



Operating Systems

Introduction

Seyyed Ahmad Javadi

sajavadi@aut.ac.ir

Fall 2021

My Background and Contact Details

- Seyyed Ahmad Javadi
 - PhD from New York State University at Stony Brook
 - Postdoc from University of Cambridge
 - Interest: Cloud computing, operating systems, performance analysis
-
- Office: CE department, 3rd floor
 - Email: sajavadi@aut.ac.ir
 - Include CE303 in your email subject
 - Home page: <https://ce.aut.ac.ir/~sajavadi/>

Course Introduction

- Saturday and Monday (13:30-15)
 - Attend class on time
- Course web page
 - Check the webpage on regular basis
 - Everything will be posted on CW
 - Post All your Questions on CW Forums
 - ▶ Check forum history before posting any question
- Office hours and TA classes
 - TBD



Textbook

- **Operating System Concepts**, 10th Edition, Wiley publishing
 - By A. Silberschatz, P. Galvin, & G. Gagne
- Other References:
 - Operating systems: design & implementation,
 - ▶ By A. Tanenbaum and A. Woodhull, 3rd edition, 2006.
 - Operating systems: internals and design principles,
 - ▶ By W. Stallings, 5th edition, 2005.



Grading

Section	Score	Considerations
assignments	2.5	five homework
midterm exam	4	1400/08/22
project	4 + 1	in three phases
final exam	8	1400/10/20
quiz	1	two quizzes
class participation	0.5	ask/answer questions be active in the course webpage

Harsh penalty for plagiarism and cheating

Project

- Adding new features to XV6 created in MIT's Operating System Engineering course; isn't this exciting 😊
 - XV6 is used in most of the well-known universities.
 - <https://pdos.csail.mit.edu/6.828/2012/xv6.html>
- Three Phases:
 - Phase 1: getting to know XV6 basics (solo work)
 - Phase 2: getting to know XV6 advanced features (solo work)
 - Phase 3: final project (teamwork)

Syllabus

- Introduction to operating systems
- Process management
 - Threads
 - Synchronization
 - Scheduling
- Memory management
- Storage management
- Protection and security



Copyright Notice

Slides are based on the slides of the main **textbook**.

Silberschatz

<https://www.os-book.com/OS10/slide-dir/index.html>



Part 1

What is an Operating System?

- A **program** that acts as an **intermediary** between a user of a computer and the computer hardware.
 - User can execute programs **conveniently** & **efficiently**
- **Operating system goals:**
 - Execute user programs and make solving user problems easier.
 - Make the computer system convenient to use.
 - Use the computer hardware in an efficient manner.

