OPERATING SYSTEMS

Input - Output Management and Security - 2

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Transforming I/O Requests to Hardware Operations, Device interaction, device driver, buffering

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

OPERATING SYSTEMS

Course Syllabus - Unit 5

10 Hours

Unit-5:Unit 5: IO Management and Security

I/O Hardware, polling and interrupts, DMA, Kernel I/O Subsystem and Transforming I/O Requests to Hardware Operations - Device interaction, device driver, buffering System Protection: Goals, Principles and Domain of Protection, Access Matrix, Access control, Access rights. System Security: The Security Problem,Program Threats, System Threats and Network Threats. Case Study: Windows 7/Windows 10

OPERATING SYSTEMS

Course Outline

47	I/O Hardware, polling and interrupts	13.1,13.2
48	DMA	13.2.3
49	Transforming I/O Requests to Hardware Operations, Device interaction, device driver, buffering.	13.5
50	Goals, Principles and Domain of Protection	14.1-14.3
51	Access Matrix	14.4
52	Access control, Access rights	14.5-14.7
53	The Security Problem	15.1
54	Program Threats	15.2
55	System Threats and Network Threats	15.3
56	Case Study : Windows File System	17.5

- # Transforming I/O Requests to Hardware Operations

Transforming I/O Requests to Hardware Operations

- Users request data using file names, which must ultimately be mapped to specific blocks of data from a specific device managed by a specific device driver.

- DOS uses the colon separator to specify a particular device (e.g. C:, LPT:, etc.)

- UNIX uses a mount table to map filename prefixes (e.g. /usr) to specific mounted devices.

- Where multiple entries in the mount table match different prefixes of the filename the one that matches the longest prefix is chosen. (e.g. /usr/home instead of /usr where both exist in the mount table and both match the desired file.)

Transforming I/O Requests to Hardware Operations

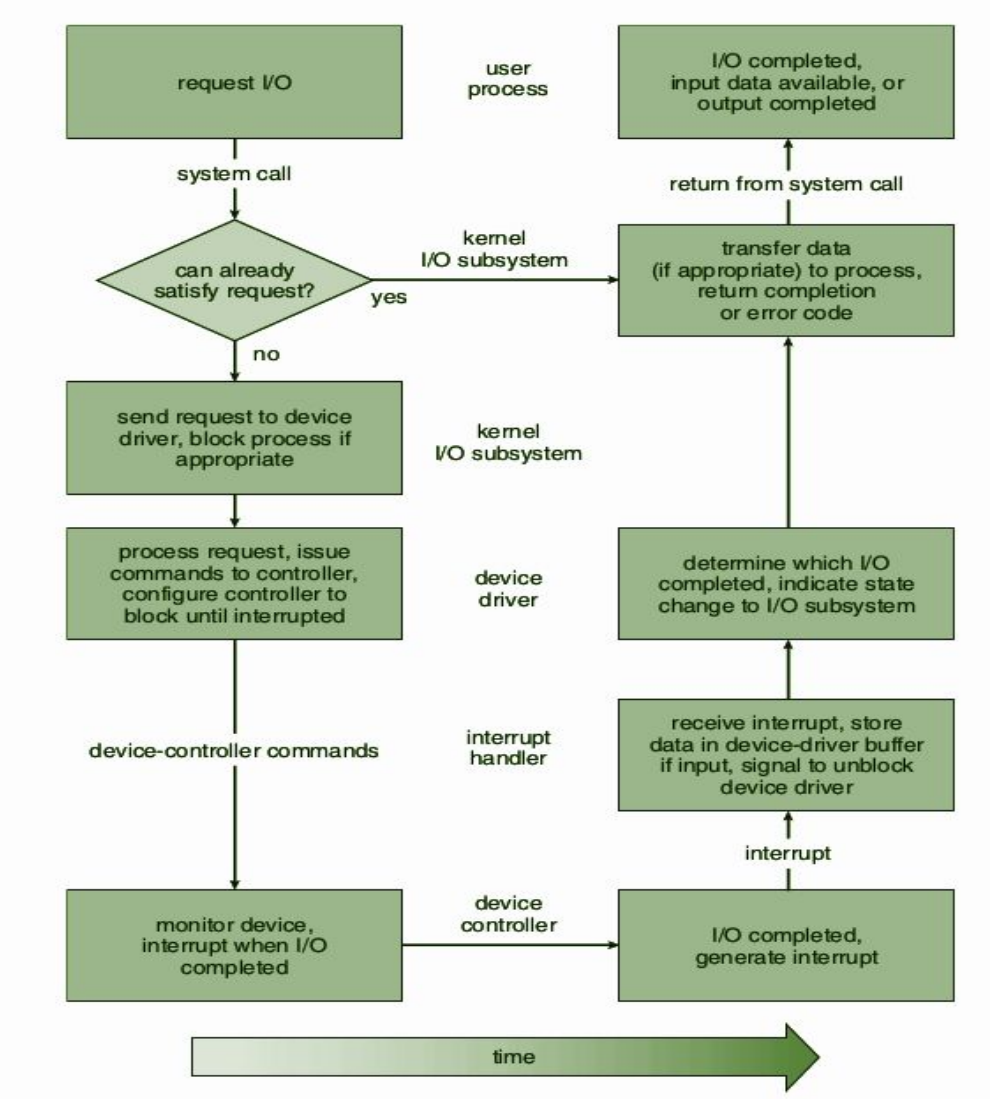
- UNIX uses special device files, usually located in /dev, to represent and access physical devices directly.
    - Each device file has a major and minor number associated with it, stored and displayed where the file size would normally go.
    - The major number is an index into a table of device drivers, and indicates which device driver handles this device. (E.g. the disk drive handler.)
    - The minor number is a parameter passed to the device driver, and indicates which specific device is to be accessed, out of the many which may be handled by a particular device driver. (e.g. a particular disk drive or partition.)
- A series of lookup tables and mappings makes the access of different devices flexible, and somewhat transparent to users.

