Contents

1. Introduction
2. System Structures
3. Process Concept
4. Multithreaded Programming
5. Process Scheduling
6. Synchronization
7. Deadlocks
8. Memory-Management Strategies
9. Virtual-Memory Management
10. File System
11. Mass-Storage Structures
12. I/O Systems
13. Protection, Security, Distributed Systems

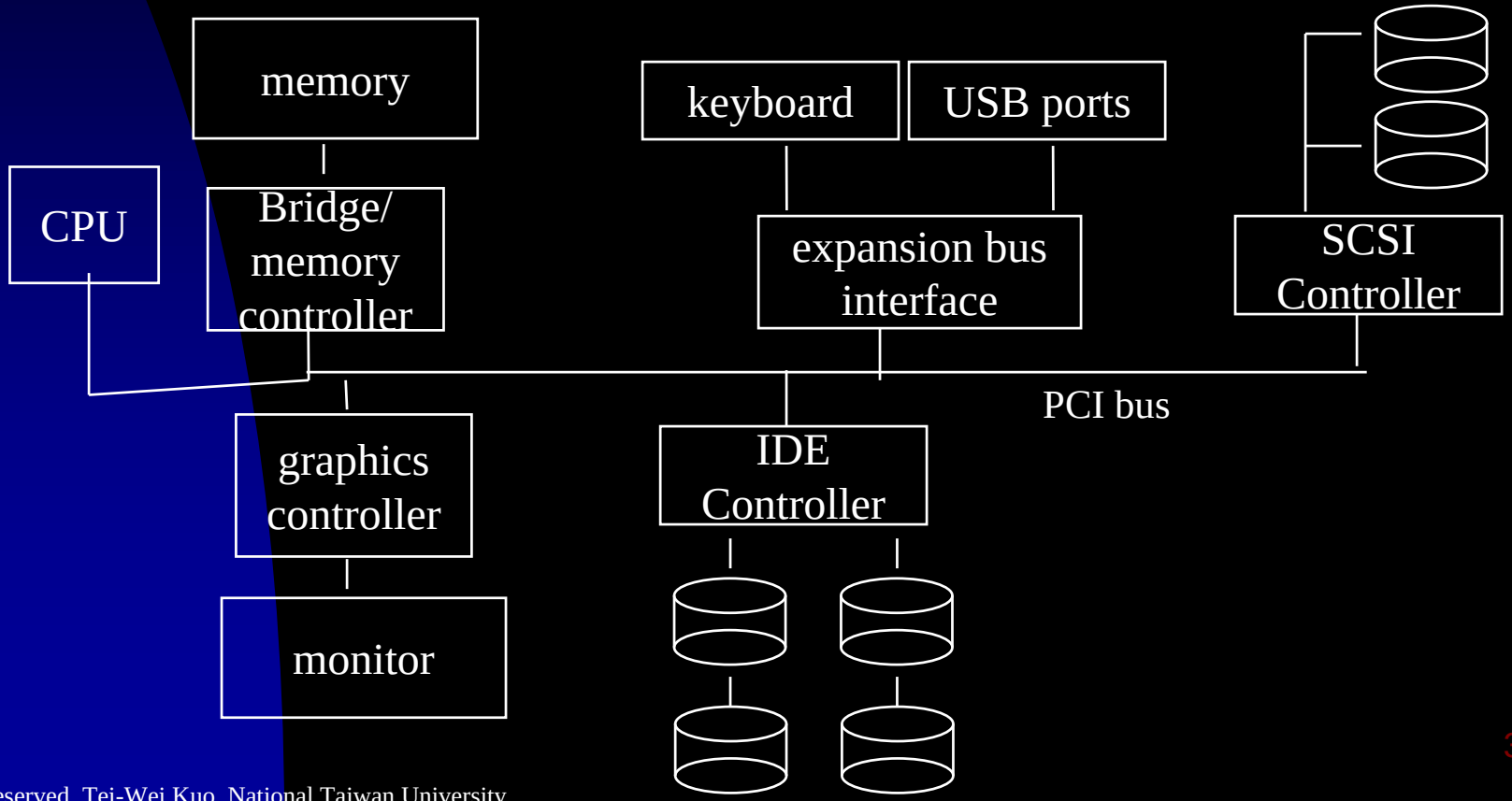
pp. 575-595.

