EECS 111:

System Software

Lecture: I/O Systems

Prof. Mohammad Al Faruque

The Henry Samueli School of Engineering
Electrical Engineering & Computer Science
University of California Irvine (UCI)

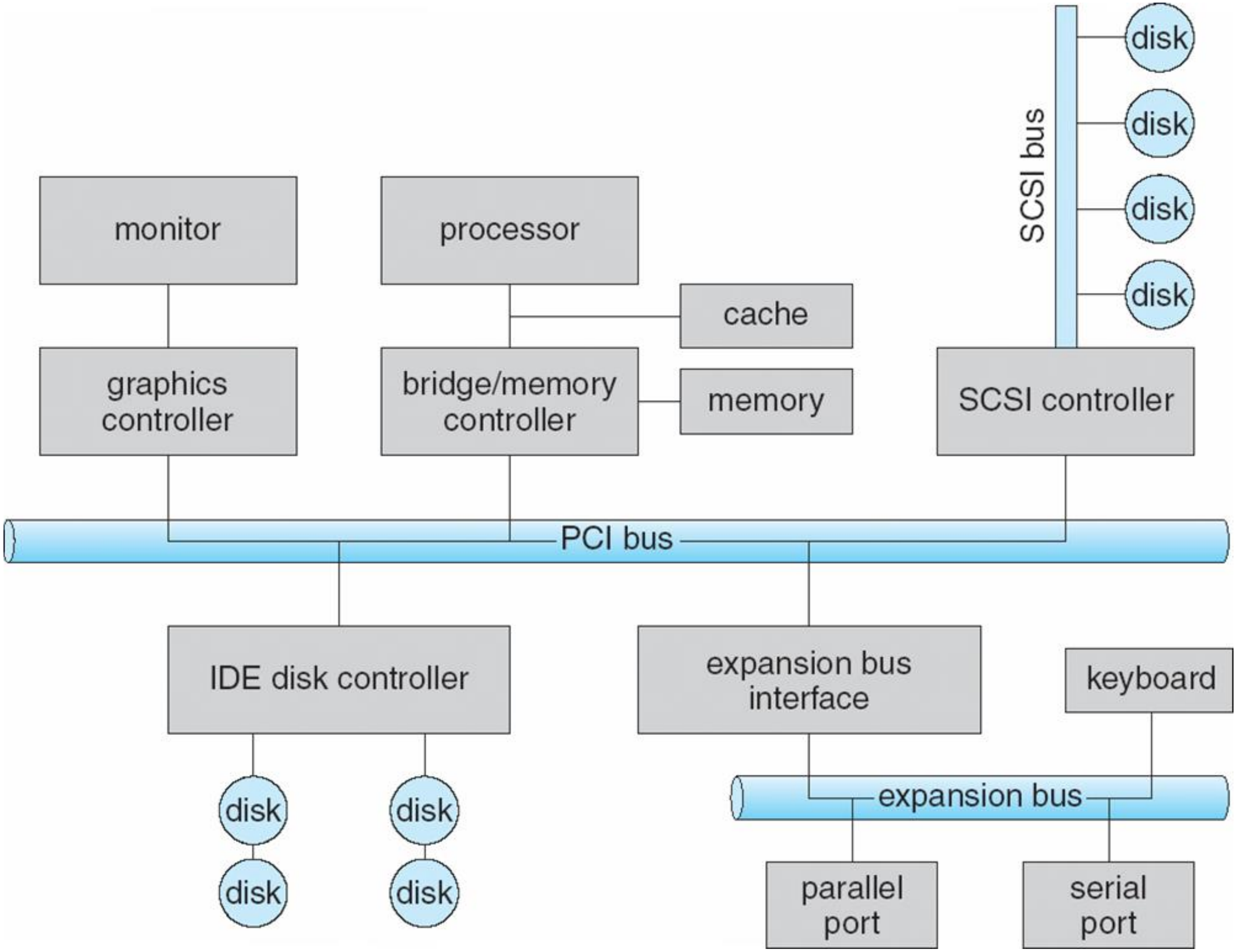
I/O Systems

- I/O Hardware
- Application I/O Interface
- Kernel I/O Subsystem
- Transforming I/O Requests to Hardware Operations