Transforming I/O Requests to Hardware Operations

- Modern operating systems gain significant flexibility from the multiple stages of lookup tables in the path between a request and a physical device controller.
- The mechanisms that pass requests between applications and drivers are general.
- Thus, we can introduce new devices and drivers into a computer without recompiling the kernel. In fact, some operating systems have the ability to load device drivers on demand.
- At boot time, the system first probes the hardware buses to determine what devices are present.
- It then loads in the necessary drivers, either immediately or when first required by an I/O request.

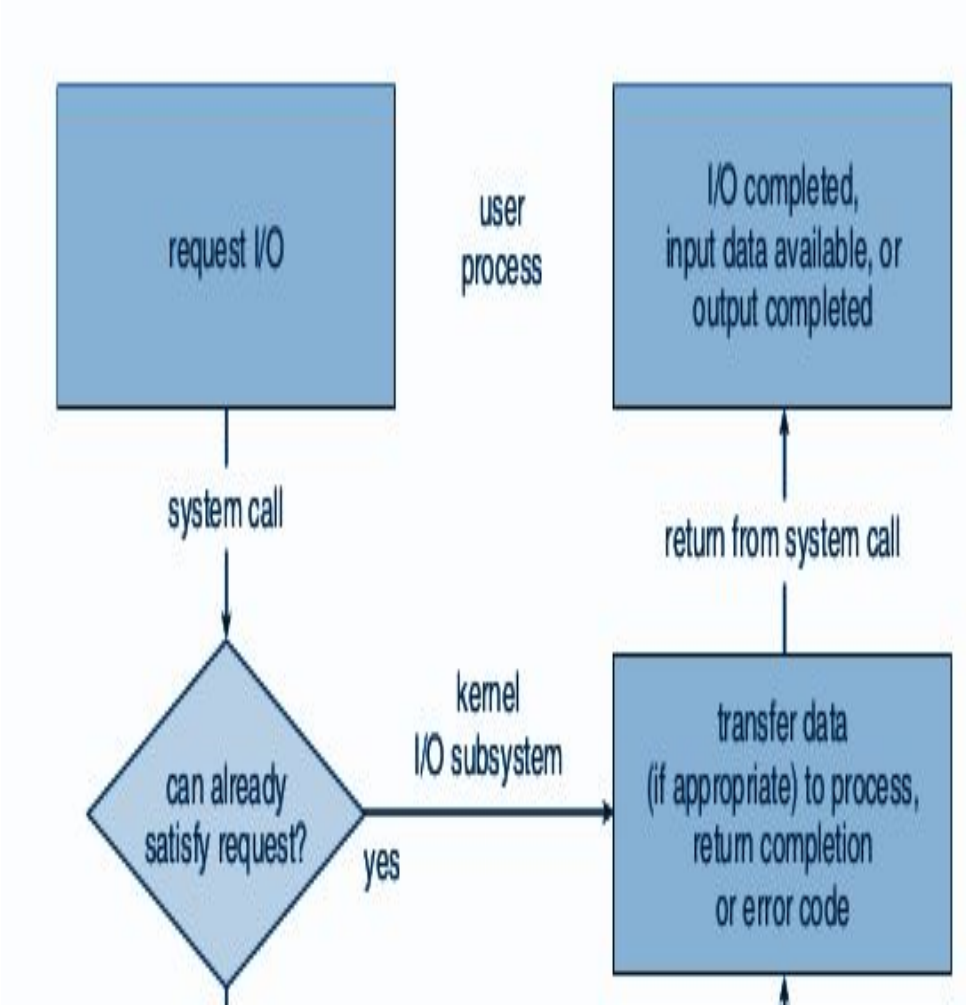
Transforming I/O Requests to Hardware Operations