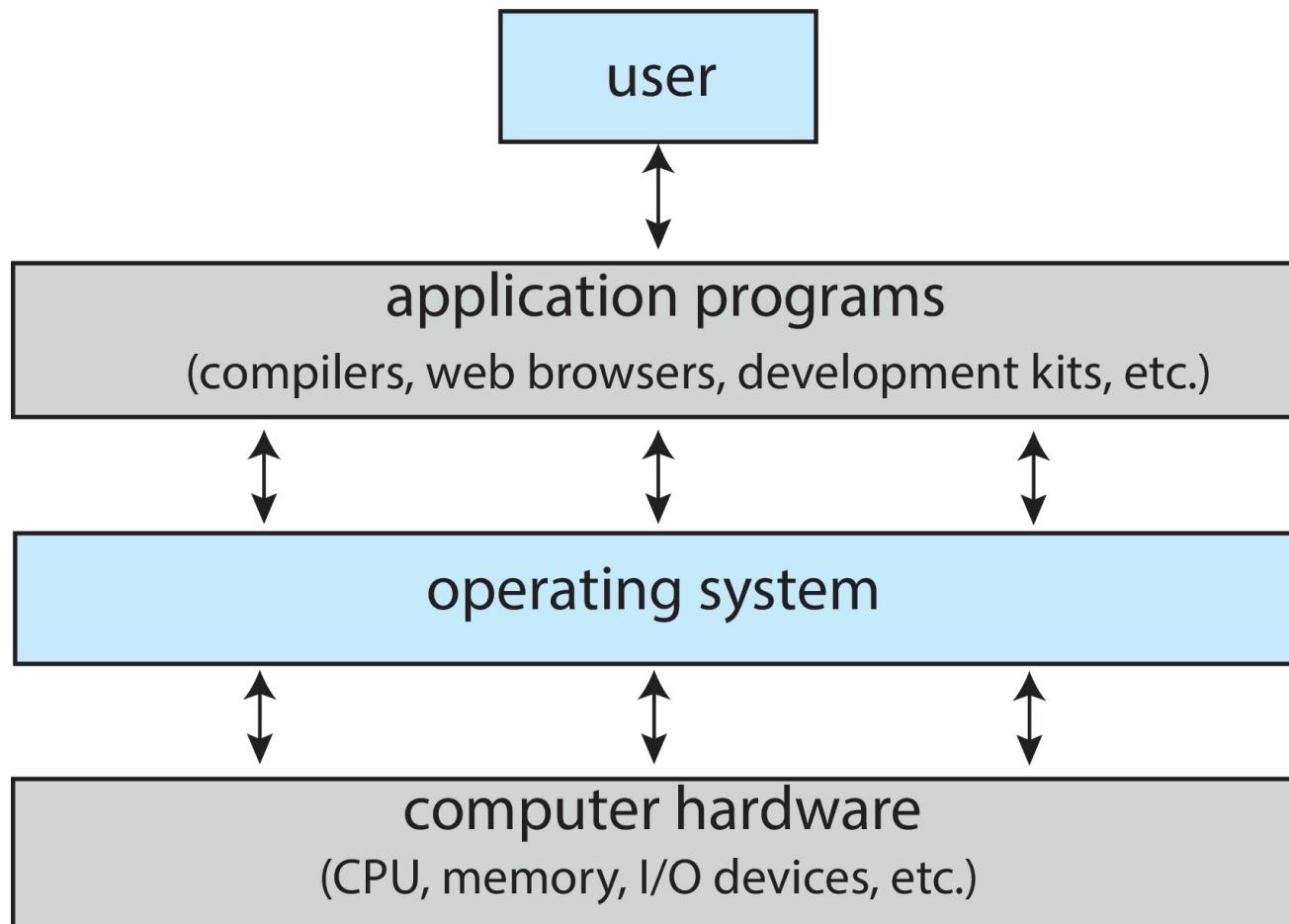


OS: Mandatory or Optional?

- **Can we run a computer without an operating system?**
 - Yes, earliest computers did not have OS.

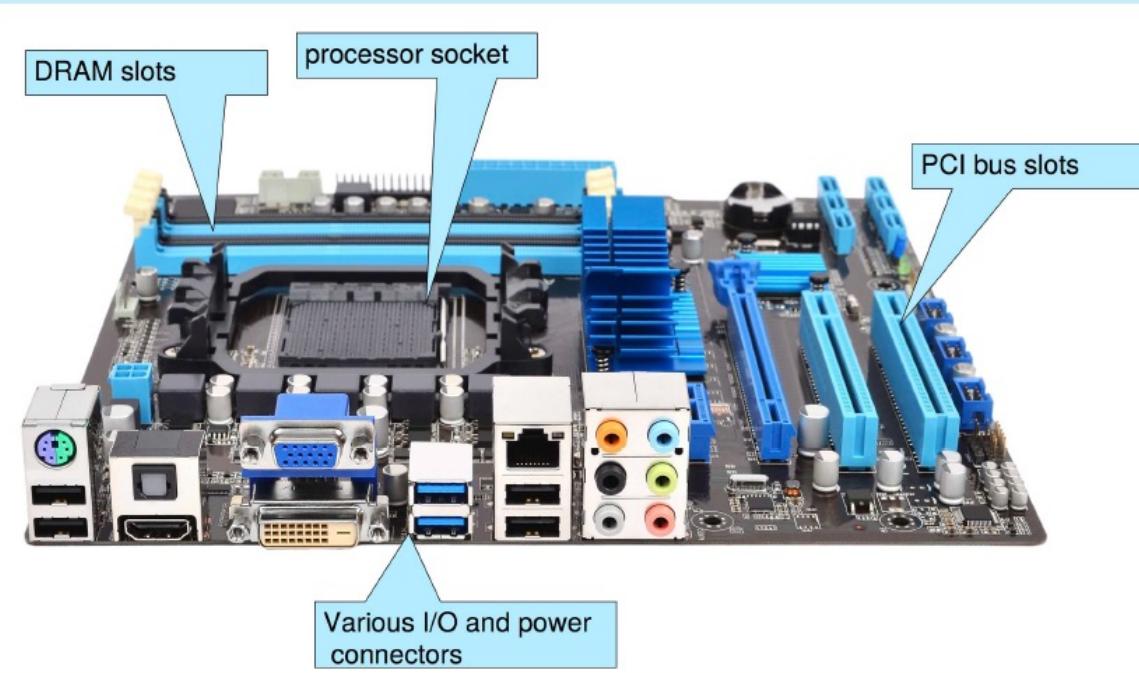
- **What does a compute without an OS look like?**
 - Machines tasked with one program at a time.
 - ▶ Cannot read a pdf while listening to a music.
 - Each program has a lot of work to do.
 - ▶ Where to load a program
 - ▶ IO access

Abstract View of Components of Computer



PC Motherboard

Consider the desktop PC motherboard with a processor socket shown below:



This board is a fully-functioning computer, once its slots are populated. It consists of a processor socket containing a CPU, DRAM sockets, PCIe bus slots, and I/O connectors of various types. Even the lowest-cost general-purpose CPU contains multiple cores. Some motherboards contain multiple processor sockets. More advanced computers allow more than one system board, creating NUMA systems.

