
L40- INTRODUCTION TO IO

Overview of I/O Subsystem Management

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

Objectives

- Explore the structure of an operating system’s I/O subsystem
- Discuss the principles of I/O hardware and its complexity
- Provide details of the performance aspects of I/O hardware and software

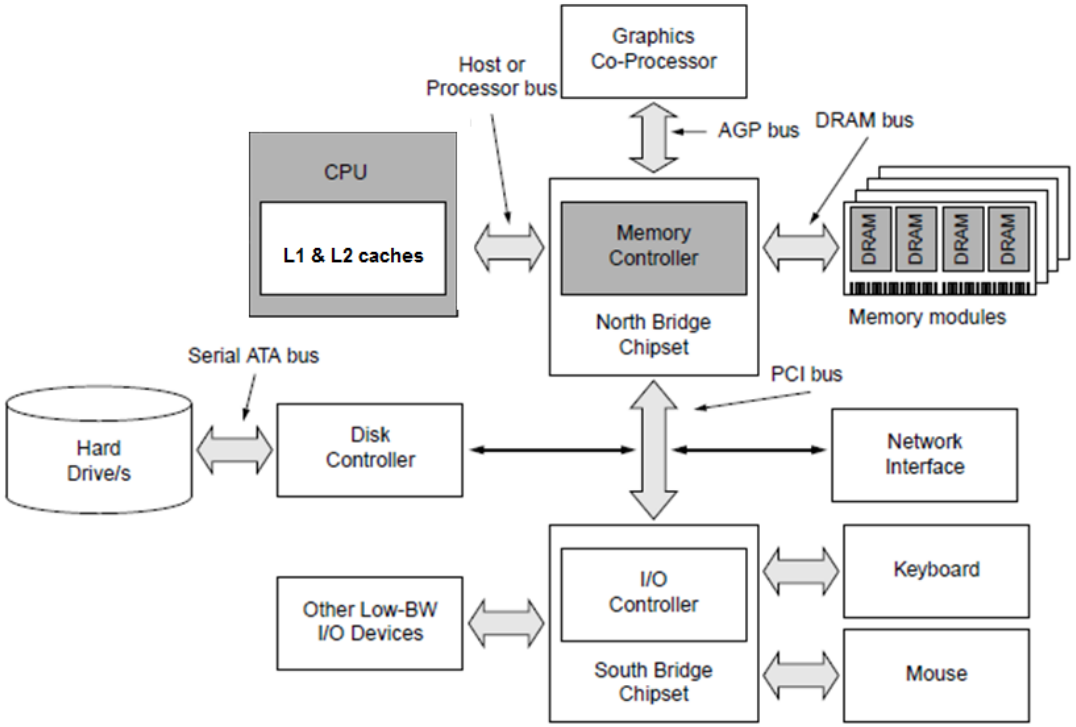
Importance and Challenges in I/O Management