- The figure suggests that an I/O operation requires a great many steps that together consume a tremendous number of CPU cycles.
- A **blocking read** will wait until there is data available (or a timeout, if any, expires), and then returns from the function call.
- A **non-blocking read** will (or at least should) always return immediately, but it might not return any data, if none is available at the moment



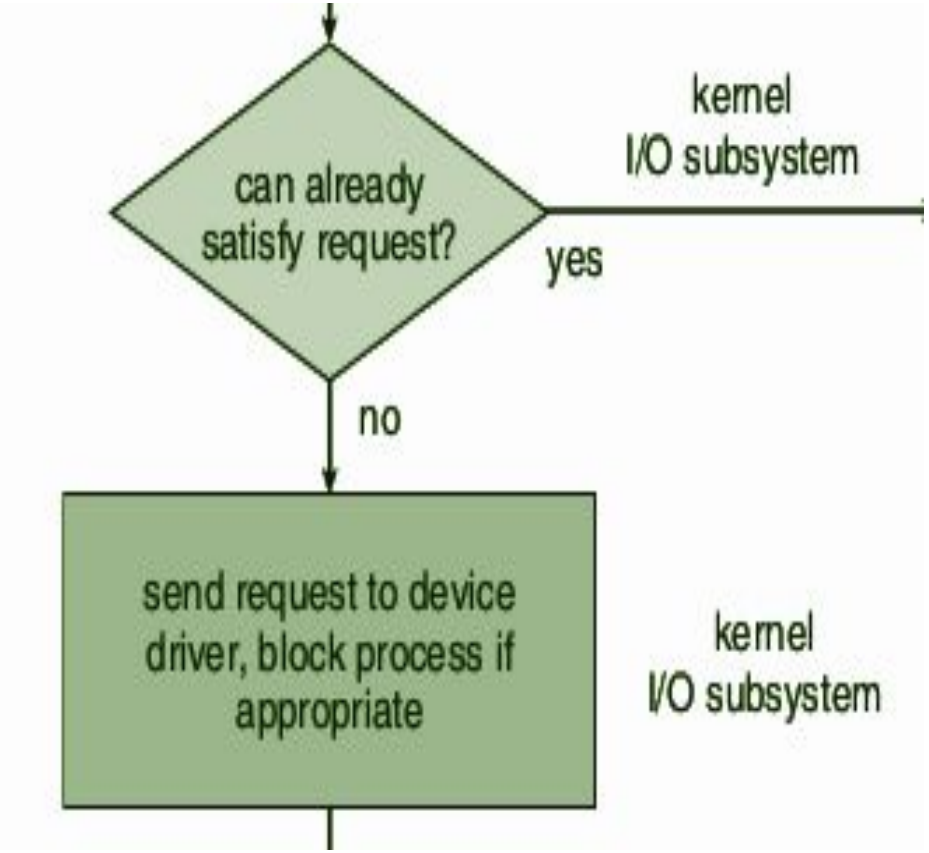
Transforming I/O Requests to Hardware Operations