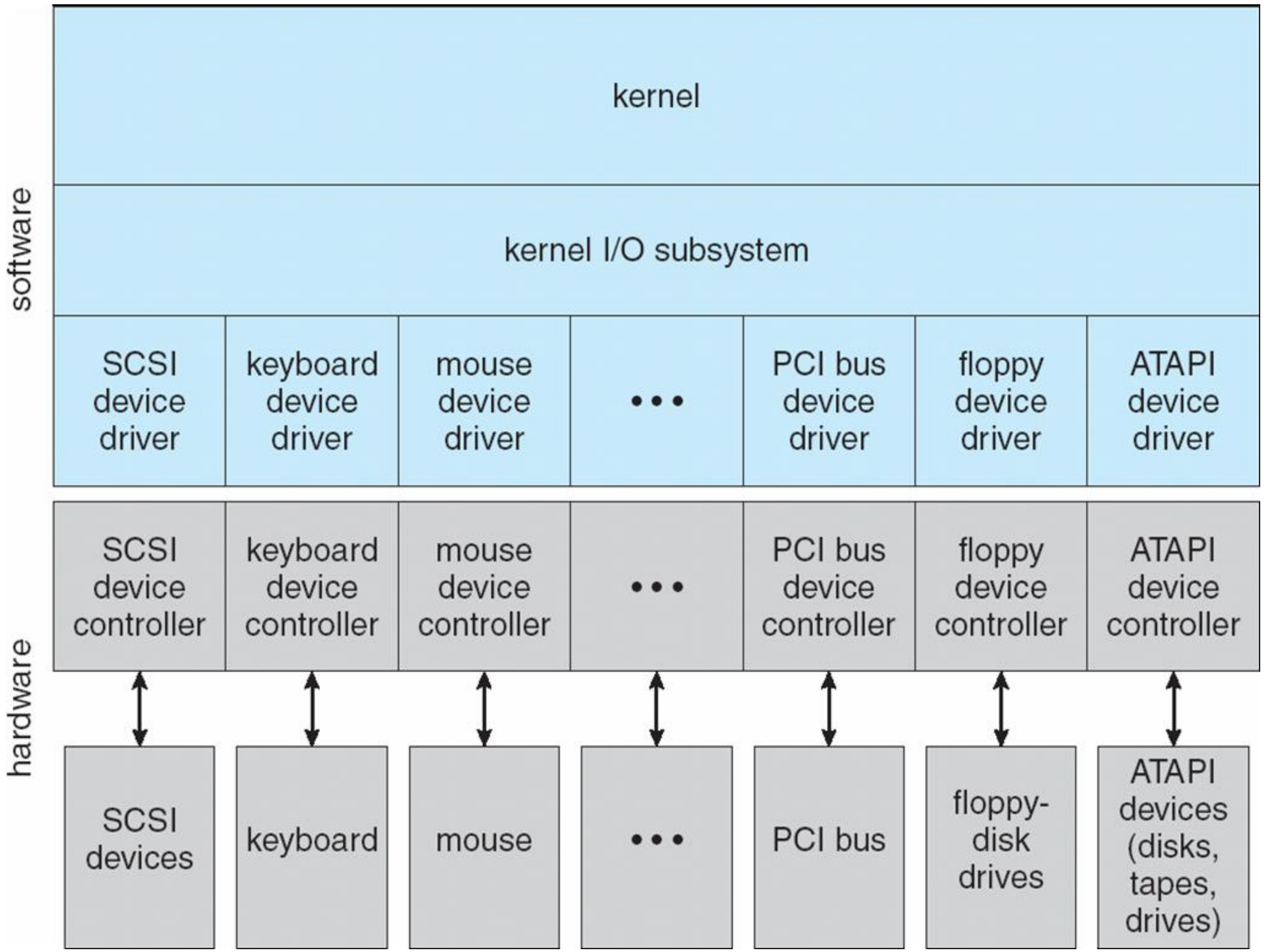
Overview

- I/O management is a major component of operating system design and operation
  - Important aspect of computer operation
  - I/O devices vary greatly
  - Various methods to control them
  - Performance management
  - New types of devices frequent
- Ports, busses, device controllers connect to various devices
- Device drivers encapsulate device details
  - Present uniform device-access interface to I/O subsystem