Operating System Story

■ Vital goal of a computer system

- Execute user program and make solving user problem easier.

■ Shall user program use hardware directly?

- Hardware alone is **not easy to use**.
- Application programs require certain **common operations**.

تمامی برنامه ها یک سری ویژگی ها و عملیات مشترک

دارند که اینها رو یکبار پیاده کردند تحت عنوان سیستم عامل

Common functions of controlling and allocating resources brought together into one piece called **OS**

مثل کار با کیبورد

در واقع سیستم عامل سخت افزار رو قابل اجرا کرده



Operating System Definition (cont.)

- No universally accepted definition.
- “**The one program running at all times on the computer**” is the **kernel**, part of the operating system.
- Everything else is either
 - A **system program** (ships with the operating system, **but not part of the kernel**) , or Like task manager, MS config, CMD
 - An **application program**, all programs not associated with the operating system. Like fire fox, vpn

یک تعریف دیگه: یک لایه روی سخت افزار که کار با اون رو ساده میکنه و پیچیدگی هاشو پنهان میکنه

اولین برنامه ای که وقتی کامپیوتر رو روشن میکنیم اجرا میشه

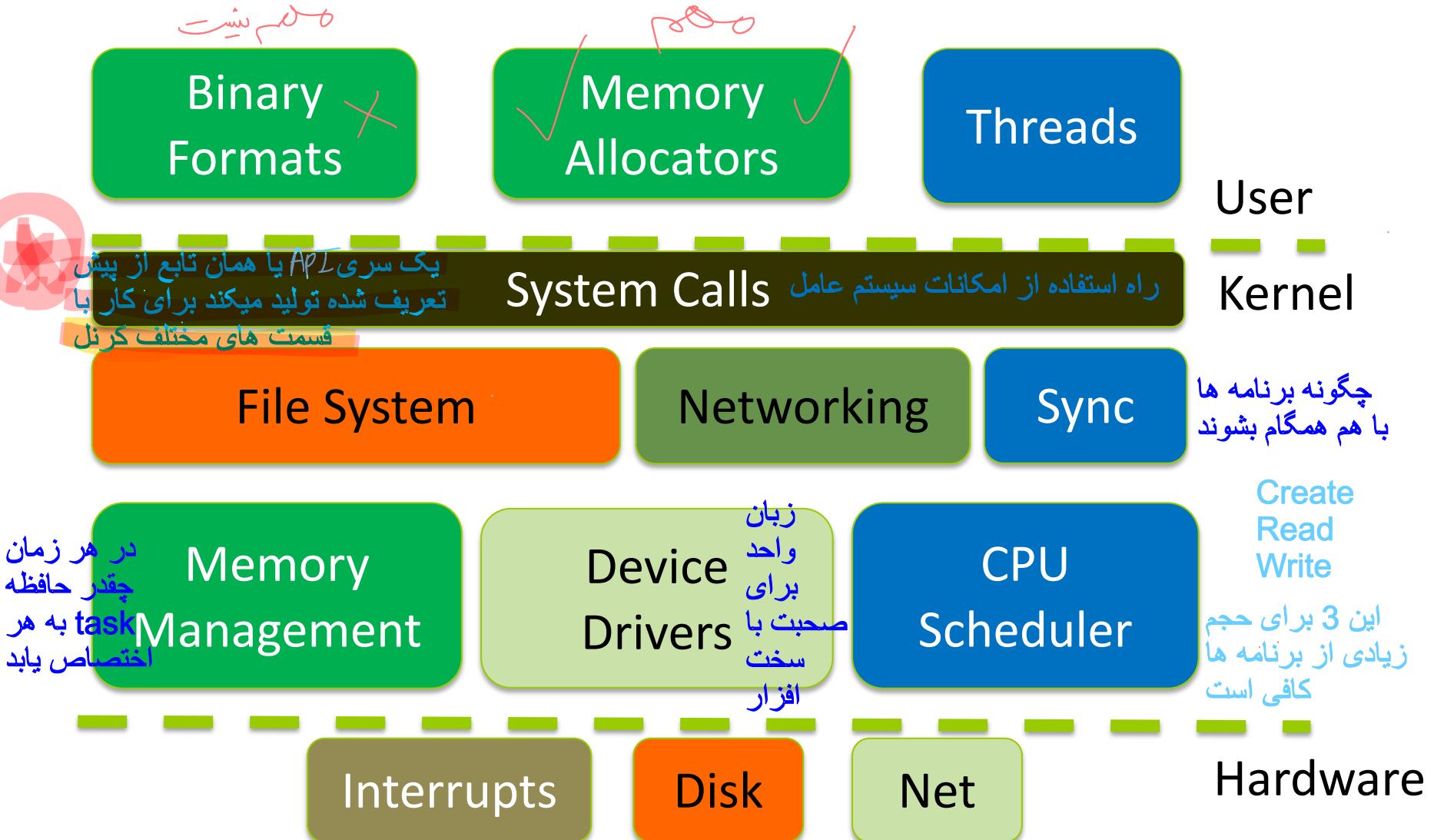
کرنل مهم ترین بخش سیستم عامل است

کرنل خودش داره همه چیو اجرا میکنه

برنامه هایی که همراه با سیستم عامل نصب میشن



A logical view of the OS



Computer Startup

BootHeader با

■ Bootstrap program is loaded at power-up or reboot.

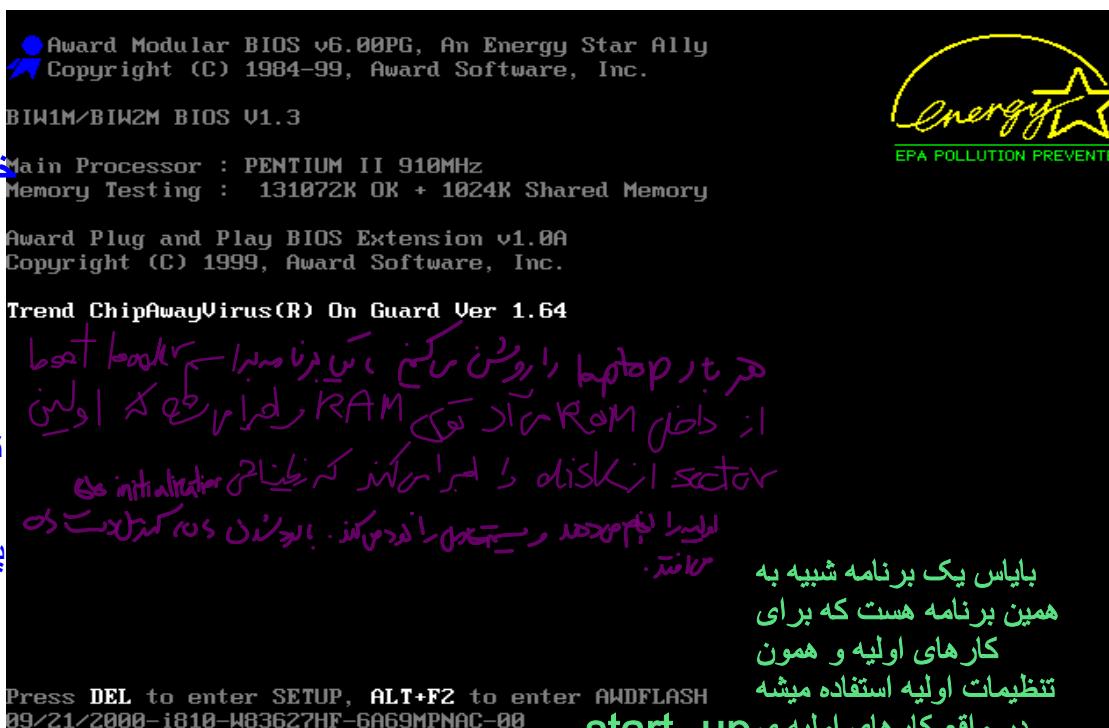
- Typically stored in ROM or EPROM, generally known as **firmware**.
- Initializes all aspects of system.
- Loads operating system **kernel** and starts execution.

اولین برنامه ای که اجرا میشے و میاد بخش اولیه از 28 میلیون خط کد سیستم عامل رو لود میکنه داخل حافظه

وظیفه‌ی این بخش آغازین این است که همه‌ی بخش‌ها را مقدار دهی اولیه می‌کند. (**initialize**)

تمام ابعاد سیستم عامل را لود میکند داخل حافظه

وقتی دو تا سیستم عامل نصب میکنیم تری قسمت **Bootstrap** میاد میینه دو تا سیستم عامل داره یعنی میاد داخل **Bootsector** میینه اطلاعات دو تا سیستم عامل نوشته شده است.