- A process issues a blocking read() system call to a file descriptor of a file that has been opened previously.
- The system-call code in the kernel checks the parameters for correctness.
- In the case of input, if the data are already available in the buffer cache, the data are returned to the process, and the I/O request is completed.



Transforming I/O Requests to Hardware Operations

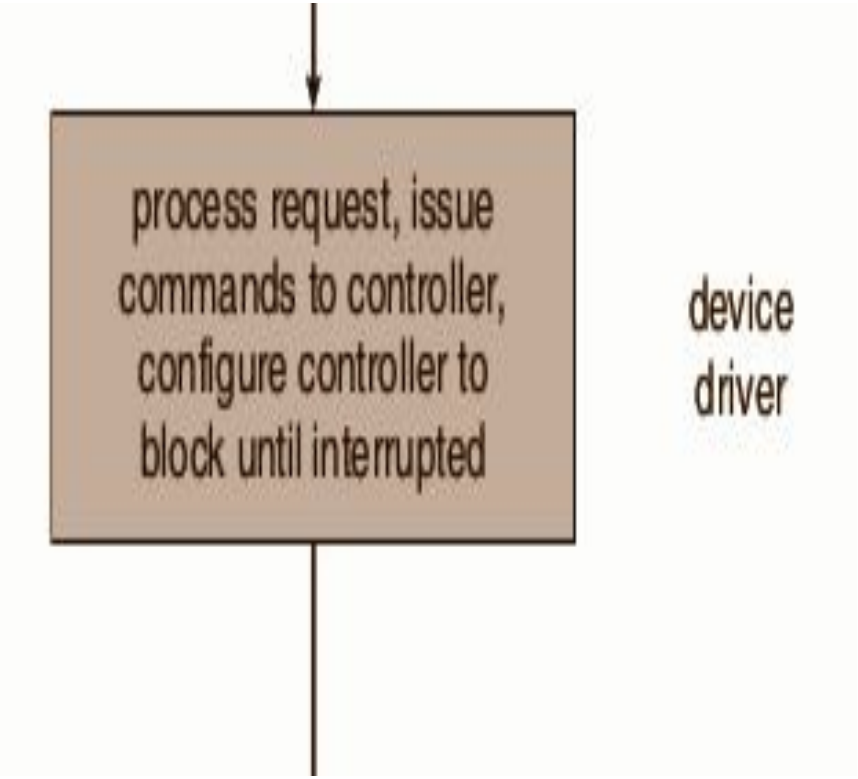
- Otherwise, a physical I/O must be performed.
- The process is removed from the run queue and is placed on the wait queue for the device, and the I/O request is scheduled.
- Eventually, the I/O subsystem sends the request to the device driver.
- Depending on the operating system, the request is sent via a subroutine call or an in-kernel message.