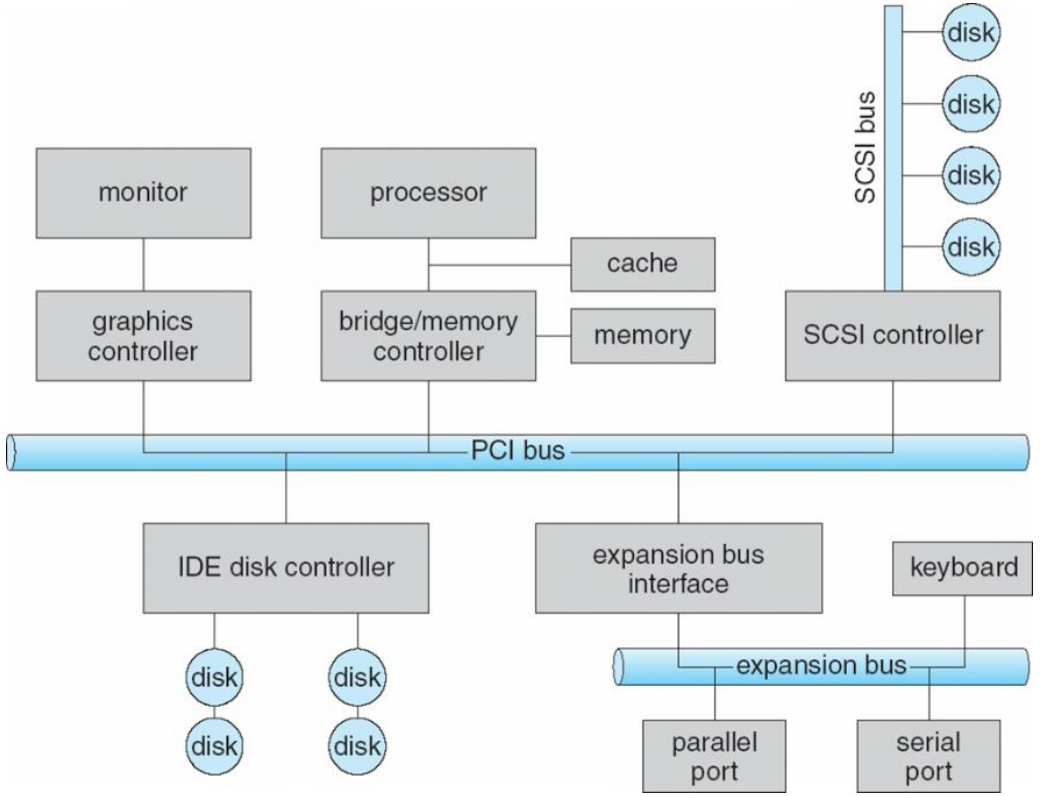
- I/O management is a major component of operating system design and operation
  - Important aspect of computer operation
  - I/O devices vary greatly
  - Various methods to control them
  - Performance management
  - New types of devices frequent
- Ports, busses, device controllers connect to various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

I/O Hardware

- Incredible variety of I/O devices
  - Storage
  - Transmission
  - Human-interface

PCI Bus Structure



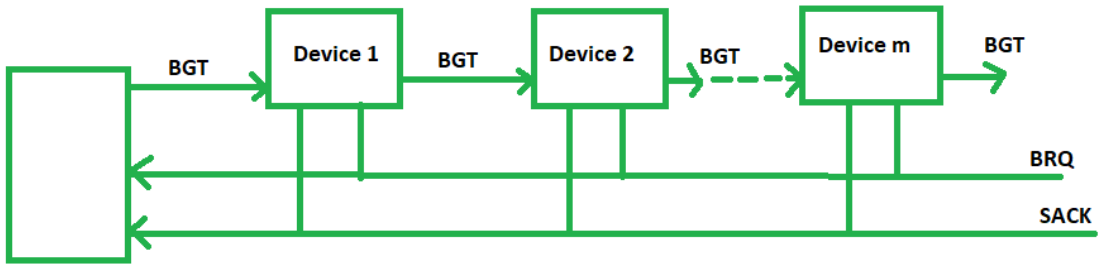


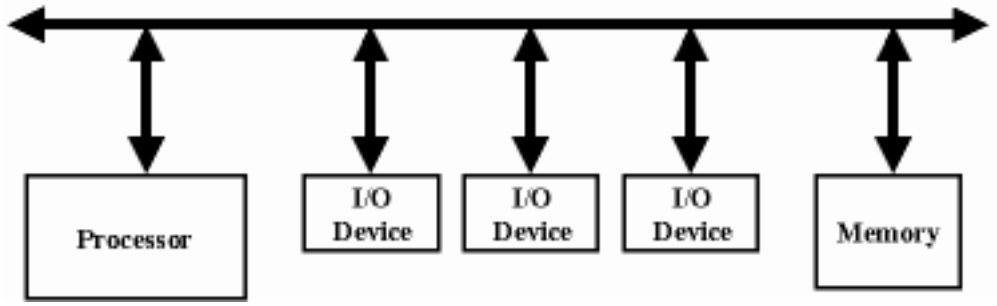
Components of I/O Subsystem

- I/O Hardware
  - ports, buses, devices, controllers
- I/O Software
  - Interrupt Handlers, Device Driver,
  - Device-Independent Software,
  - User-Space I/O Software
- I/O Data transfer mechanisms
    - Polling, Interrupt and DMAs