Bootstr
ap

تمامی این خطوط رو همین برنامه‌ی برای ما نوشته است

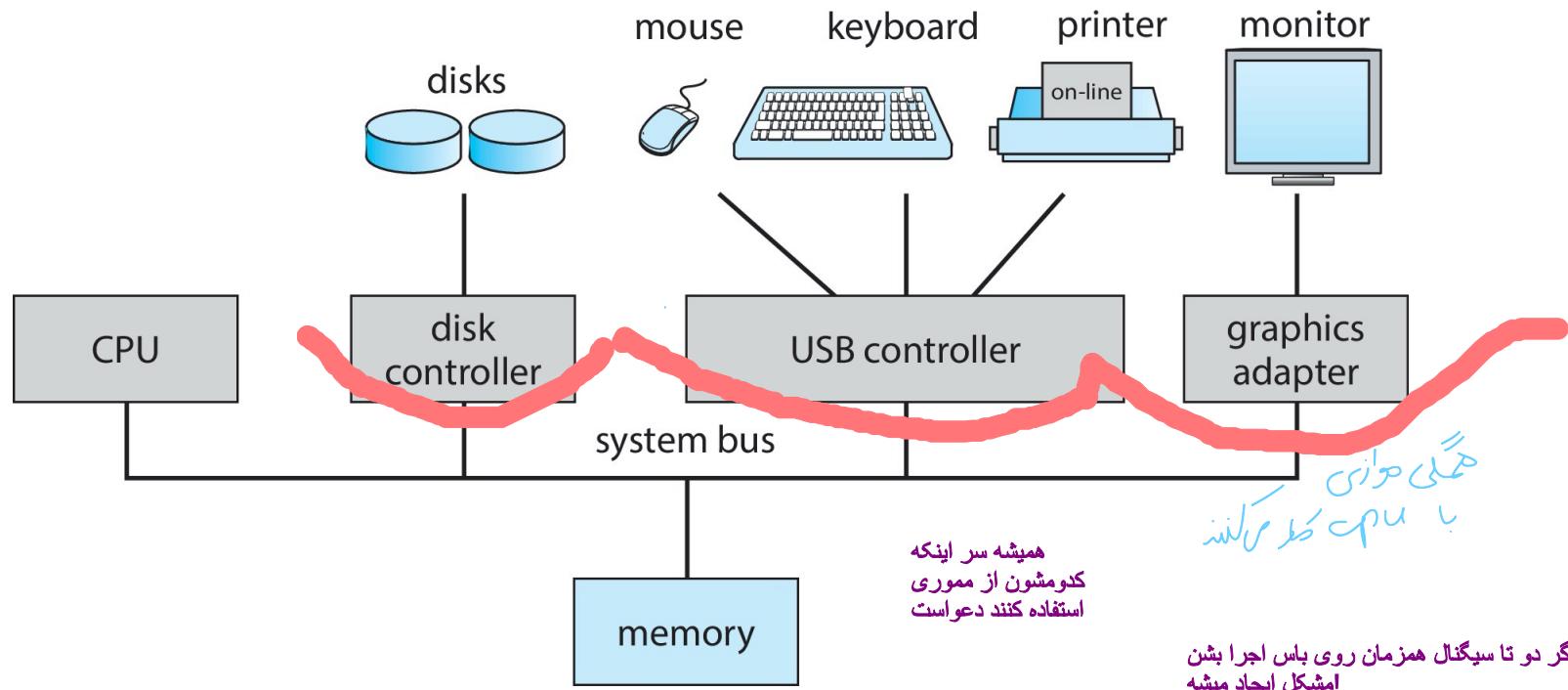
تمامی اون کارهای اولیه مثل انتخاب منطقه، چند پارتبیشنی و ... را همه‌را همین خطوط اولیه انجام میده