Transforming I/O Requests to Hardware Operations

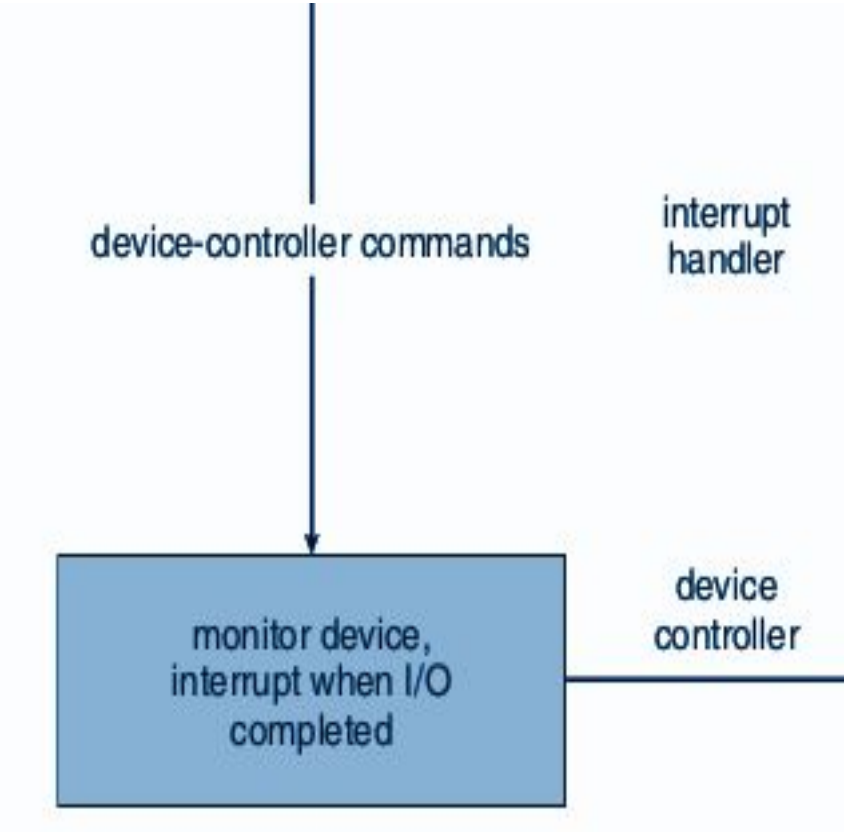
- The device driver allocates kernel buffer space to receive the data and schedules the I/O .

- Eventually, the driver sends commands to the device controller by writing into the device-control registers.



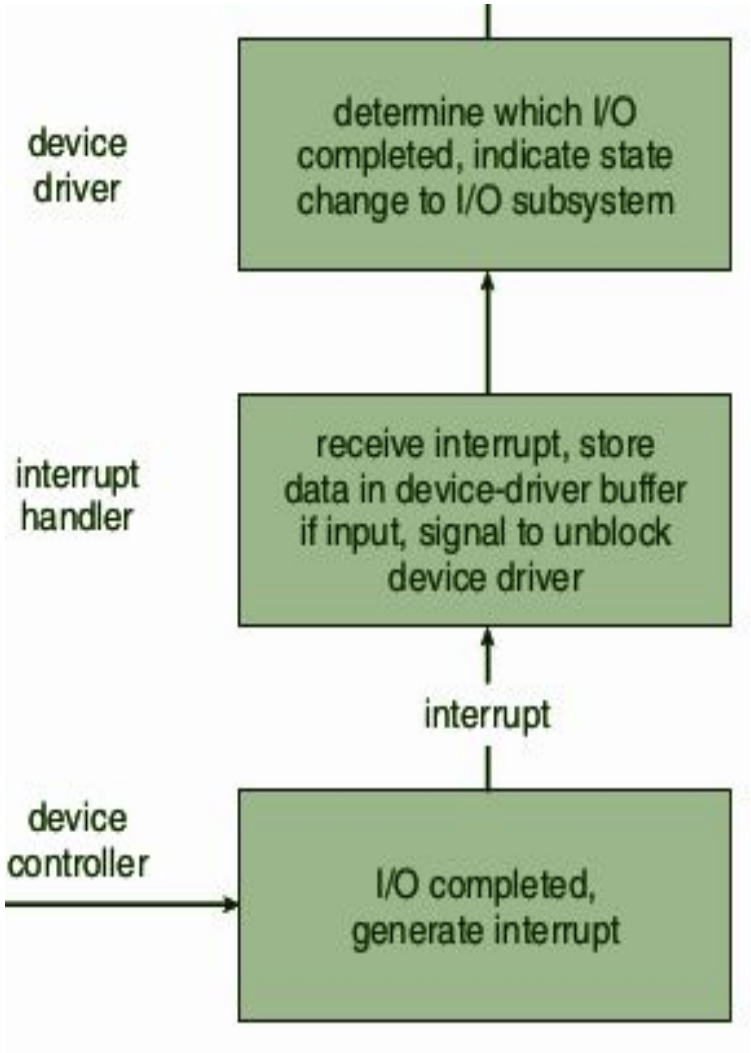
Transforming I/O Requests to Hardware Operations

- The device controller operates the device hardware to perform the data transfer.
- The driver may poll for status and data, or it may have set up a DMA transfer into kernel memory.
- We assume that the transfer is managed by a DMA controller, which generates an interrupt when the transfer completes.