A Typical PC Bus Structure



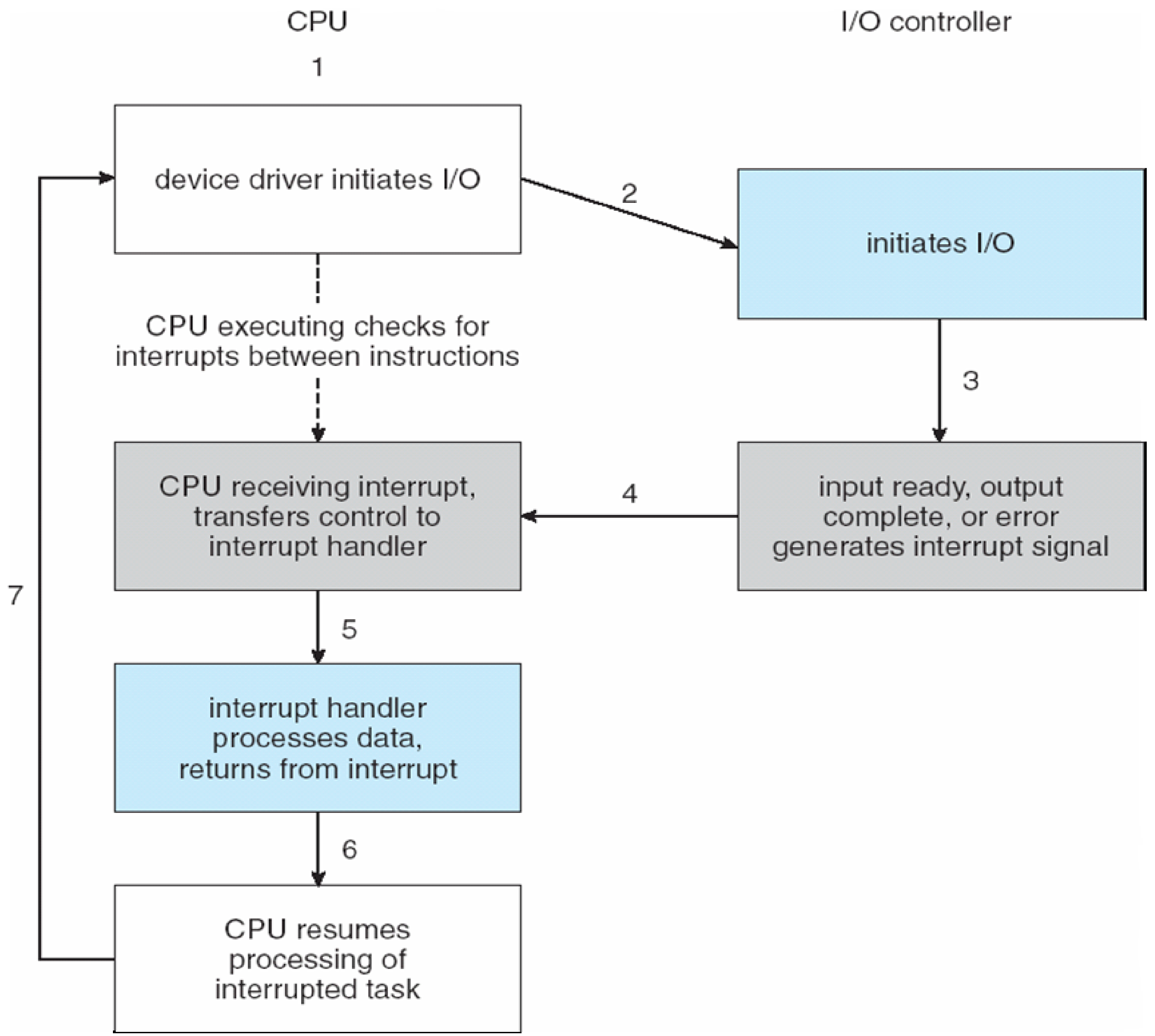
A Kernel I/O Structure



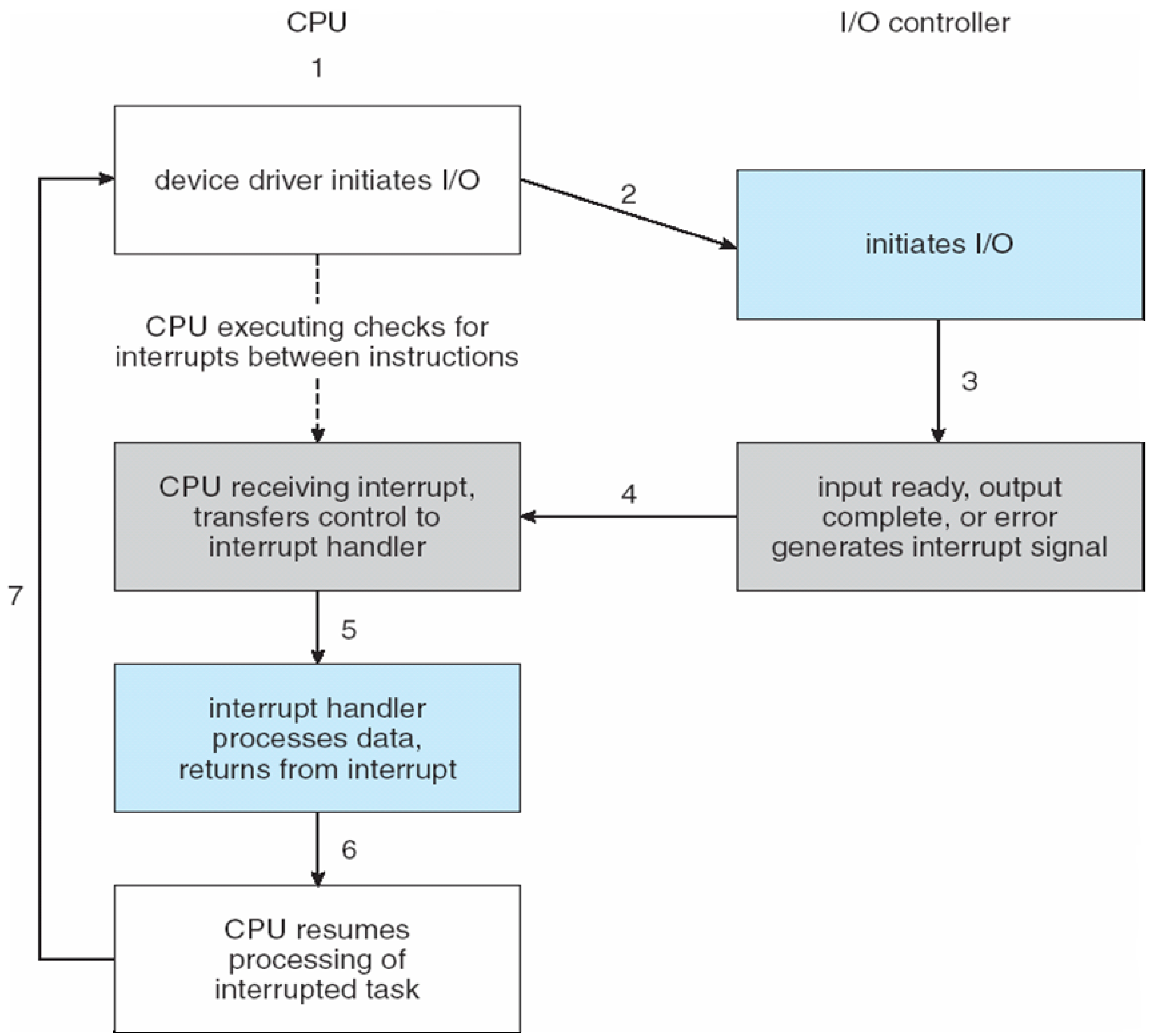
Interrupts

- Polling can happen in 3 instruction cycles
  - 1) Read status, 2) logical-and to extract status bit, 3) branch if not zero
  - How to be more efficient if non-zero infrequently?
- CPU Interrupt-request line triggered by I/O device
  - Checked by processor after each instruction
- Interrupt handler receives interrupts
  - Maskable to ignore or delay some interrupts → CPU can turn ON/OFF
- Interrupt vector to dispatch interrupt to correct handler
  - Context switch at start and end
  - Based on priority
  - Some nonmaskable → reserved for events such as unrecoverable memory errors
  - Interrupt chaining if more than one device at same interrupt number

