CS304 Operating Systems

DR GAYATHRI ANANTHANARAYANAN

gayathri@iitdh.ac.in

Materials in these slides are borrowed from textbooks and existing operating systems courses

Course Details & Logistics

Monday – 10.30 AM to 11.20 AM

Wednesday – 11.30 AM to 12.20 PM

Friday– 8.30 AM to 9.20 AM

TA information:
Chandrasekhar - chandrashekar.s@iitdh.ac.in
Omkar Shende – 212011004@iitdh.ac.in

CS314 – Operating Systems Lab

Thursday – 3.20 PM to 5.55 PM

Books:

"Operating Systems: Three Easy Pieces" - by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau (University of Wisconsin-Madison) [https://pages.cs.wisc.edu/~remzi/OSTEP/]

"Operating System Concepts", 8th edition, by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, Wiley-India edition

"Operating System: Design and Implementation (The Minix Book)", 3rd edition by Andrew S Tannenbaum and Albert S Woodhull

Tentative Grading Components

CS304 – OS Theory

Quizzes (2) – 25 % **(Feb 1, March 21)**

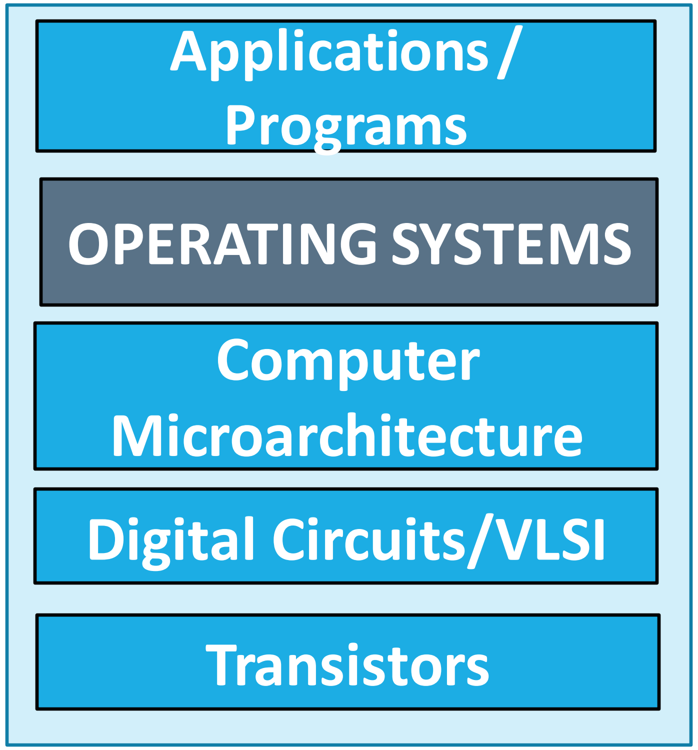
Midsem – 25% (**Feb 19 –25**)

Endsem – 40% (**April 13- April 20**)

CS314 – OS-Lab

1. 10 Labs/Experiments in total (each of varying complexity) - **75% {Jan 4 – April 3}**
2. For each experiment, marks will be awarded based on code and report submitted, Demos & Viva
3. The experiments will be performed by each student (Some experiments will be done in Groups of 2 and will be clearly mentioned in the lab handout)
4. **Lab Test/Exam – 25% - {April 4, 2024}**
5. Minix OS will be used for some assignments and Linux also will be used for some. Students can install minix on their laptops or can make use of the desktops available in the SP-16 lab. The decision needs to be conveyed (laptop/desktop) to the TA Chandrashekar (latest by EoD 4th January) so that appropriate arrangements can be made before the next lab.

Abstractions in a Computer System



What is an OS?

Middleware between user programs and system hardware that provides the following:

Abstraction

OS makes HW easy-to-use and program by providing interfaces

Virtualization

OS creates an illusion of dedicated HW for each user and application

Concurrency

OS enables controlled interaction between multiple applications

What is an OS?

Manages hardware: CPU, main memory, I/O devices (disk, network card, mouse, keyboard etc.)

- Hardware Abstracter - turns hardware into something that applications can use

- Resource Manager - manage system's resources [What about security?]

What happens when you execute a program on the CPU?

Execution of a Program

1. A compiler translates high level programs into an executable (“.c” to “a.out”) [eg. gcc, icc]
2. The exe contains instructions that the CPU can understand, and data of the program (all numbered with addresses)
3. Instructions run on CPU: hardware implements an instruction set architecture (ISA)
4. CPU also consists of a few registers, pointer to current instruction (program counter or PC), Operands of instructions, memory addresses

Execution of a Program

To run an exe, CPU

–fetches instruction pointed at by PC from memory

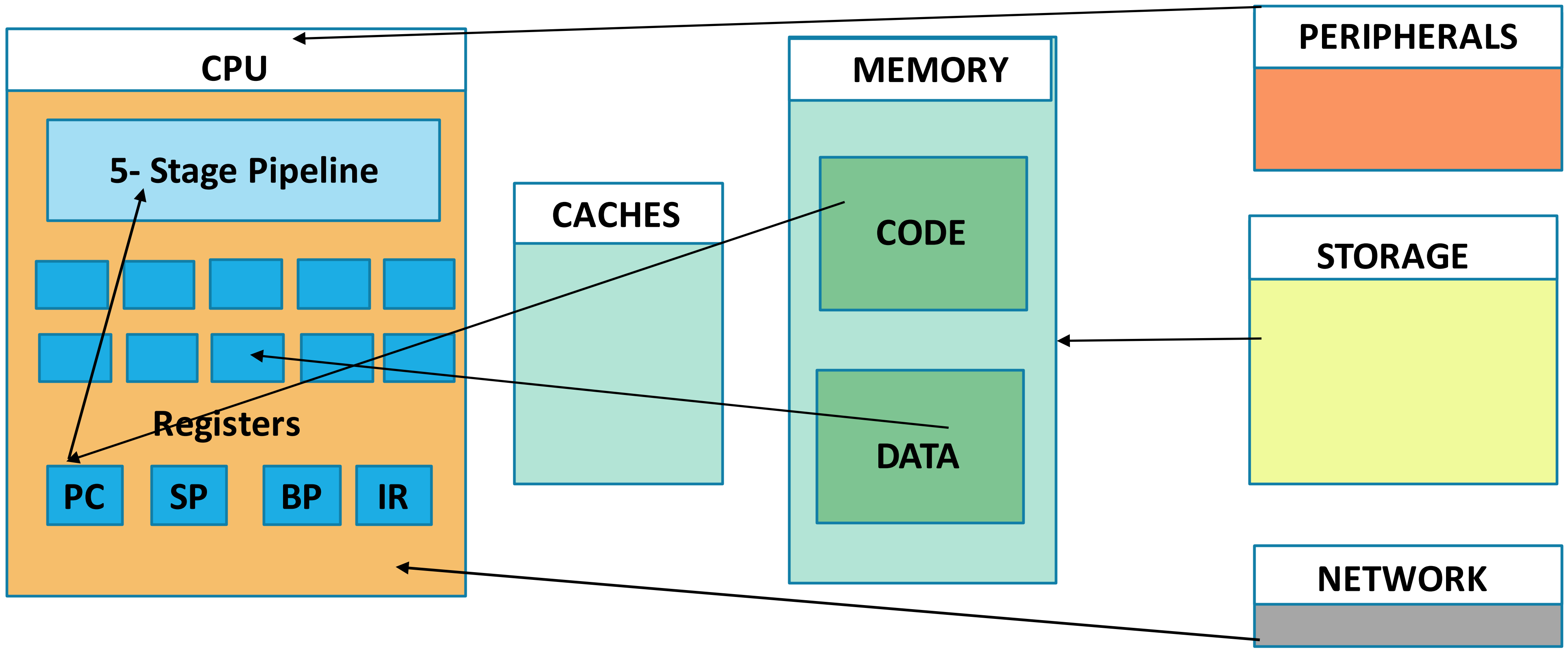
–loads data required by the instructions into registers