I/O Hardware

- Signals from I/O devices interface with computer
  - **Port** – connection point for device
  - **Bus - daisy chain** or shared direct access
    - **PCI** bus common in PCs and servers, PCI Express (**PCIe**)
    - **expansion bus** connects relatively slow devices





I/O Hardware

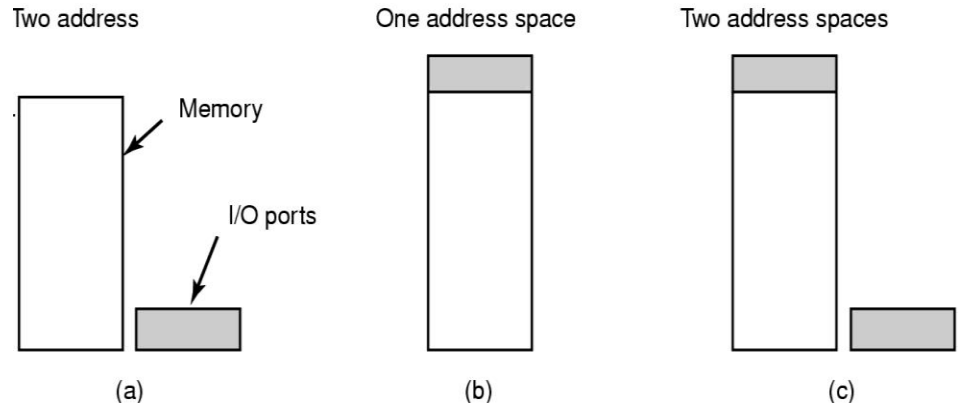
- **Controller** (**host adapter**) – electronics that operate port, bus, device
  - Sometimes integrated
  - Sometimes separate circuit board (host adapter)
  - Contains processor, microcode, private memory, bus controller, etc

I/O Hardware

- I/O instructions control devices
- Devices usually have registers where device driver places commands, addresses, and data to write, or read data from registers after command execution
  - Data-in register, data-out register, status register, control register
  - Typically 1-4 bytes, or FIFO buffer
- Devices have addresses, used by I/O instructions