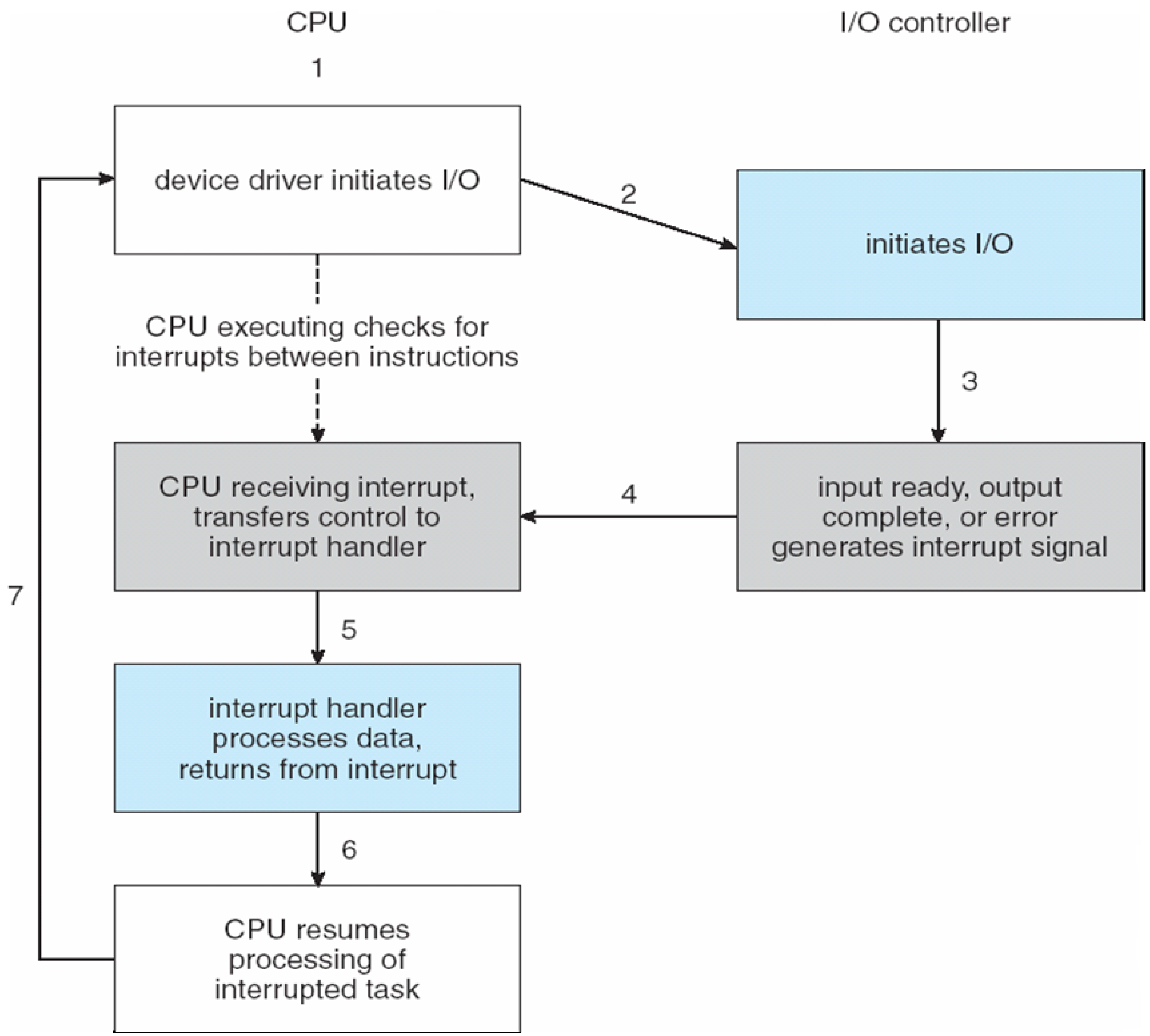
Interrupt-Driven I/O Cycle



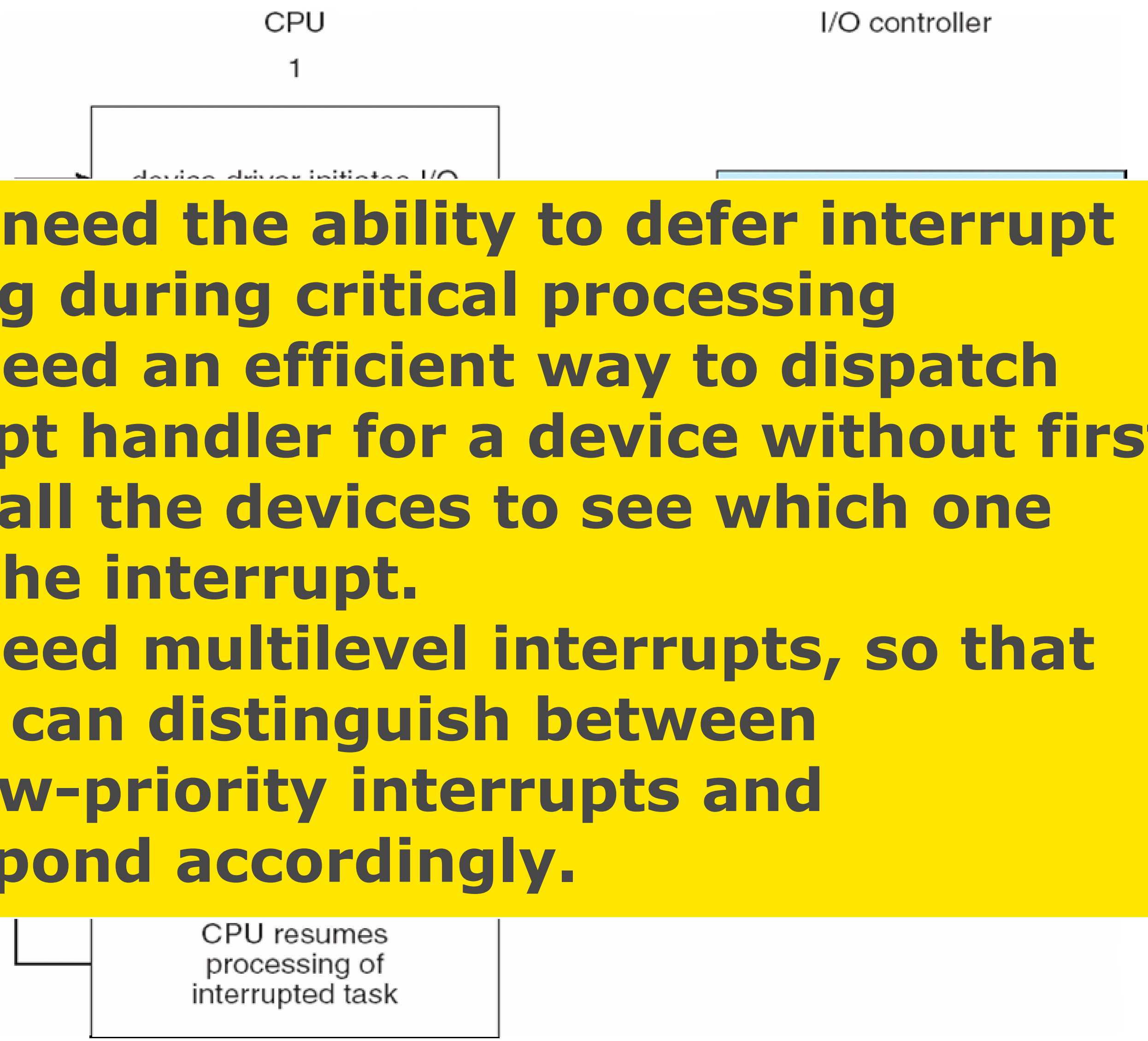
Interrupt-Driven I/O Cycle



Interrupt-Driven I/O Cycle



Interrupt-Driven I/O Cycle



1. We need the ability to defer interrupt handling during critical processing
2. We need an efficient way to dispatch Interrupt handler for a device without first polling all the devices to see which one raised the interrupt.
3. We need multilevel interrupts, so that the OS can distinguish between high/low-priority interrupts and can respond accordingly.

Interrupts (Cont.)

- Interrupt mechanism also used for exceptions
  - Terminate process, crash system due to hardware error
- Page fault executes when memory access error
- System call executes via trap to trigger kernel to execute request
- Multi-CPU systems can process interrupts concurrently
  - If operating system designed to handle it
- Used for time-sensitive processing, frequent, must be fast

Interrupts (Cont.)

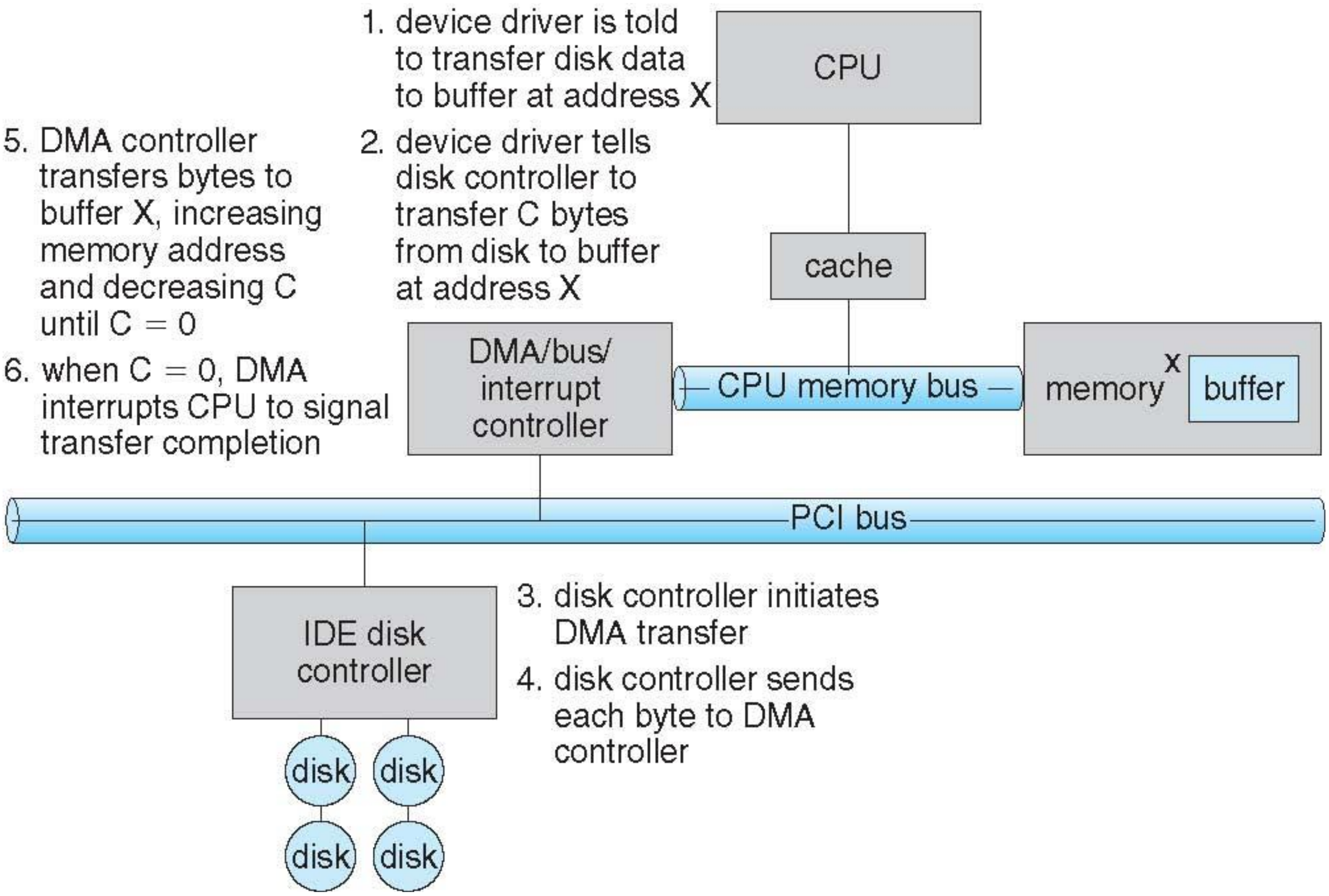
- Interrupt mechanism also used for exceptions
  - Terminate process, crash system due to hardware error
- Page fault executes when memory access error
- System call executes via trap to trigger kernel to execute request
- Multi-CPU systems can process interrupts concurrently
  - If operating system designed to handle it
- Used for time-sensitive processing, frequent, must be fast

Interrupts are used to handle asynchronous events and to trap to supervisor-mode routines in the kernel.