–decodes and executes the instruction

–stores results to memory

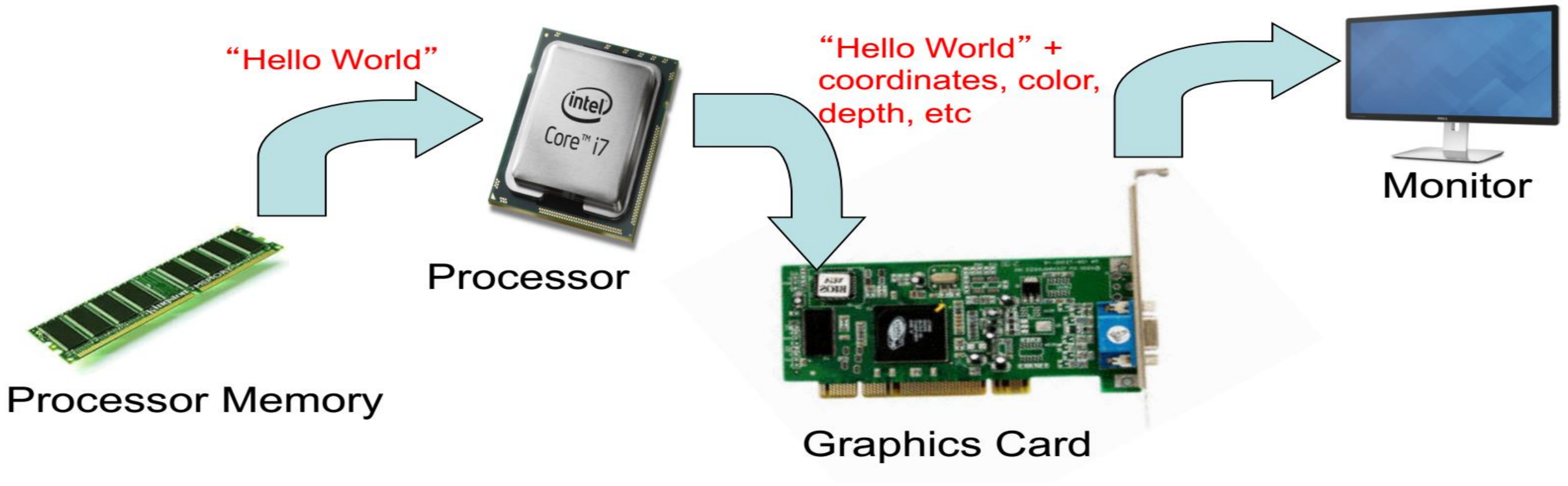
Most recently used instructions and data are in CPU caches for faster access

Demystifying Program Execution



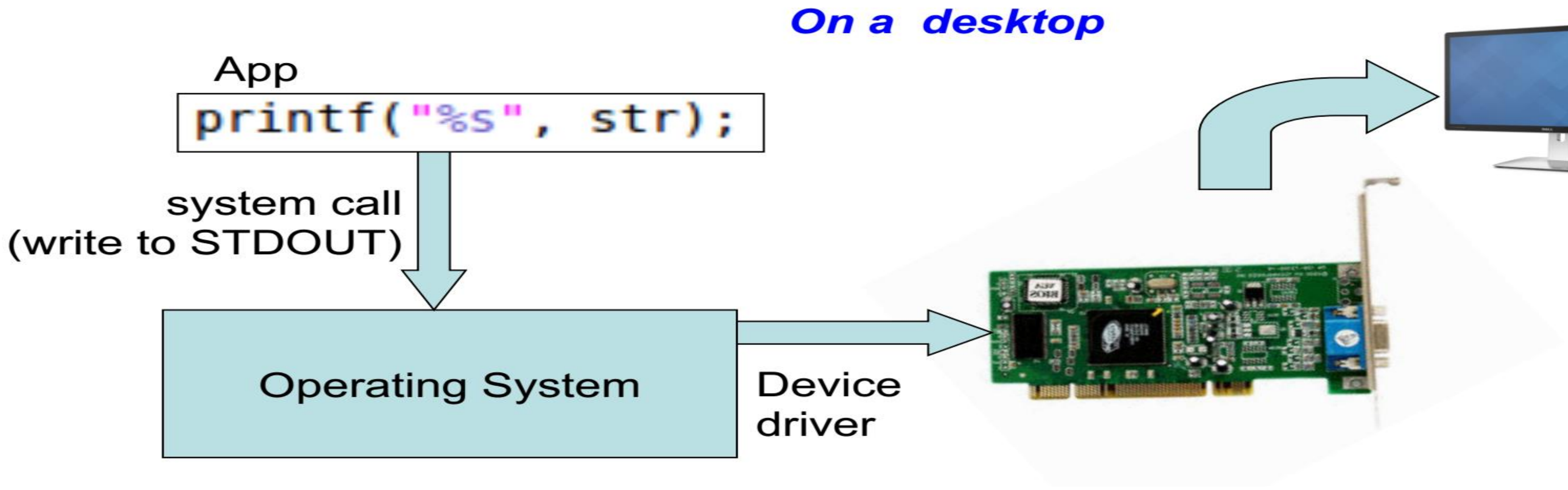
What happens in this case?

```c
#include<stdio.h>
int main(){
char str[] = "Hello World\n";
printf("%s", str);
}
```



Hardware Dependent - What should be done for a new hardware? Can be complex and tedious

Without an OS, all programs need to take care of every nitty gritty detail.



