully appreciate to transform a nasty, low-level interface
  - Ability to put a collection of useful abstractions
  - **Sockets, file systems, and address spaces into a single convenient package**
  - **Probably one of the best contributions of computer science.**
- This is so common place that it's easy to overlook the real awesomeness.

OS Motivation: No Magic Here

- It's easy to view the OS as a magical force
  - **Good : Giving us smooth multitasking, efficient storage management, etc**
  - **Evil : giving us blue screens, thrashing, security problems, and scheduling anomalies.**
- Public and You
  - This view is fine for the general public.
  - On the other hand, if you plan to tell people you're a computer scientist, you need to have peeked behind this curtain.

OS Motivation: No Magic Here

- What will you find there in OS kernel?
- Too often it seems like sort of a sad collection of
  - Dull linked lists,
  - Dodgy resource heuristics,
  - And ill-maintained device drivers.

OS Motivation: No Magic Here

- A good OS course should show students ➔
  - There is great code in the kernel
  - You just have to know where to look.
- When you first see it
  - You will not understand
  - But when you understand a bit, feel easier……
- Mostly, kernel code is perfectly earthly world
  - Anyone can write it,
- A bit more care and attention to detail than
  - User-mode code because the consequences of a bug are greater.

OS Motivation: Dealing with Big Software

- OS code : multi-million line codes and base is a nightmare
  - Documentation is incorrect and scattered
  - Interfaces are clunky (makes noises), wide
  - Interactions are subtle (so precise as to be difficult to analyze),
  - And error messages inscrutable (impossible to understand or interpret)
- But welcome to real life
  - We can't usually just start over due to problems like these.

OS Motivation: Dealing with Big Software

- As a student: if you can begin to develop
  - A systematic approach to learning the relevant parts of a big piece of software that your code has to fit in
  - Your life will be a lot easier later on.
- You can hate the Linux kernel
  - But it's far better software than other stuff you'll run into during your career.

OS Motivation: System Design

- Designing any engineered system, including a software system
  - Is an exercise in compromise.
- How much emphasis is placed on
  - Reliability? Performance? Cost? Maintainability?
- Since operating systems are large
  - Performance-critical programs that tend to last for decades
  - They are a great place to learn about these kinds of tradeoffs.

OS Motivation: System Design

- Students who develop a sharp eye
  - For finding an appropriate design point are incredibly useful in industry
- This stuff is more of an art than a science
  - You need to look at a lot of code
  - Understand the issues,
  - And learn to think about this stuff for yourself.

Fact

- There has never been as exciting a time to work on systems hardware and software as now!!!
- The world is increasingly dependent on computer systems
  - **Connected, networked, interlinked**
- People just **do not know how to build** good systems…
  - Still evolving…..**heuristics and heuristics**….

Thanks