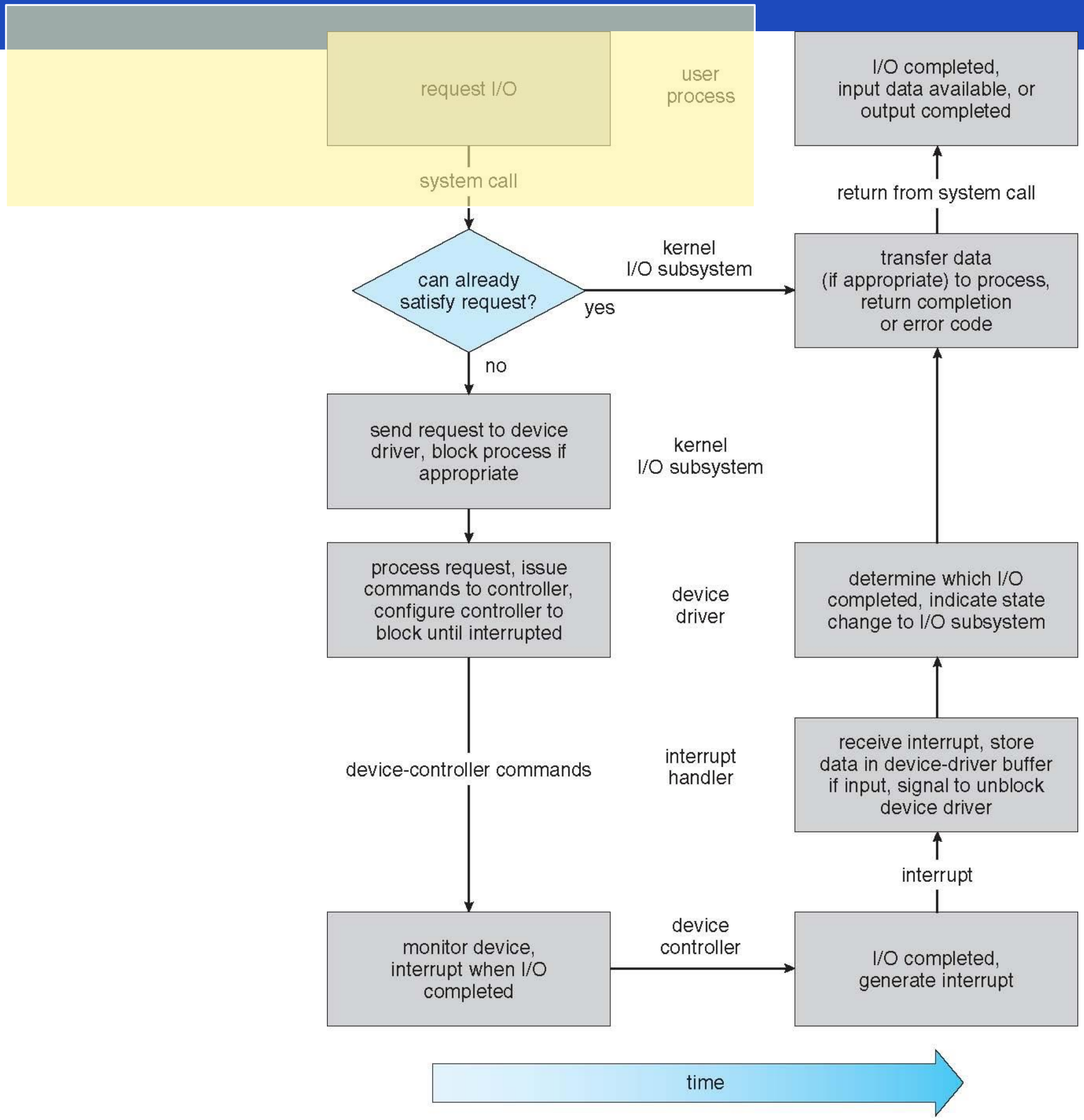
Direct Memory Access

- Used to avoid programmed I/O (one byte at a time) for large data movement
- Requires DMA controller
- Bypasses CPU to transfer data directly between I/O device and memory
- OS writes DMA command block into memory
  - Source and destination addresses
  - Read or write mode
  - Count of bytes
  - Writes location of command block to DMA controller
  - Bus mastering of DMA controller – grabs bus from CPU
  - When done, interrupts to signal completion