Easy to program apps – No more nitty gritty details for programmers
Reusable functionality – Apps can reuse the OS functionality
Portable - OS interfaces are consistent. The app does not change when hardware changes

What is the role of OS?

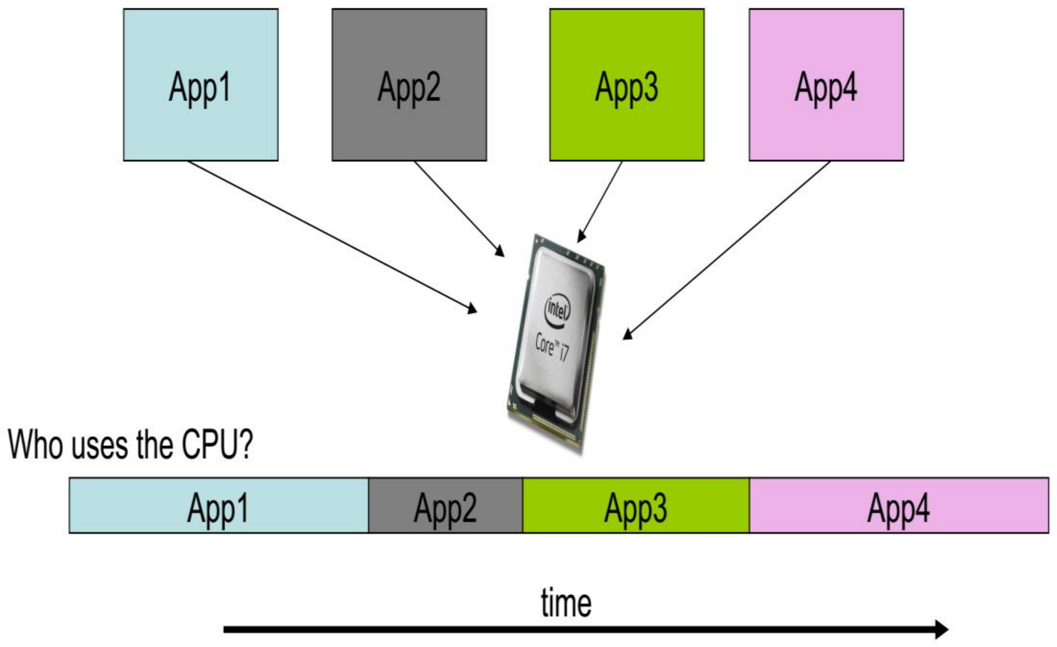
1. Provides basic interfaces

- Loads our executable with instructions and data from disk to memory
- Initializes processor state such as setting PCs, interrupt masks etc
- Interfaces external devices such as disks, keyboard, display etc

What is the role of OS?

2. Creates Process abstraction

- Defines the Process 'state'
- Virtualizes the hardware across processes
- Schedules multiples processes with timesharing
- Enables data sharing between processes
- Provides support for concurrent threads
- Provides synchronization among threads



OS provides the process abstraction

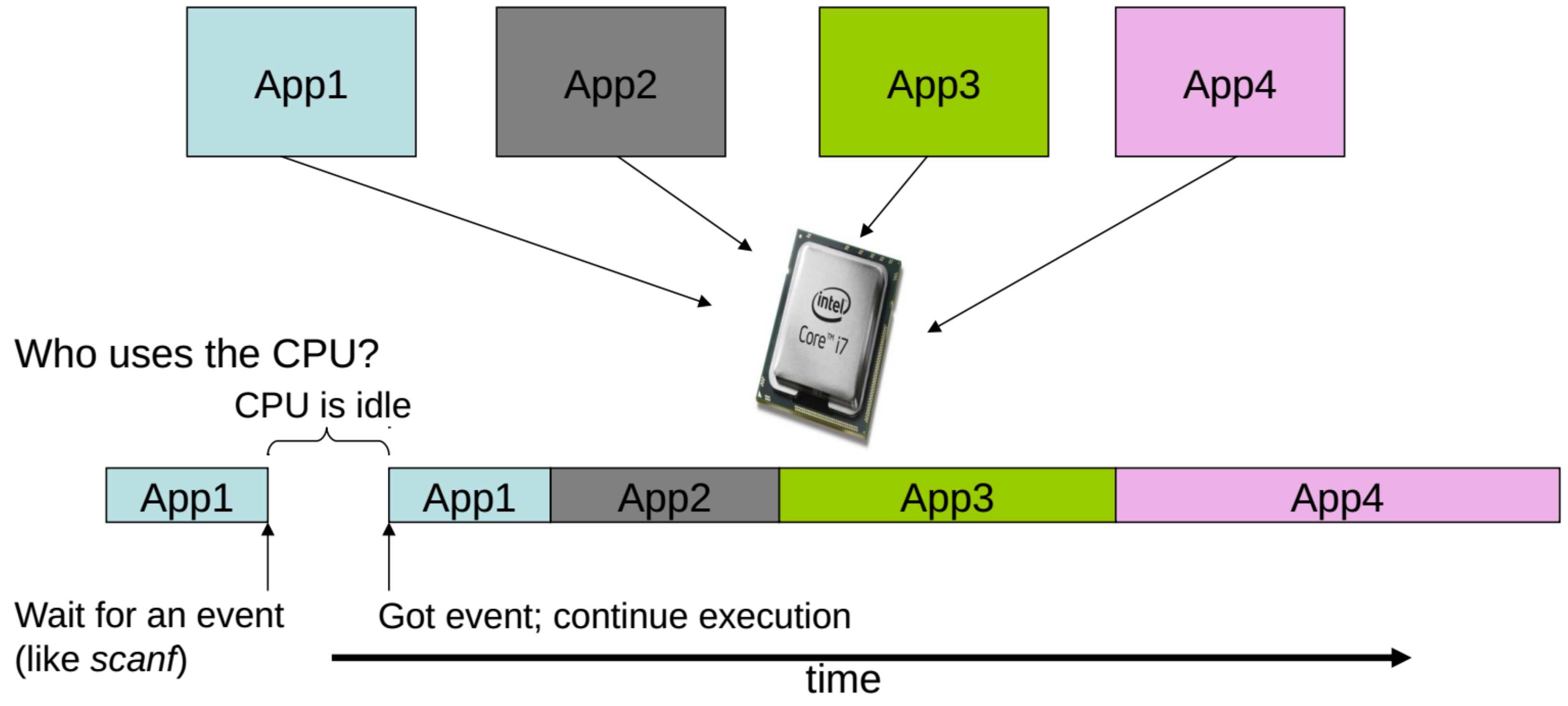
–Process: a running program

–OS creates and manages processes

•Each process has the illusion of having the complete CPU, i.e., OS virtualizes CPU