Computer System Organization

■ Computer-system operation

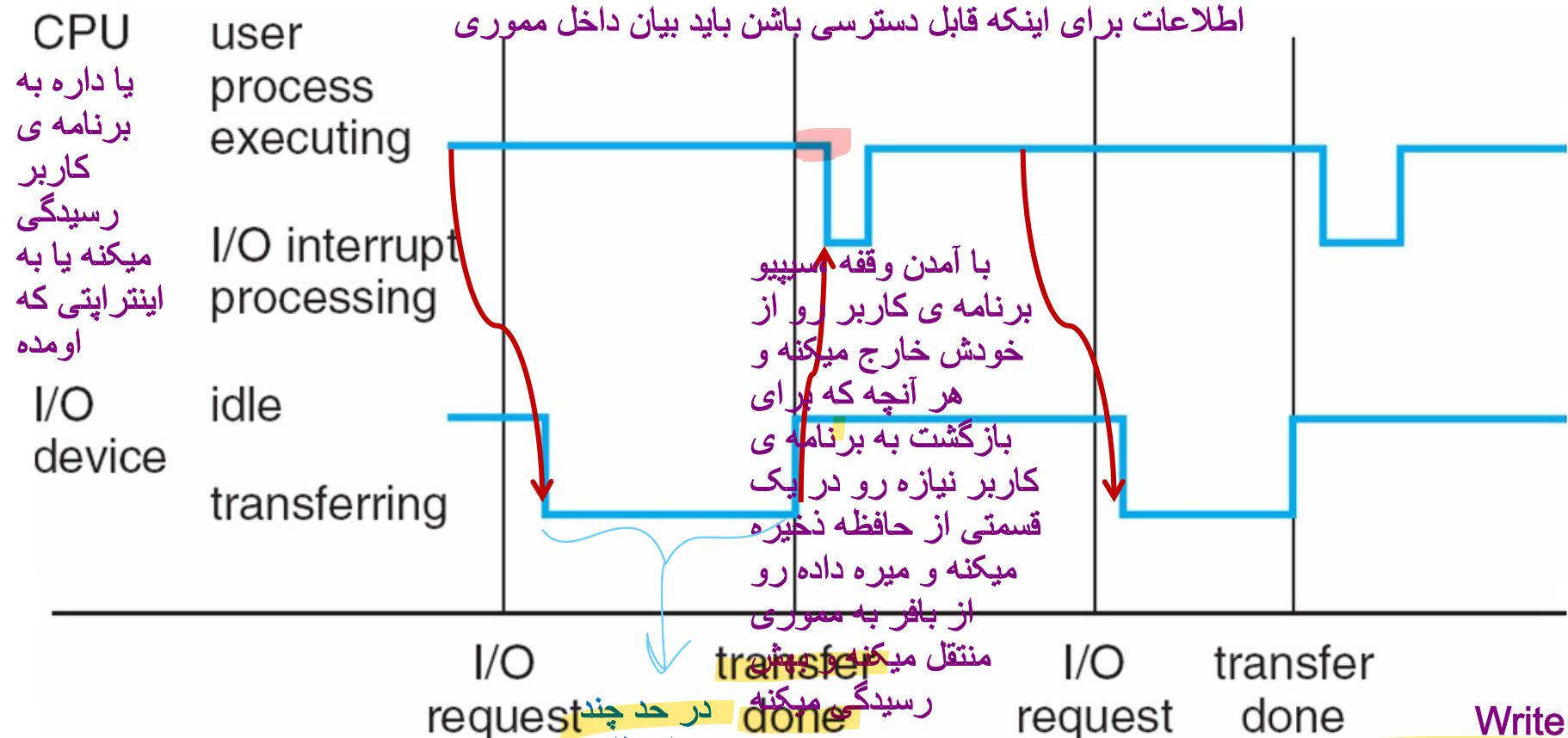
- One or more CPUs, device controllers connect through common **bus** providing access to shared memory.
- Concurrent execution of CPUs and devices competing for memory cycles.



Computer-System Operation

- Each device controller is in charge of a particular device type (e.g., disk drives, audio devices).
 صهاریج کنترلر دارهای بلکه صدای
- Each device controller has a local buffer.
 چهارتا رجیستر
 یک سخت افزاری که
 اطلاعات روشنخیره
 میشوند
- I/O devices and the CPU can execute **concurrently**
 هم روند
 بهتره اینجا از Parallel استفاده کنیم
- I/O: device \leftrightarrow local buffer of controller.
 مثل آنتقال اینکه یک دکمه کیبورد رو فشار میدم و تبدیل
 میشه به یک تکه کد و داخل بافر ذخیره میشه
- Device controller informs CPU that it has finished its operation by causing an **interrupt**.
 با استفاده از این به سیپیو میگه یه اتفاقی افتاده و من یه
 کاری باهات دارم
- CPU moves data
 - Main memory \leftrightarrow local buffers