Six Step Process to Perform DMA Transfer

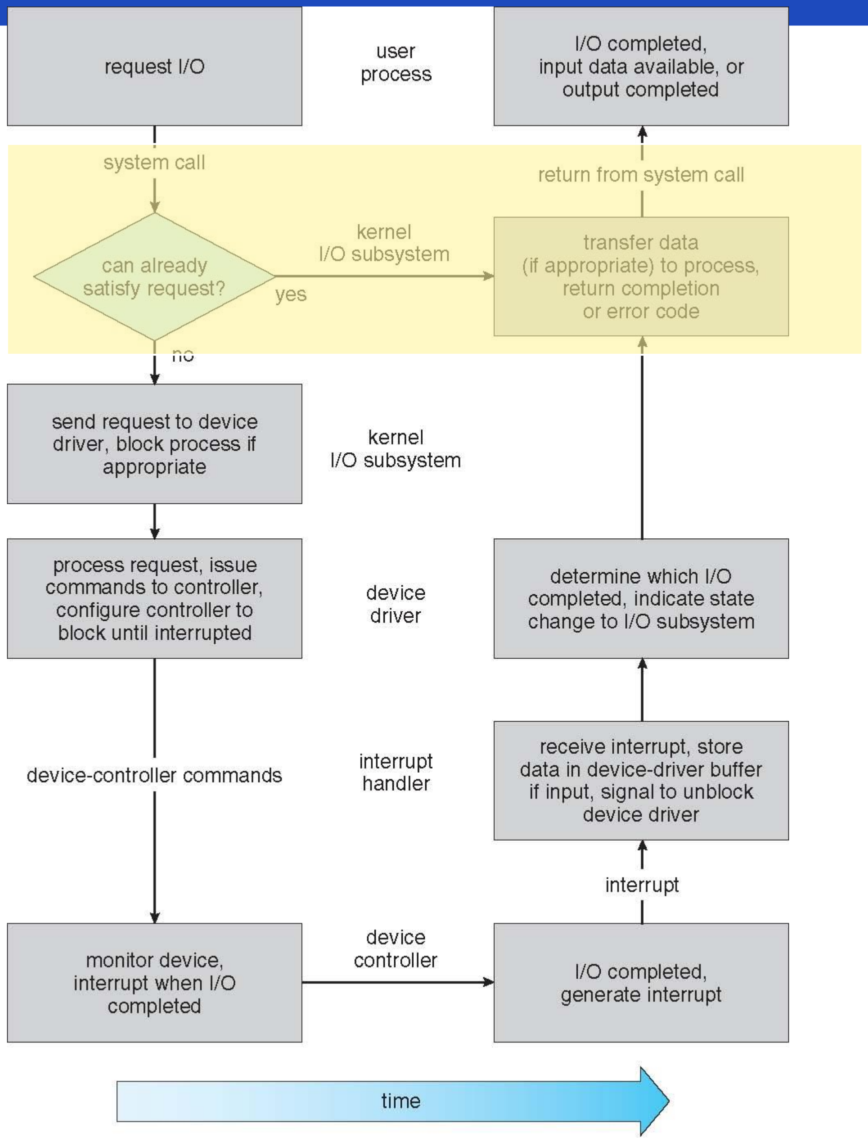


Life Cycle of An I/O Request



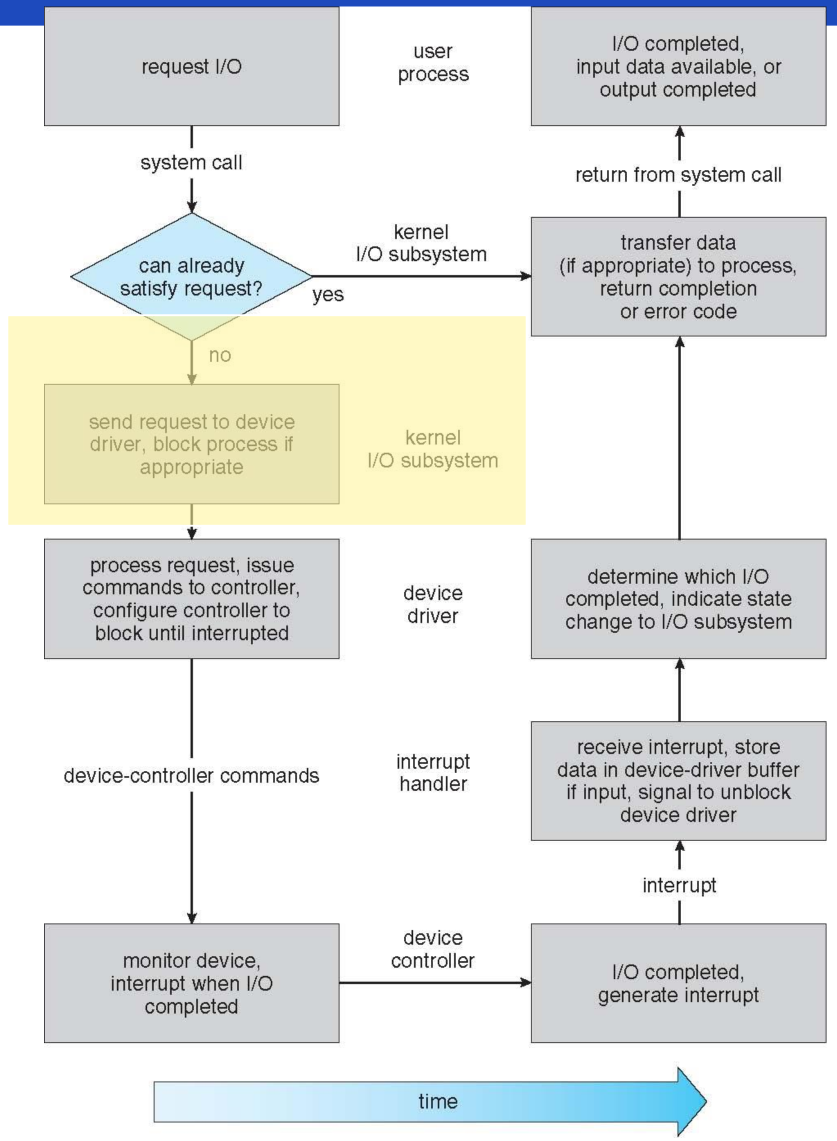
(1) A process issues a blocking read () to a file descriptor of a file → opened previously

Life Cycle of An I/O Request



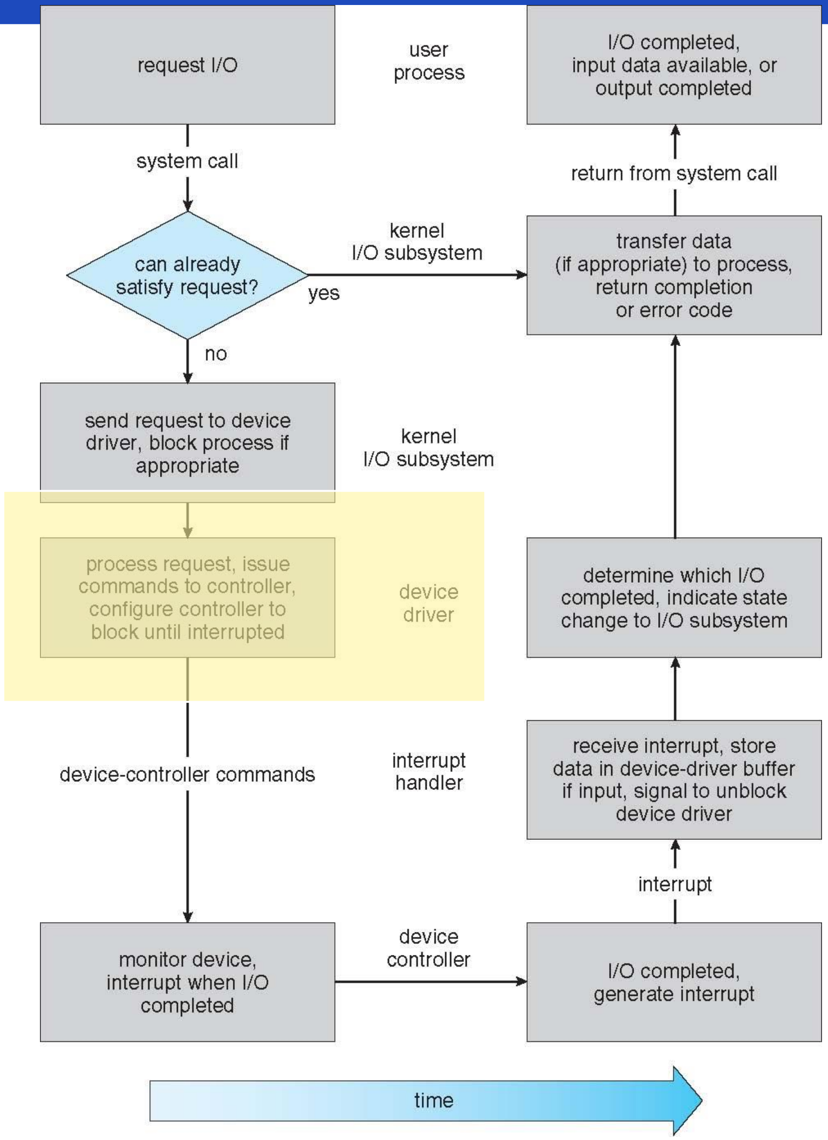
(2) If the data is available in buffer cache the data are returned to the process.