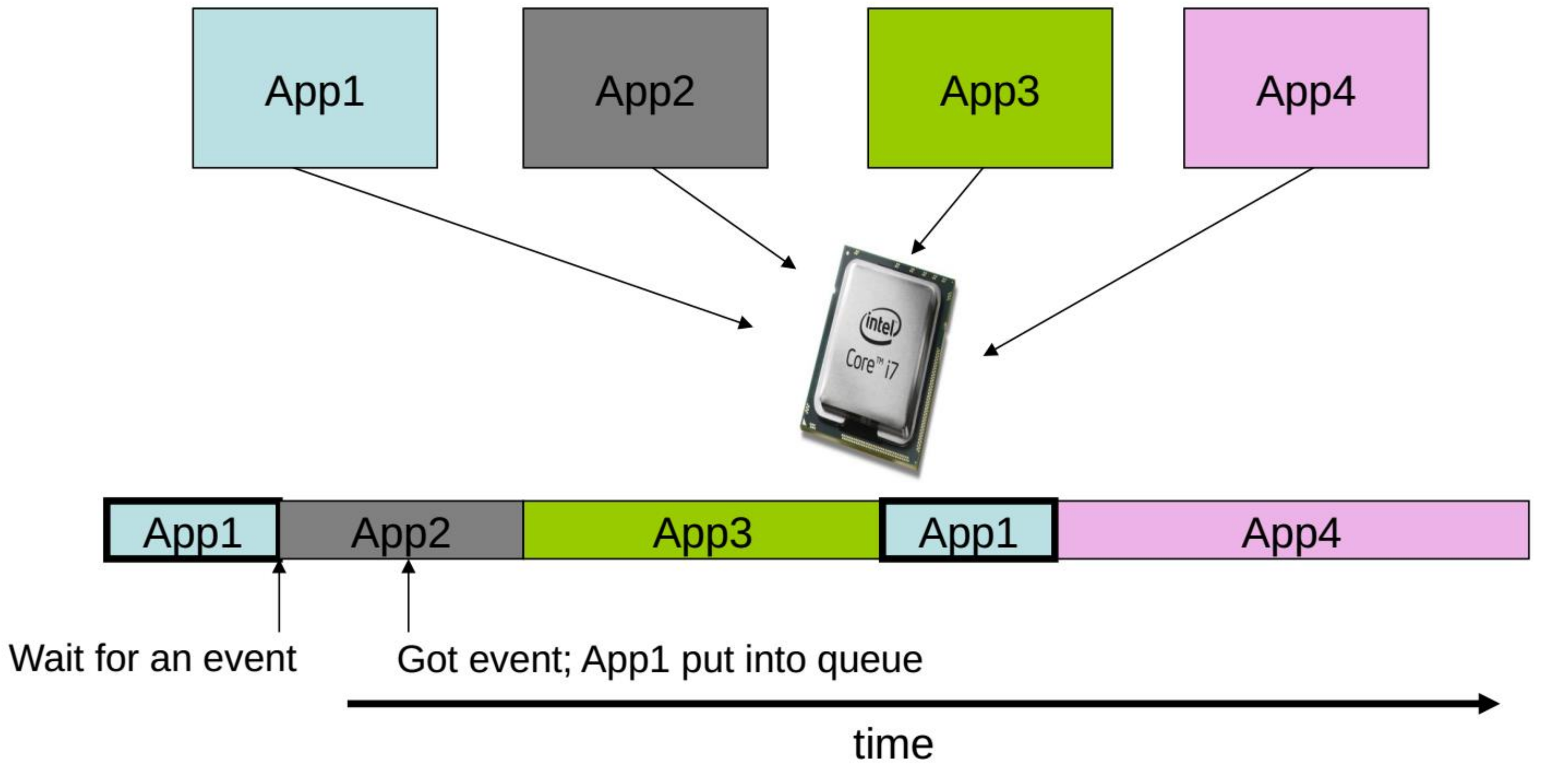
•Timeshares CPU between processes

•Enables coordination between processes



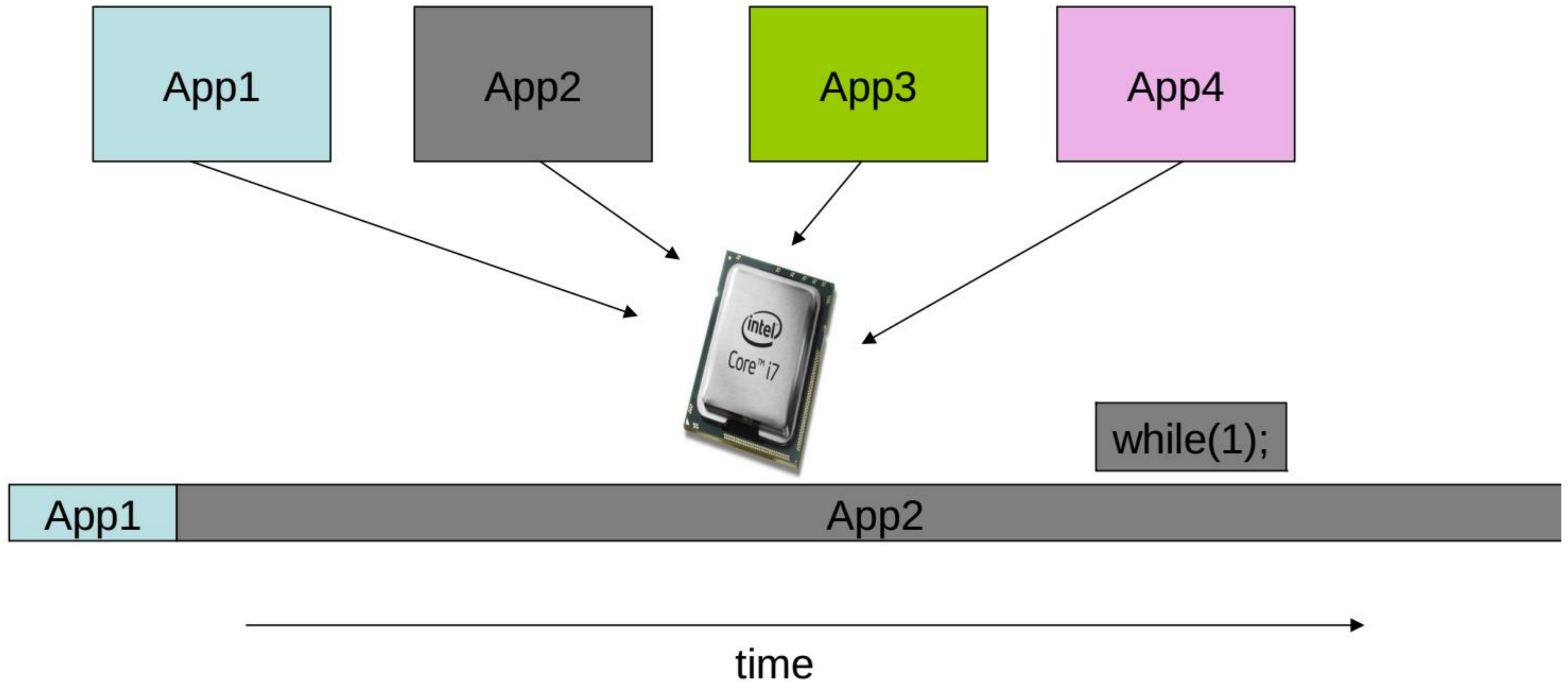
CPU is idle when executing app waits for an event.
Reduced performance.

OS Multiprogramming Support



When CPU idle, switch to another app