Interrupt Timeline

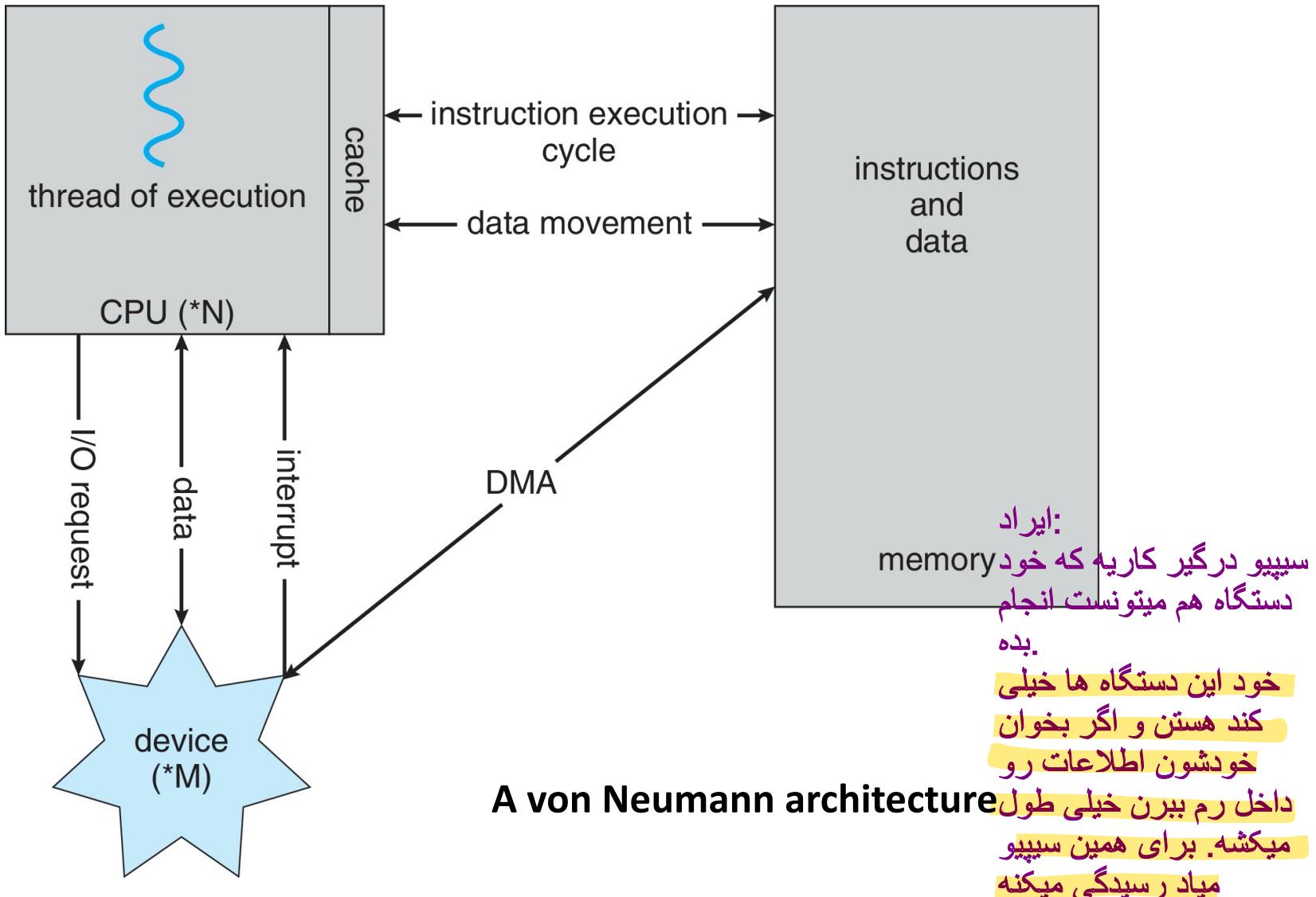


مثلًا کیبورد رو که فشار ميديم، اطلاعات تى چند نانو ثانیه منتقل ميشه به بافر

در نوشتن، سپيبيو عبارت رو از مموری منتقل ميکنه به بافر و سپس از ديسيک كنترلر ميخداد که اينها رو منتقل کنه به ديسيك و وقتی ديسيك كنترلر نوشتن رو تموم کرد دوباره با استفاده از اينتراپت به سپيبيو ميگه کارم تموم شد. همه چيز موازي



How a Modern Computer Works



Direct Memory Access Structure

- Used for **high-speed I/O devices** able to transmit information at close to memory speeds.
- Device controller transfers blocks of data from buffer storage directly to main memory **without CPU intervention.**
- Only one interrupt is generated per block**, rather than the one interrupt per byte.

صرف اینتراپت ایجاد میکنه برای اطلاع دادن ولی هر بلاک یکبار نه هر بایت یکبار! که باعث میشه سیپیو کمتر درگیر باشے

این باعث میشه سیپیو میتونه کارهای دیگه ای بکنه و زمان بهینه بشه. ولی اگر باس داده در دسترس

یک دستگاه باشه، سیپیو باید بره به جنگ دستگاه برای استفاده از باس

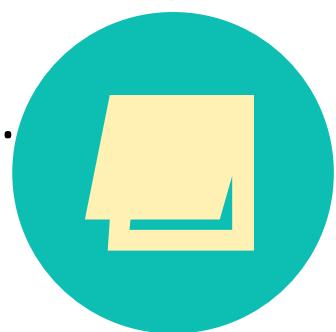


Multiprogramming (Batch System)

اسکرین شات های
25 و 26 و 28 خونده

پشن

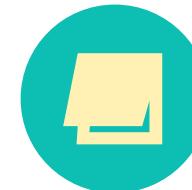
- Single user cannot always keep CPU and I/O devices busy.
- Multiprogramming organizes jobs (code and data) so CPU always has one to execute.
- A subset of total jobs in system is kept in memory.
- One job selected and run via **job scheduling**.
- When job has to wait (for I/O for example), OS switches to another job.



توضیح اصلی

Multitasking (Timesharing)

- A logical extension of Batch systems
- The CPU **switches jobs so frequently** that users can interact with each job while it is running, creating **interactive** computing.