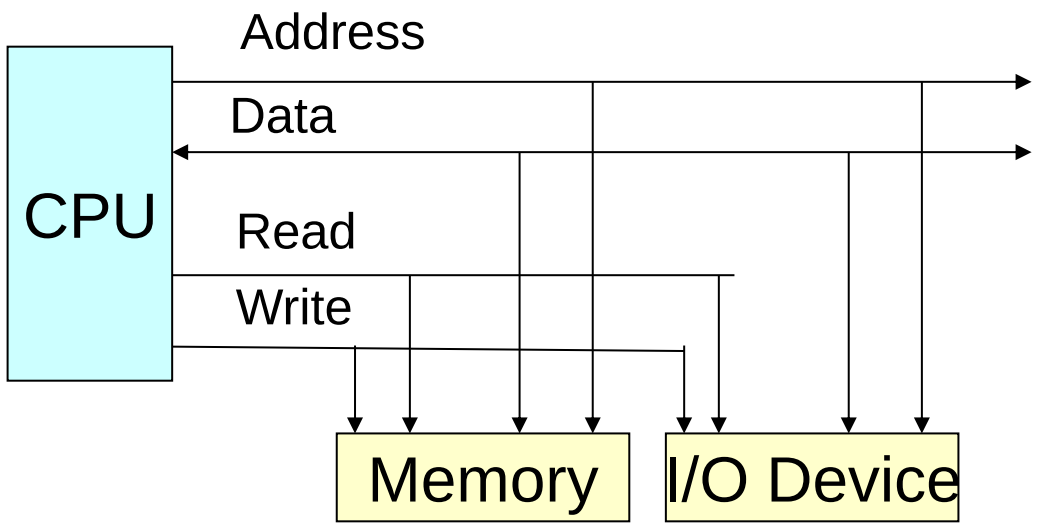
I/O Mapping

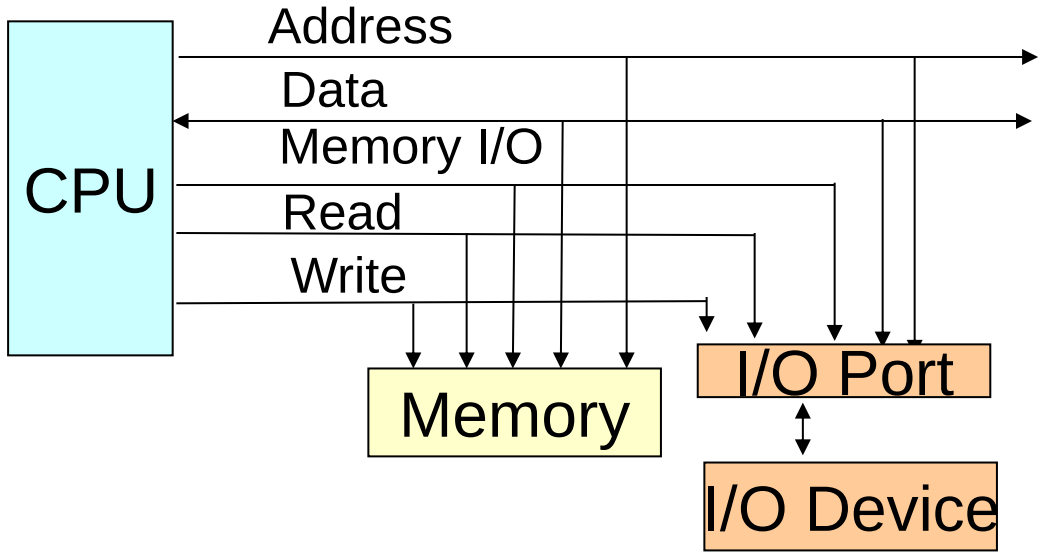
- Memory mapped I/O
  - Devices and memory share an address space
  - I/O looks just like memory read/write
  - No special commands for I/O
  - Large selection of memory access commands available
- Isolated I/O (I/O mapped I/O)
  - Separate address spaces
  - Need I/O or memory select lines
  - Special commands for I/O; Limited set



I/O Mapping

- CPU needs to talk to I/O
- **Memory mapped I/O**
  - Devices mapped to reserved memory locations - like RAM
  - Uses load/store instructions just like accesses to memory
- **I/O mapped I/O**
  - Special bus line
  - Special instructions





I/O Basics

- I/O module interface I/O to CPU and Memory
- **I/O controller ←→ I/O devices ports**
  - Transfers data to/from device
  - Synchronizes operations with software
- **Status/ control registers**: device status, errors
- **Data registers**
  - Write: CPU/RAM data → device [eg Transmit]
  - Read: CPU← device [eg Receive]

Functions of I/O Module

- Control & Timing
- Processor Communication
- Device Communication
- Data Buffering
- Error Detection (e.g., extra parity bit)

Basic I/O Steps

- CPU checks I/O module device status
- I/O module returns status
- If ready, CPU requests data transfer by sending a command to the I/O module
- I/O module gets a unit of data (byte, word, etc.) from device
- I/O module transfers data to CPU
- Variations of these steps for different I/O mechanisms like poling, interrupt and DMA based I/O.

Thank You