

课程内容概要：

## 1. 操作系统含义与意义、主要特征与发展历史

（会讲一些这个操作系统的基本概念，什么叫虚拟机，什么叫裸机，然后操作系统的组成，以及它主要的一些核心，以及包括什么文件系统是不是必要的）

含义及意义：

用户观点：os 是用户与计算机硬件之间的接口

资源管理观点：os 是计算机系统资源的管理者

虚拟机观点：一台完全无软件的计算机称为裸机，即使功能再强，也是难于使用的。os 是一组控制和管理计算机硬件和软件资源，合理地各类作业进行调度，以及方便用户使用的程序的集合。

综上：os 是一组控制和管理计算机硬件和软件资源，合理地各类作业进行调度，以及方便用户使用的程序的集合。

操作系统的功能：处理器管理（在层次结构中，是操作系统的核心部分，位于最内层）；存储器管理；设备管理；文件管理；提供用户接口（命令接口、应用程序接口、图形接口）。

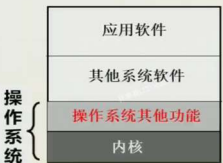
操作系统的内核结构：整体结构；模块结构；层次结构；微内核结构

- 整体结构（linux 操作系统使用）

系统效率高、系统灵活性好

## 1.5 操作系统内核结构

### 一. 整体式结构（单体结构，宏内核）



**概念：**操作系统是一组过程的集合，每个过程有接口定义，包括入口参数和返回值，过程间可任意调用。

**设计重点：**功能的实现和高效率

**缺点：**缺乏清晰的程序结构、错误多、难以维护


**实例：**Linux、AT&T SystemV

**Linux内核模块概念**

- 模块结构