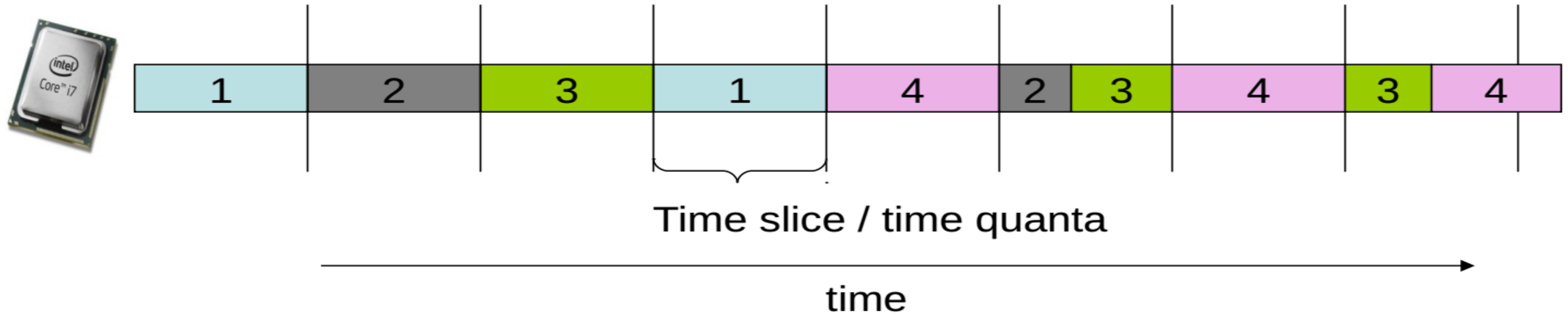
Multiprogramming – Starvation ??



One app can hang the entire system

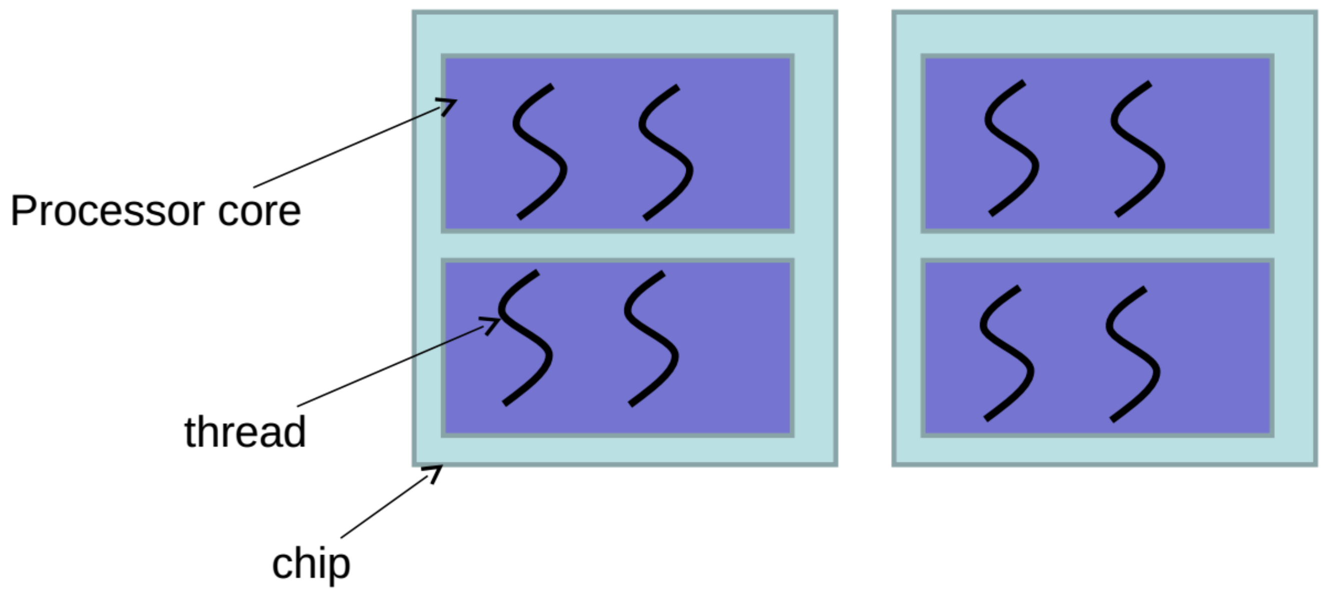
Time Sharing Support

- Time sliced
- Each app executes within a slice
- Gives impression that apps run concurrently
- No starvation. Performance improved