Life Cycle of An I/O Request



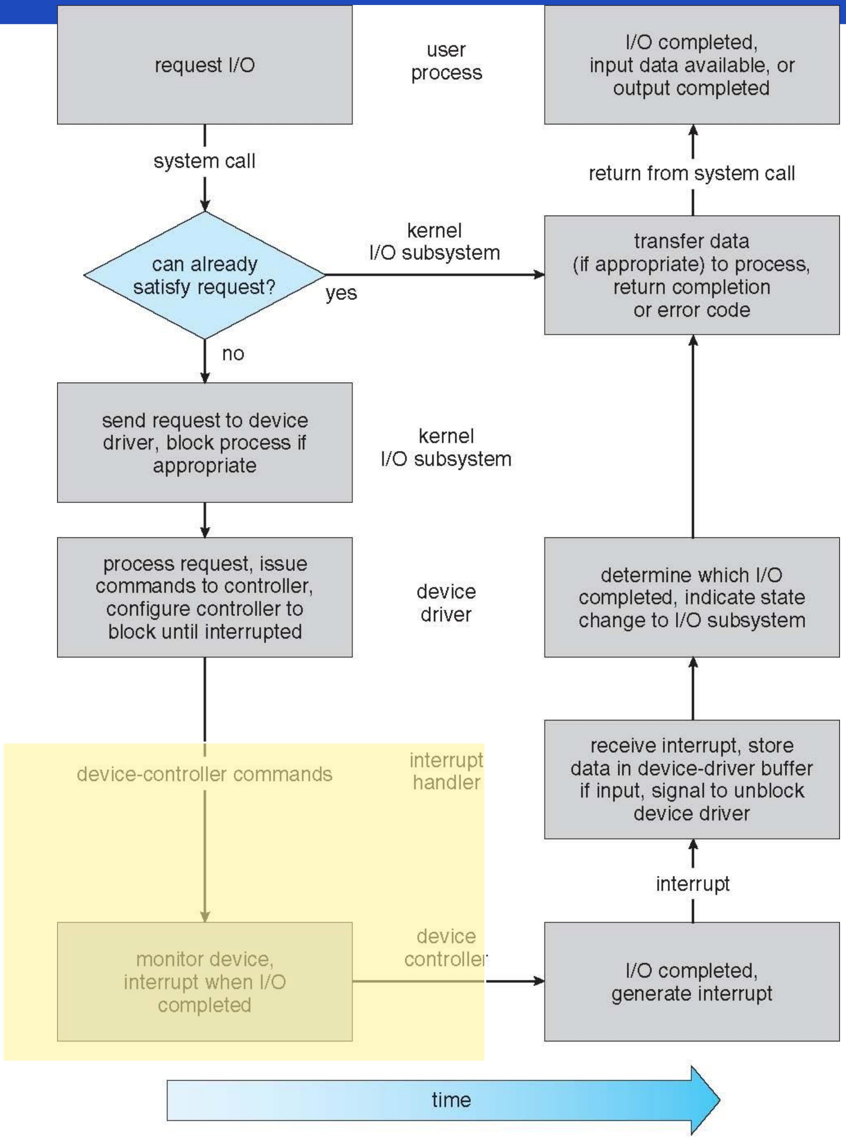
(3) The process is moved from the run queue to wait queue

Life Cycle of An I/O Request



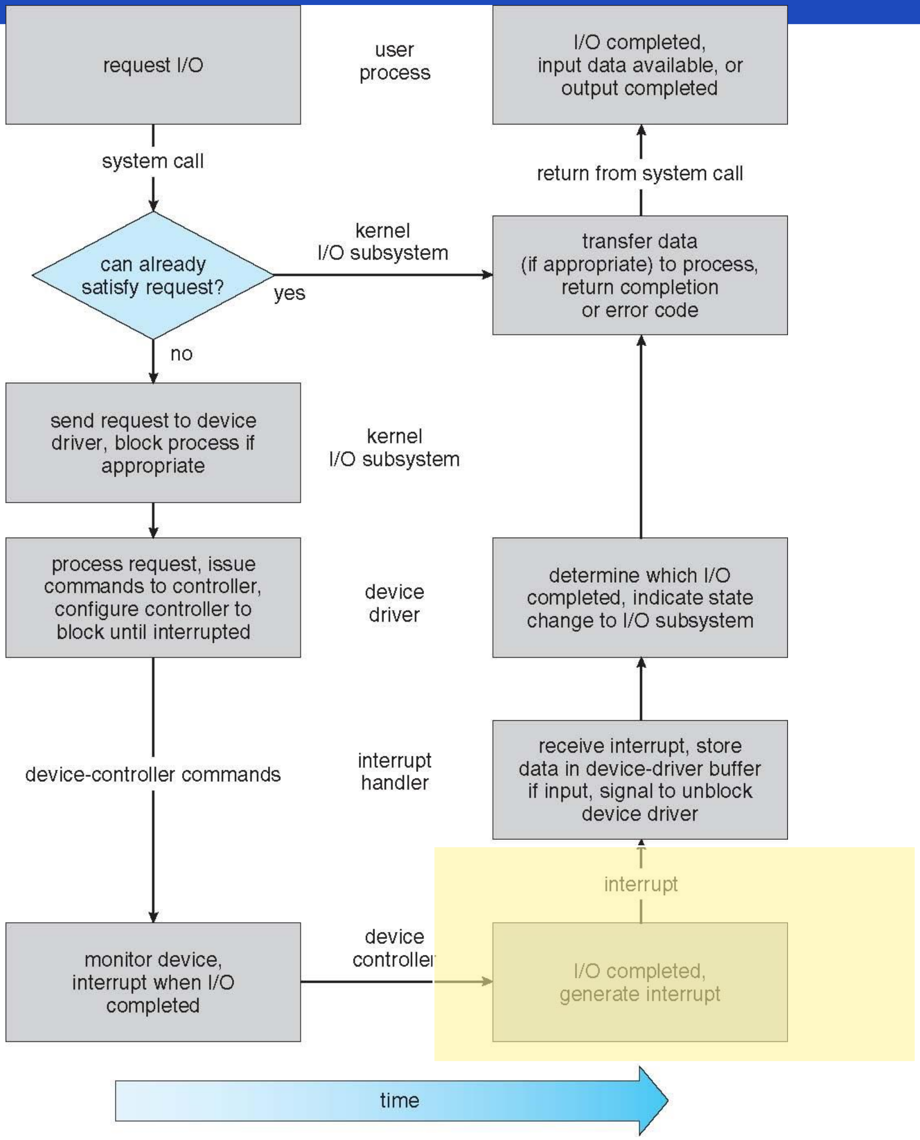
(4) Device driver allocates kernel buffer. Driver sends commands to the device controller by writing into the device-control registers