## P35 操作系统内核结构

### 二. 模块化结构

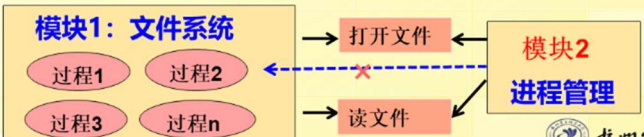


**概念：**操作系统按功能划分模块，模块要封装，具有良好接口定义，模块间可任意调用

**特点：**有利于系统设计和扩展

**缺点：**模块间依赖关系复杂，使OS结构不清

**实例：**Choices系统，实验性质



- 层次结构

8 整体问题局部化，结构清晰，有利于功能的增加、删除、修改

上层模块可以调用下层模块

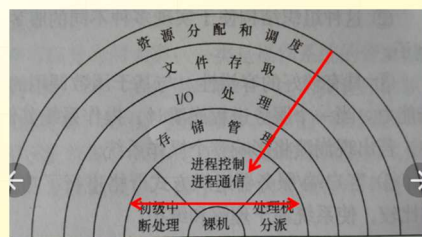
## P36 操作系统内核结构

### 三. 层次结构模型

把OS的**功能模块**划分为若干层，每层之间的模块只能**单向调用**



## P37 结构划分分析



**优点：**易保证系统的正确性，易扩充和易维护性

**缺点：**系统效率降低

**实例：**Dijkstra的THE系统，Unix系统

## P38 操作系统内核结构

### 三. 层次结构模型

#### 小组讨论回答：

一个分层结构操作系统有裸机、用户管理、CPU调度、文件管理、作业管理、内存管理、设备管理和命令管理等部分组成，试按层次结构的原则从内到外将各部分进行层次组织。

#### 可能顺序：

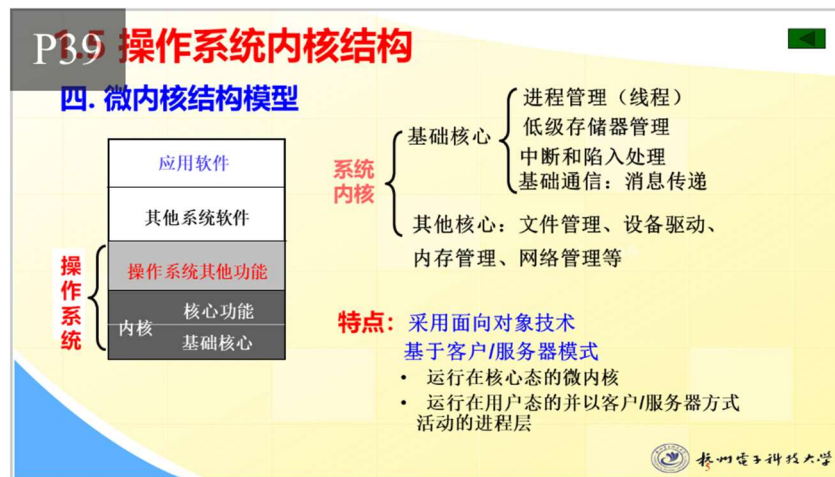
裸机、CPU调度、内存管理、设备管理、文件管理、作业管理、命令管理、用户管理



### • 微内核结构

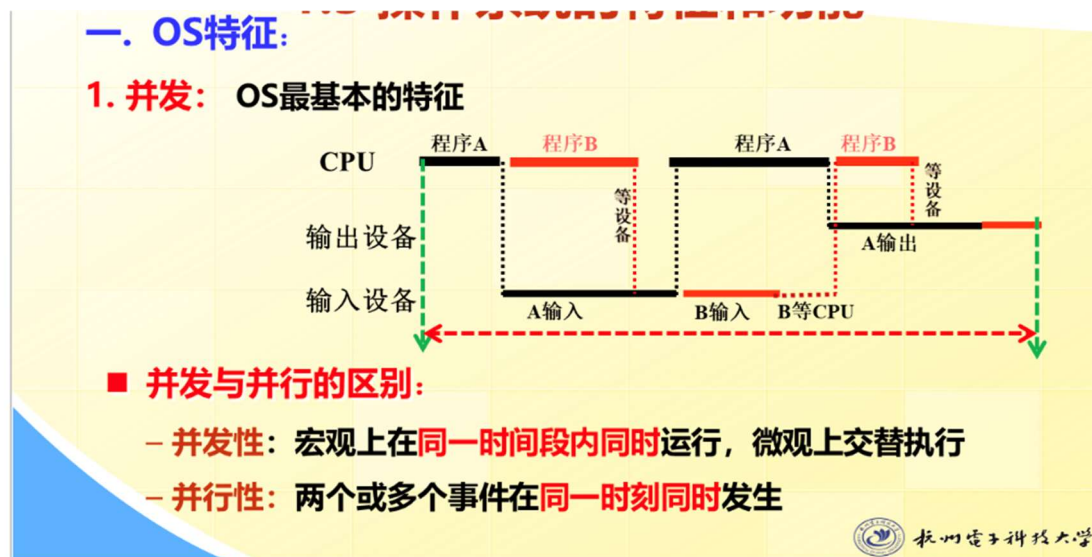