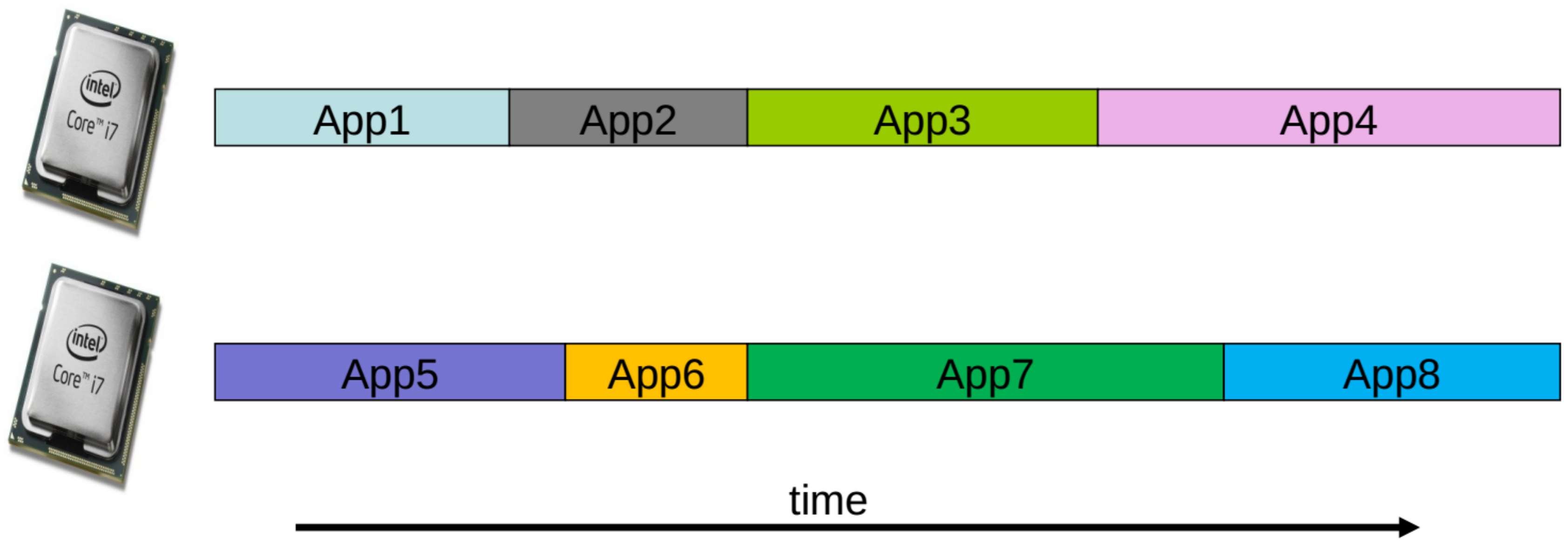
Multiprocessors

- Multiple processors chips in a single system
- Multiple cores in a single chip
- Multiple threads in a single core