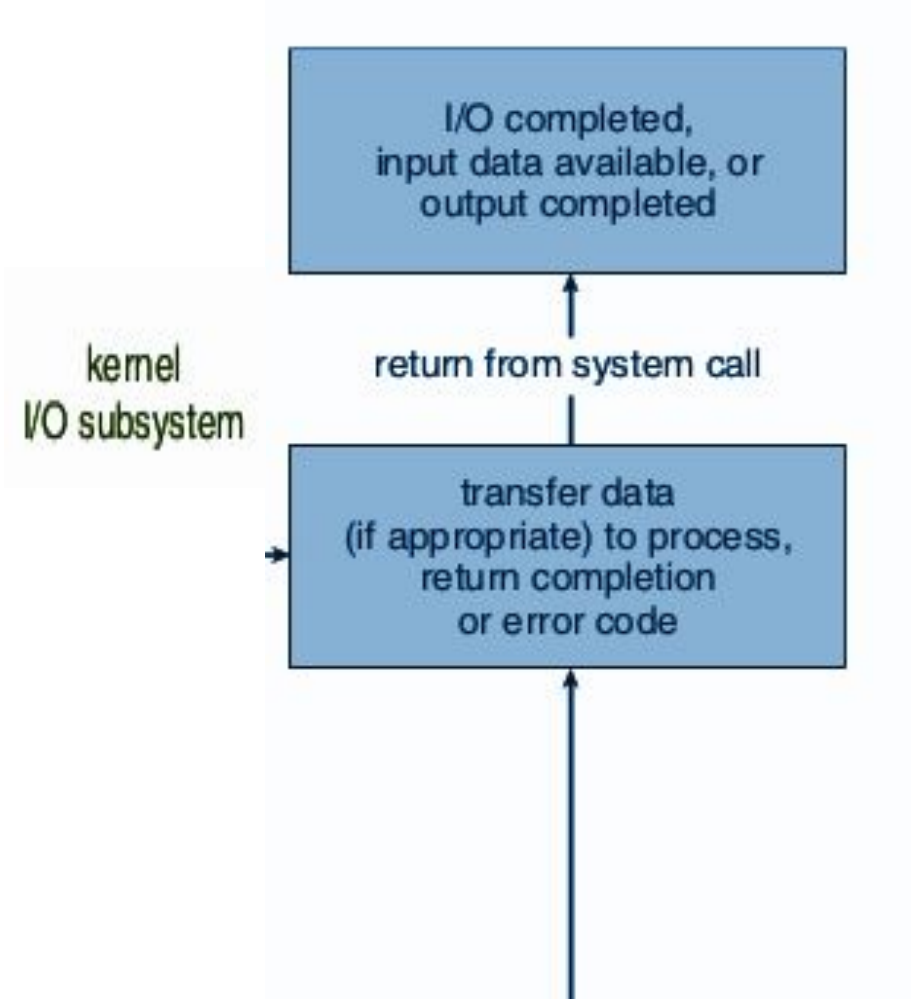
Transforming I/O Requests to Hardware Operations

- The kernel transfers data or return codes to the address space of the requesting process and moves the process from the wait queue back to the ready queue.

