

# 计算机操作系统

# Operating Systems

田卫东

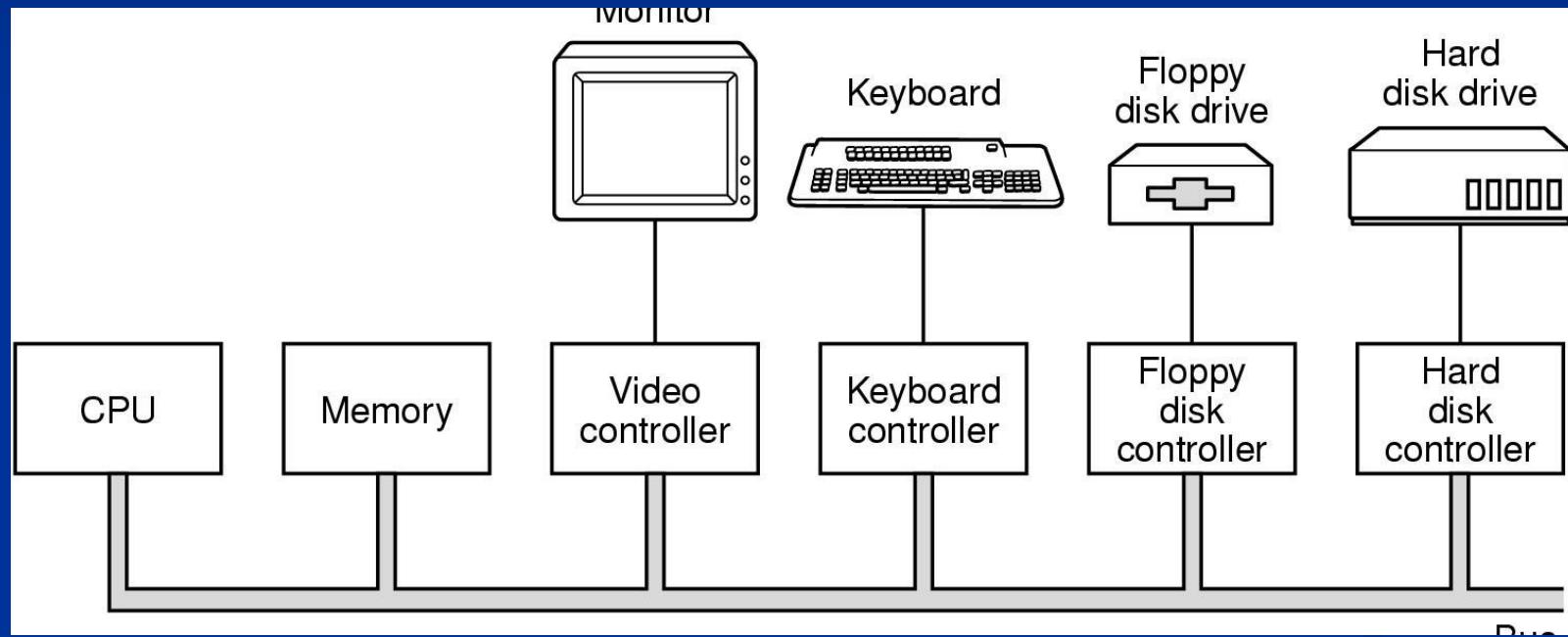
March, 2014

# 第2章 进程管理

# 补充材料：OS运行与应用程序运行

## (1) 操作系统如何实现对计算机的控制

### ■ 计算机系统架构



# (1) 操作系统如何实现对计算机的控制

## ■ 程序运行与操作系统运行

- 操作系统初始化
- 运行应用程序
- 正常： 普通代码 (for, int, while etc.)
- 特殊： 调用OS代码

(printf, scanf, cont. etc.)

## 用户程序

0

max



## OS内核



# 补充材料：OS运行与应用程序运行

## (2) 指令系统：分类

- 访问存储器指令：

- 处理器和存储器间数据传送

- I/O指令：

- 处理器和I/O模块间数据传送和命令发送

- 算术逻辑指令（数据处理指令）：

- 执行数据算术和逻辑操作

- 控制转移指令：

- 指定一个新的指令的执行起点

- 处理器控制指令：

- 修改处理器状态，改变处理器工作方式

# 补充材料： OS运行与应用程序运行

## (2) 指令系统：特权指令和非特权指令

### ■ 特权指令

权限级别较高的指令，只能由操作系统使用。

例如：I/O和处理器控制指令: IN AL, 3FH; OUT 80H,AL

### ■ 非特权指令

权限级别较低的指令，任何程序都可以使用。

例如：数据传送指令、算术逻辑运算及跳转指令等

特权指令

非特权指令

完整指令集

# 补充材料：OS运行与应用程序运行

## (3) 系统态和用户态：限制特权指令的执行

■ 根据运行程序对资源和机器指令的使用权限将处理器设置为不同工作状态：系统态（管态）和用户态（目态）。

□ 系统态：操作系统管理程序运行的状态，较高的特权级别，又称为特权态（特态）、核心态、内核态、管态，可以运行所有指令（包括特权指令）；

□ 用户态：用户程序运行时的状态，较低的特权级别，又称为普通态（普态）、目态，只可以运行非特权指令。



# 补充材料： OS运行与应用程序运行

## (4) 执行

### ■ 系统态与用户态转换

□ 用户态切换到系统态：执行特殊的访管指令（陷入）；

□ 系统态切换到用户态：执行特殊指令；

### ■ 特权指令的执行

系统态下执行，没有任何问题；

用户态下执行，引起处理器异常（用户态转变为系统态），由异常处理程序（属于操作系统内核）接管。

## (5) 实例：Intel x86环保护机制



- 4级：Ring0-Ring4；  
各级有保护性检查（地址校验、I/O限制）
- 4个级别运行不同类程序：
  - Ring0-操作系统核心代码
  - Ring1-运行关键设备驱动程序和I/O处理例程
  - Ring2-其他受保护共享代码，如语言系统运行环境
  - Ring3-各种用户程序
- 现有基于x86处理器的操作系统，多数UNIX、Linux以及Windows系列只用了Ring0和Ring3两级