اینو بخون!!!!!!!

- Response time should be < 1 second.
- Each user has at least one program executing in memory \Rightarrow process.
- If several jobs ready to run at the same time \Rightarrow CPU scheduling.
- If processes don't fit in memory, **swapping** moves them in&out to run.
- **Virtual memory** allows execution of processes not completely in memory.



Dual-mode Operation

- Dual-mode operation allows OS to protect itself and other system components.

هر سیپیو حداقل باید 2 حالت
داشته باشه!

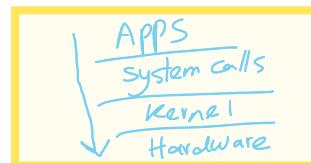
- User mode and kernel mode
- در داخل سیپیو یک بیت وجود داره که اگر 1 باشه یعنی user mode و اگر 0 باشه یعنی kernel mode
- Mode bit provided by hardware
- بیت به صورت سخت افزاری است
- Provides ability to distinguish when system is running user code or kernel code.
 - When a user is running \Rightarrow mode bit is “user”.
 - When kernel code is executing \Rightarrow mode bit is “kernel”.

اسکرین شات 29 خونده بشه



Dual-mode Operation (Cont.)

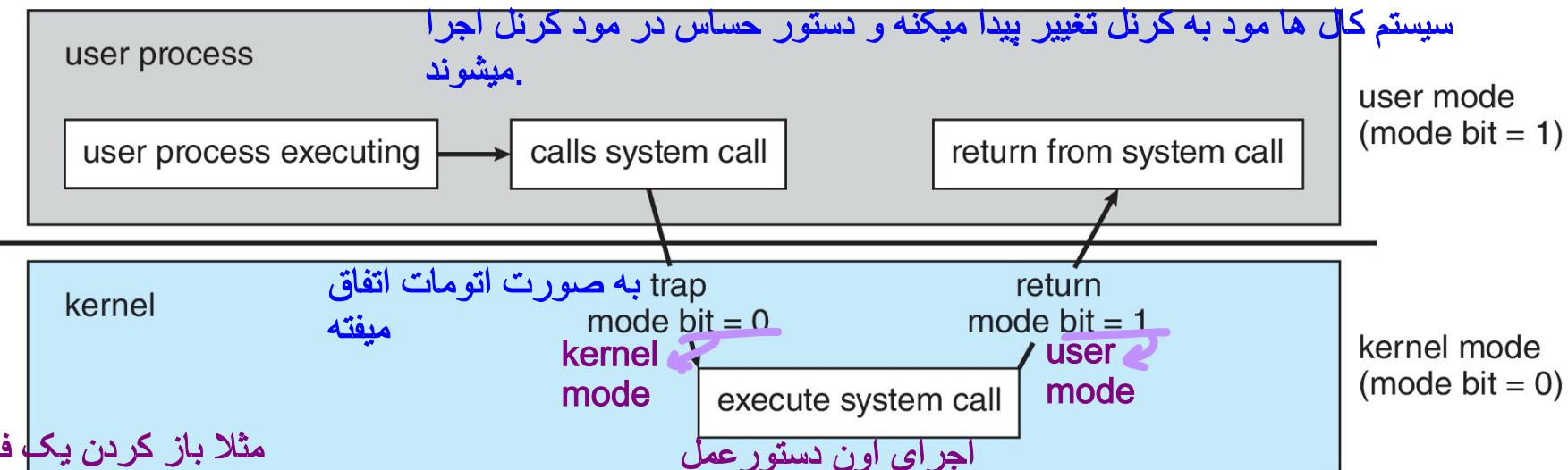
- How do we guarantee that user does not explicitly set the mode bit to “kernel”?



- System call changes mode to kernel, return from call resets it to user.

یک وقفه‌ی نرم افزاری

اگر دستور حساسی بخواهد در مود کاربر اجرا بشن، جلوی انها گرفته میشه و توسط سیستم کال ها مود به کرنل تغییر پیدا میکنه و دستور حساس در مود کرنل اجرا میشوند.



مثلا باز کردن یک فایل
یک دستور حساس است

system call ha dar kenar
bit mode baes mishan
system az khodesh
moraghebat kone

امکان نداره به
صورت دستی
بتوئیم مود بیت رو
!تغییر بدیم

Privileged instructions

- Some instructions designated as **privileged**, **only executable in kernel mode**.
 - Example: **I/O control, timer management, and interrupt management**

If an attempt is made to execute a privileged instruction in user mode



The hardware ***does not execute the instruction*** but rather treats it as ***illegal*** and ***traps*** it to the ***operating system***.

معمولاً کرش میکنه. یعنی سیتم عامل میبندشون و اجرای برنامه متوقف میشه

Questions?