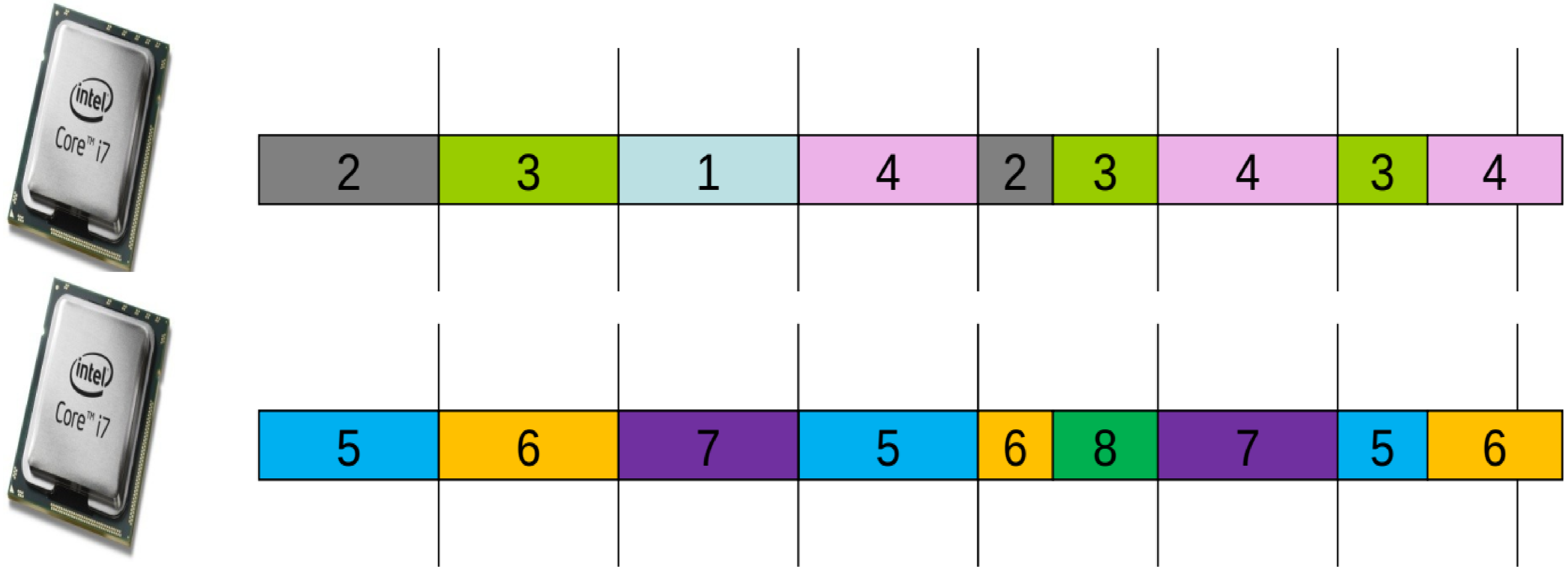


Multiprocessors

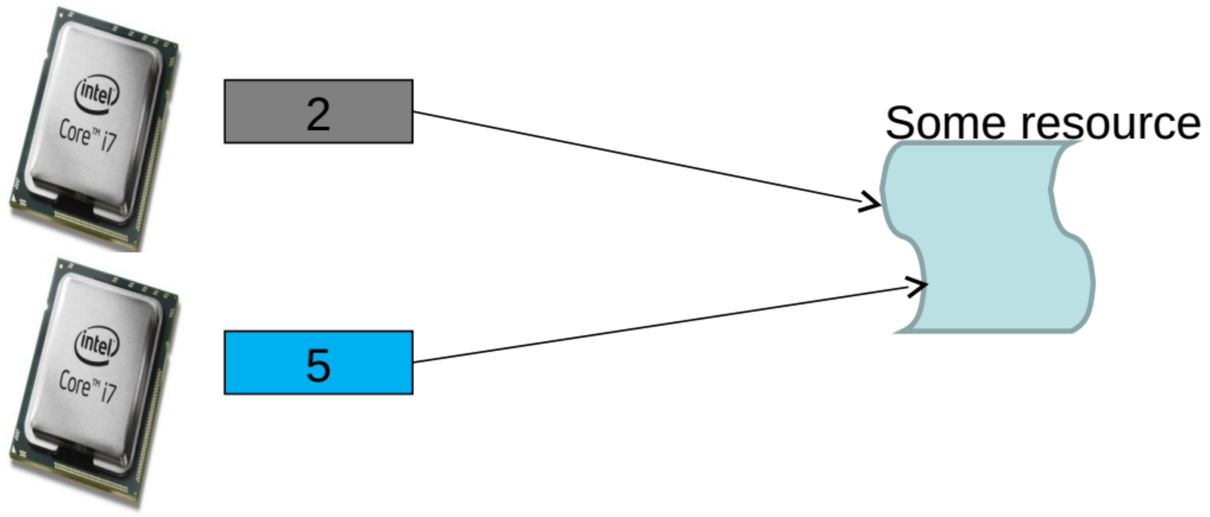
- Each processor can execute an app independent of the others



Multiprocessors and Multithreading

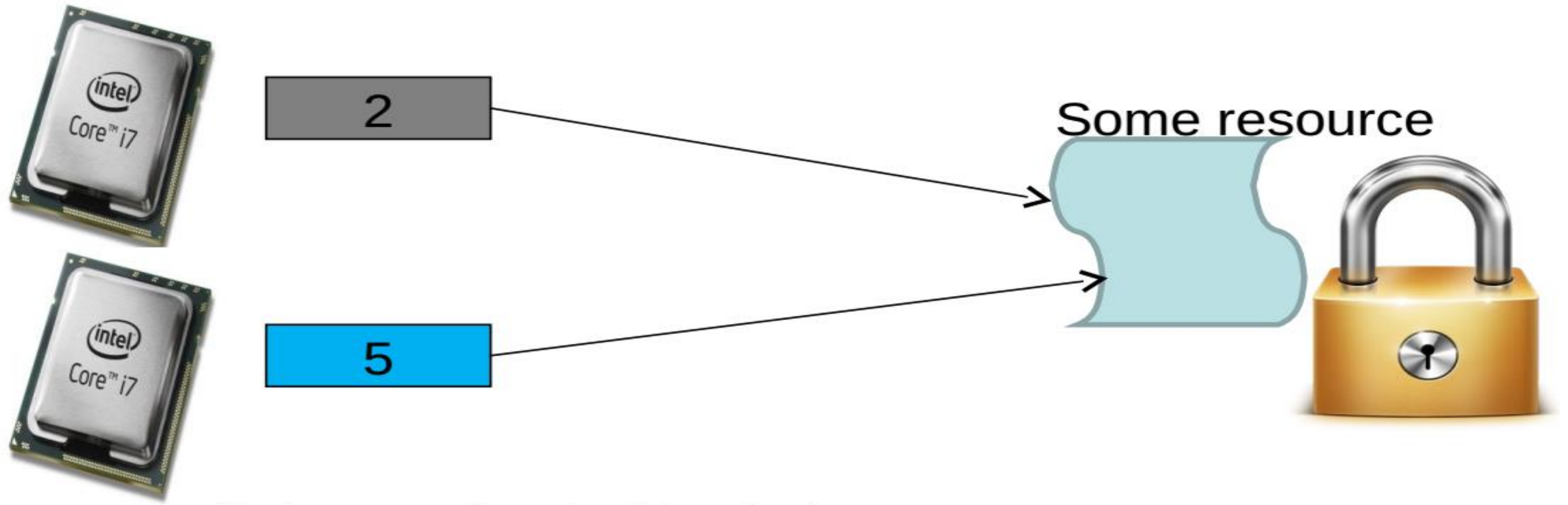


Race conditions



- App2 and App5 want to write into some resource (like a file) simultaneously
- This results in a race condition
  - Need to synchronize between the two Apps

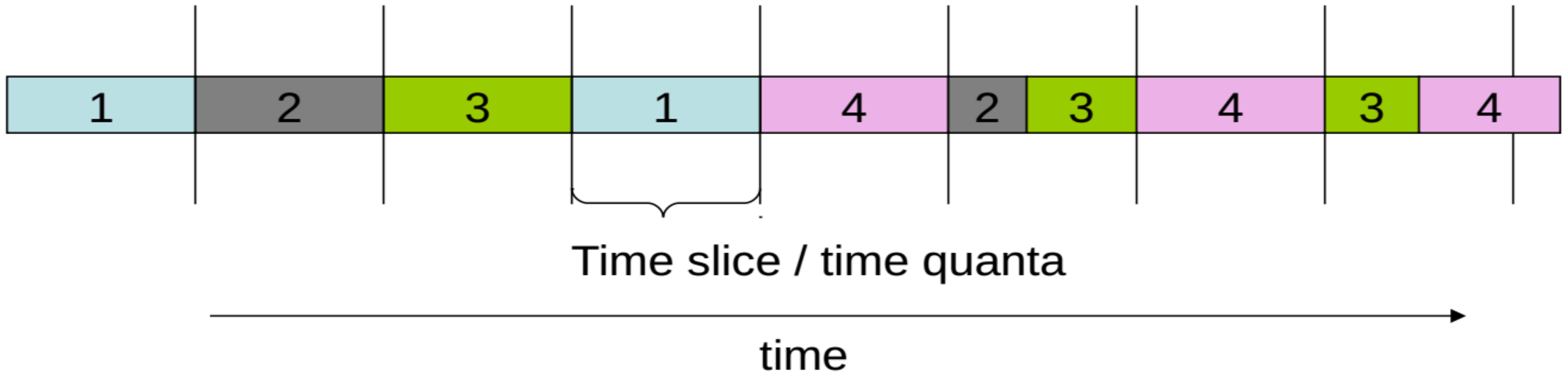
Synchronization



- The shared file is associated with a lock
- The lock ensures that only one App can access the resource at a time
- Sequence of Steps
  - App X locks the resource
  - App X accesses the resource, while App Y waits
  - App X unlocks the resource
  - App Y can now lock and then access the resource

Who should execute next ?