I/O Systems

- One Major Role of OS is to Manage and Control I/O Operations and I/O Devices.
- Two Conflicting Trends of I/O Device Technology
  - Standardization of Software and Hardware Interfaces
  - An Increasingly Broad Variety of I/O Devices
- Device Driver: A uniform device access interface to the I/O Subsystem.

I/O Hardware

- Port: A connection point
- Bus: A set of wires and a protocol
  - A Daisy Chain, PCI Bus, Expansion Bus



I/O Hardware

- Controller: A set of electronics that can operate a bus, a port, or a device.
- Host Adaptor: A controller plugged into the computer.
- Memory-Mapped I/O: Device-control registers are mapped to a processor address space
  - Data-in registers
  - Data-out registers
  - Status register
  - Control register

I/O Hardware

- Polling – Programmed I/O
- Interrupt-Driven Handling
  - When the CPU detects that a controller has asserted a signal on the interrupt request line, the CPU performs a state save and jumps to the corresponding interrupt service routine at a fixed memory location.
- Direct Memory Access
  - Release CPU from handling excessive interrupts!

Application I/O Interface

- Objective:
  - Enable I/O devices to be treated in a standard, uniform way.
- Interface
  - Each general kind of devices is accessed through a standardized set of functions.
- Device Driver
  - Hide the differences among device controllers from the I/O subsystem of the I/O Kernel

Application I/O Interface

- Characteristics of I/O Devices
  - Character-Stream or Block
  - Sequential or Random Access
  - Synchronous or Asynchronous
  - Sharable or Dedicated
  - Speed of Operation
  - Read-Write, Read Only, or Write Only

. Examples: Terminal, Disk; Modem, CD-ROM; Tape, Keyboard;-;
CD-ROM, Graphics Controller, Disk

Application I/O Interface

- Access Conventions
  - Block I/O
  - Character-Stream I/O
  - Memory-Mapped File Access
  - Network Sockets
  - Special System Calls
    - Time-of-Day Clock & Timer
  - Escape or Backdoor
    - ioctl()

Block and Character Devices

- Block-Device Interface
  - Commands: read(), write(); seek() for random-access device,
  - Raw I/O: Access a block device as a simple linear array of blocks.
  - Memory-Mapped File Access: A file is mapped to the virtual memory, and actual data transfers are done when needed to satisfy access to the memory image.
- Character-Stream Interface
  - Commands: get(), put(); library calls
  - Devies with I/O of a linear stream of bytes.

Network Devices, Clocks and Timers

- Network Socket Interface
  - Commands: Socket creation, connection of a local socket to a remote address, listen to a local socket, packet sending and receiving, select(). etc.
- Clocks and Timers
  - Basic Functions: The current time, the elapsed time, setting of a timer to trigger operation X at time T, etc.
  - Programmable Interval Timer
    - Timer granularity, clock drifting, hardware counters

Blocking and Nonblocking I/O

- Blocking System Calls
  - The suspension of an application execution due to a varying or unpredictable amount of time in performing physical actions to some I/O devices or better understanding in application programming.
- Nonblocking System Calls
  - A nonblocking read() returns immediately with whatever data are available, where an asynchronous read() requests for a transfer that would be done completely possible at some future time.

Kernel I/O Subsystem

- Services and Procedures
  - Management of the Name Space for Files and Devices
  - Access Control to Files and Devices
  - Operation Control
  - File-System Space Allocation
  - Device Allocation
  - Buffering, Caching, and Spooling
  - I/O Scheduling
  - Device-Status Monitoring, Error Handling, and Failure Recovery
  - Device-Driver Configuration and Initialization

Kernel I/O Subsystem

- Why I/O scheduling
  - Performance Improvement
  - Fairness in Access
- Why Buffering?
  - Speed Mismatching
  - Different Transfer Sizes
  - Copy Semantics for Application I/O
- Why Caching?
  - Efficiency in Accessing a Cached Copy.