Life Cycle of An I/O Request



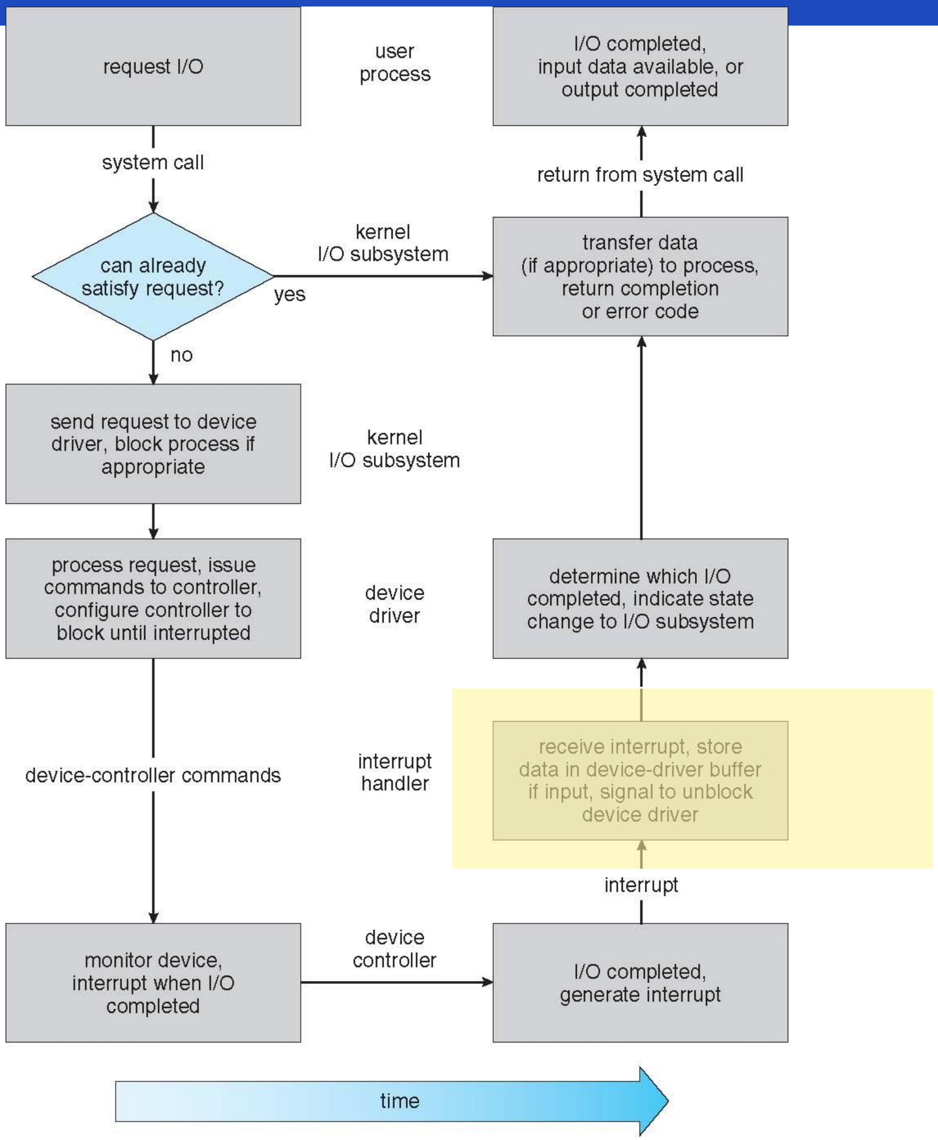
(5) Device controller operates the device hardware to perform the data transfer.

Life Cycle of An I/O Request

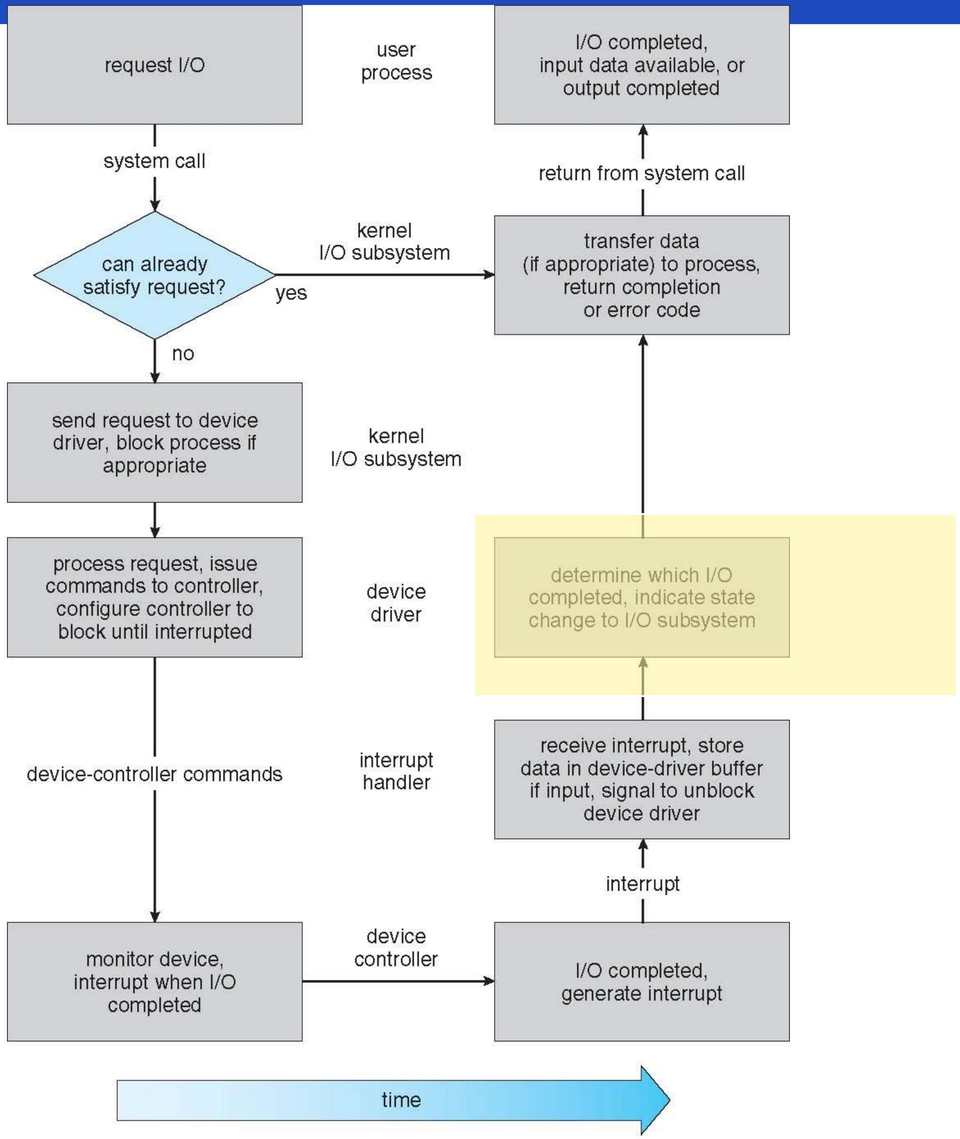


(6) Driver may poll for status and data or setup a DMA

Life Cycle of An I/O Request



Life Cycle of An I/O Request



Final Exam

Syllabus:

- **Processes**
- **Threads**
- Process Synchronization
- CPU Scheduling
- Deadlocks
- Main memory
- Mass Storage Structure
- File System Interface
- File System Implementation
- I/O Systems

References

Part of the contents of this lecture has been adapted from the book Abraham Silberschatz, Peter B. Galvin, Greg Gagne: "Operating System Concept ", Publisher : Wiley; 9 edition (December 17, 2012), ISBN-13: 978-1118063330

Slides also contain lecture materials from John Kubiatowicz (Berkeley), John Ousterhout (Stanford), Nalini (UCI), Rainer (UCI), and others

Some slides adapted from http://www-inst.eecs.berkeley.edu/~cs162/ Copyright © 2010 UCB

Thank you for your attention