- Scheduling
  - Algorithm that executes to determine which App should execute next
  - Needs to be fair
  - Needs to be able to prioritize some Apps over the others



What is the role of OS?

3. Manages the memory

- Maintains process state (code, data, stack, heap)
- Creates illusion of virtual memory of each process
  - Each process thinks it has a dedicated memory space for itself, numbers code and data starting from 0(virtual addresses)
- Provides interface to storage with physical address
  - OS abstracts out the details of the actual placement in memory, translates from virtual addresses to actual physical addresses
- Ensure isolation amongst processes

OS and Isolation

Why is it needed?

- Multiple apps execute concurrently, each app could be from a different user. Therefore needs isolation.
- Preventing a malfunctioning app from affecting other apps

What is the role of OS?

4. Manages other devices

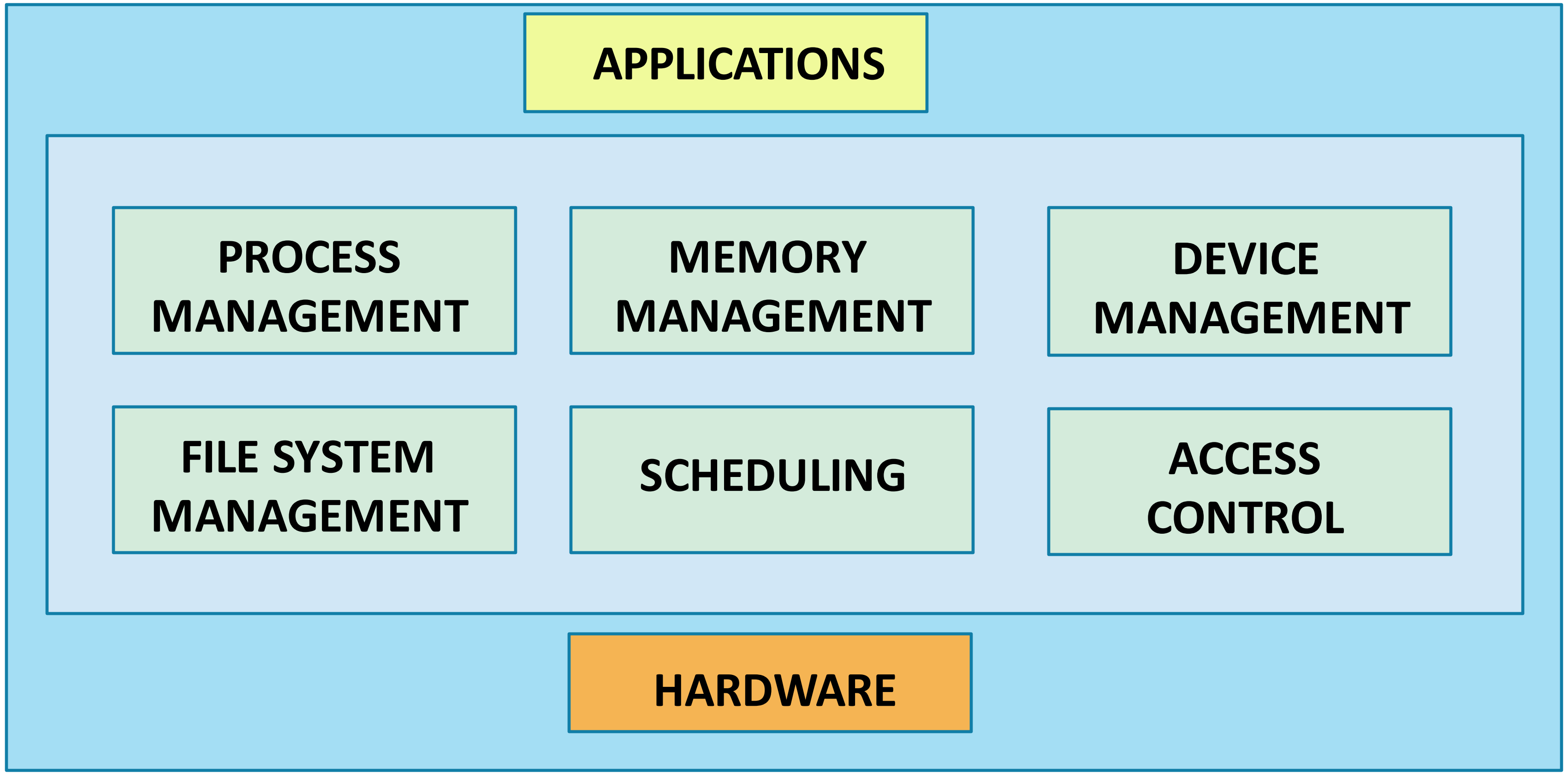
- Provides device drivers which provide interfaces to other devices such as disk, network card, GPU (Device driver talks the language of the hardware devices)
  - Issues instructions to devices (fetch data from a file)
- Responds to interrupts from other devices such as keyboard press, FTP request
- Maintains persistent data in files which is available even after a reboot

What is the role of OS?

5. Provides Security

- Defends against attacks from malware and sensitive data
- Supports multiple users with access control, provides password protection, encryption, biometrics, security assessment

Role of OS in a nutshell



Summary

An operating system is a layer of systems software that :

- Directly has privileged access to the underlying hardware;
- Hides the hardware complexity;
- Manages the hardware on behalf of one or more applications according to some predefined policies;
- In addition, it ensures that applications are isolated and protected from each other;

Trivia Time

Which of the following are likely components of an operating system ? Write all that apply

A) File editor

B) File System

C) Device Driver

D) Cache Memory

E) Web Browser

F) Scheduler

OS Examples

Based on the environment in which the hardware is operated (Desktop/ Server/Embedded)

Microsoft Windows – OS since early 1980's

Unix Based – Originated at Bell Labs in 1960's

- Mac OS X (BSD) Berkley System Distribution
- Linux – Open source, bundled with may popular s/w libraries [Ubuntu, CentOS]

Android – Embedded form of Linux

IOS -

Symbian

What makes each of them different ? Design & Implementation