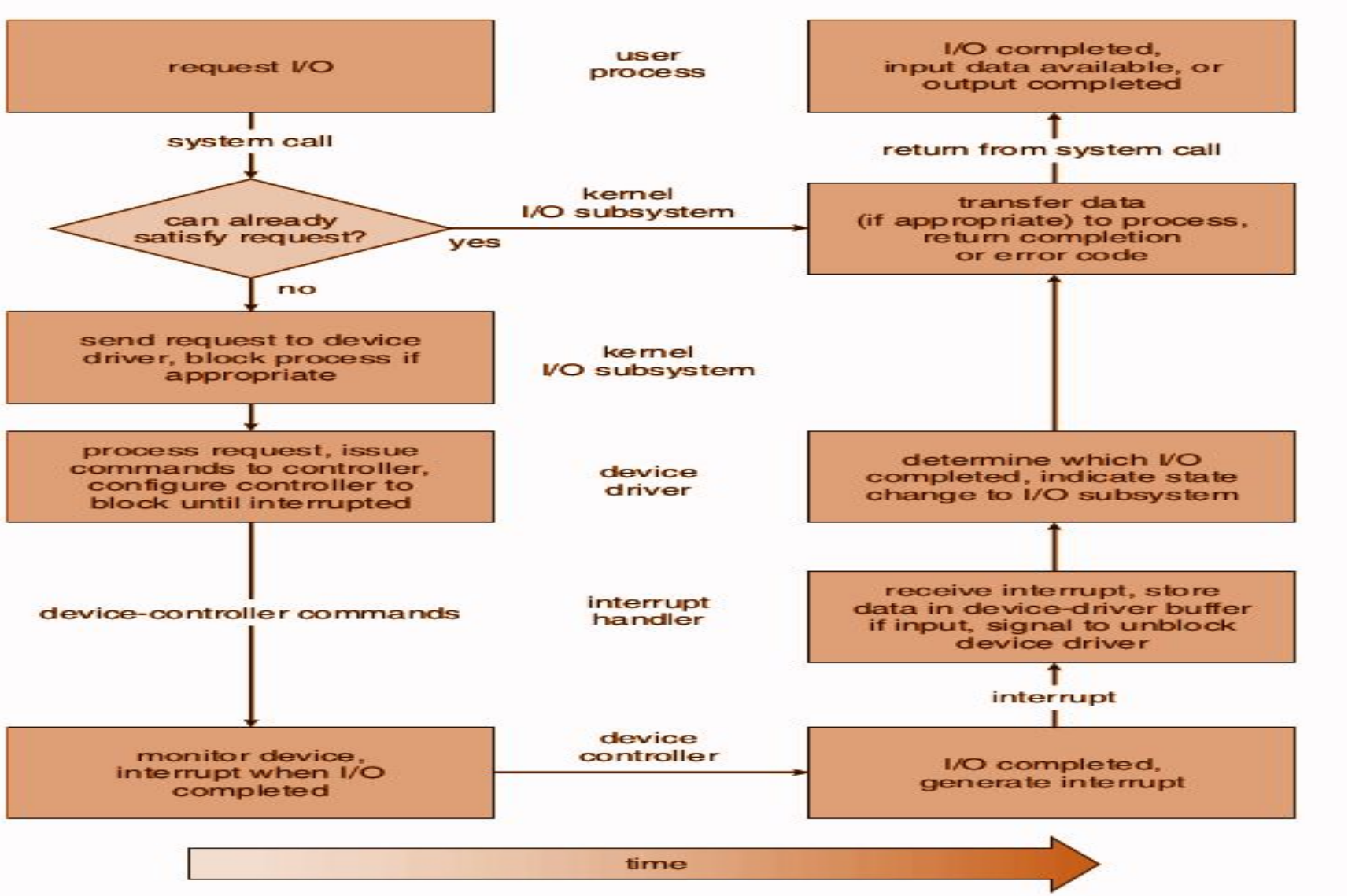


Transforming I/O Requests to Hardware Operations

- Moving the process to the ready queue unblocks the process.
- When the scheduler assigns the process to the CPU , the process resumes execution at the completion of the system call.



Transforming I/O Requests to Hardware Operations



- Transforming I/O Requests to Hardware Operations

THANK YOU

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

nitin.pujari@pes.edu

For Course Deliverables by the Anchor Faculty click on www.pesuacademy.com