增强了可靠性，可运行于分布式系统中

使系统更高效不是微内核结构的特点



主要特征：

1、并发（最基本）



2、共享

## 1.3 操作系统的特征与功能

### 一. OS特征

#### 2. 共享

##### ■ 互斥共享方式

临界资源：打印机，扫描仪等

##### ■ 同时访问方式

宏观同时，微观轮流

处理机、内存

磁盘、可重入代码



### 3、虚拟

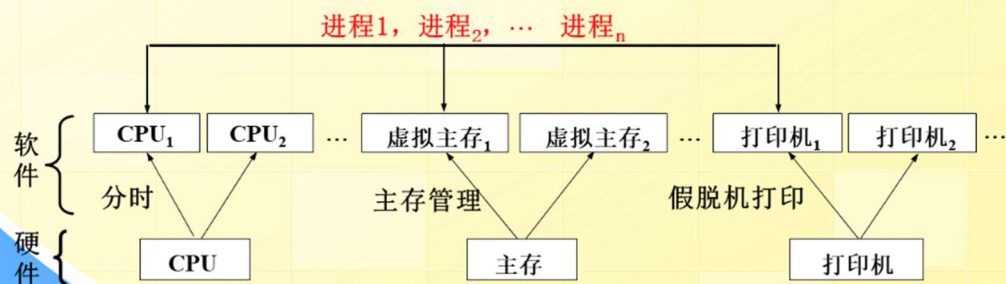
分为平台虚拟化和应用虚拟化

## 1.3 操作系统的特征与功能

### 3. 虚拟

#### 硬件虚拟化

通过某种技术把一个物理实体变为若干个逻辑上的对应物



### 4、异步

又称不确定

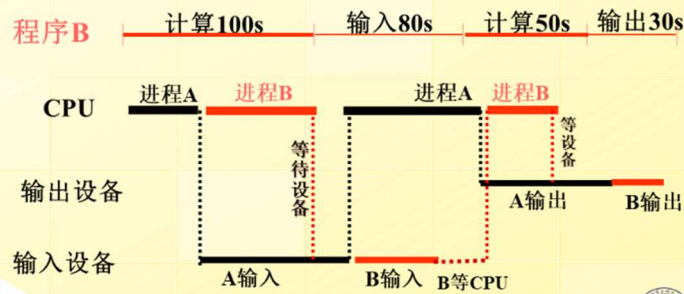


## 1.3 操作系统的特征与功能

### 一. OS特征

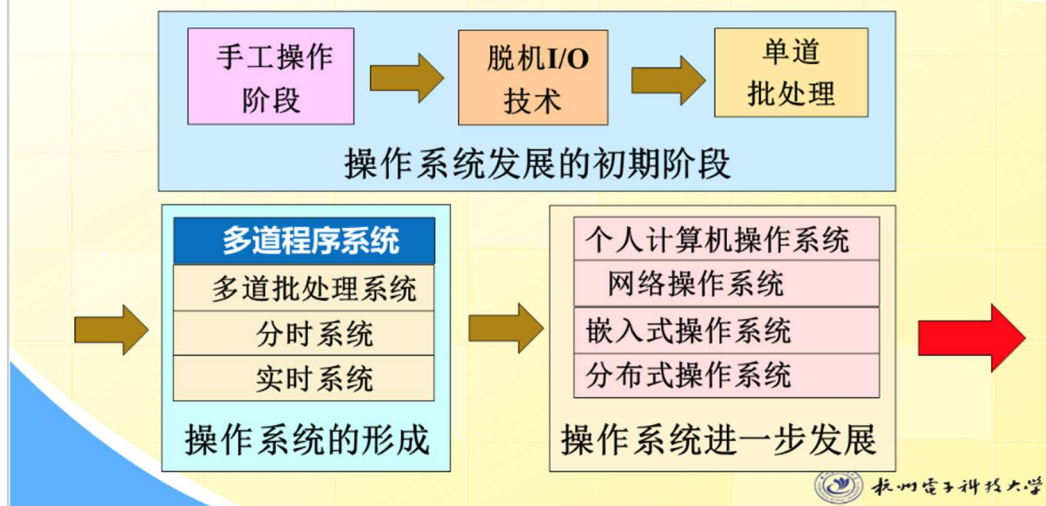
#### 4. 异步：进程以人们不可预知的速度向前推进

— 导致的原因：竞争资源



发展历史：

## 1.2 操作系统的发展与分类



• 无操作系统的计算机系统

• 手工操作阶段

• 脱机输入输出技术

P25 1.2 操作系统的发展与分类

### 一. 无操作系统的计算机系统

#### 1. 手工操作阶段

纸带(卡片) → 输入机 → 计算机 → 打印结果 → 取走纸带

该方式有什么不好呢?

- CPU等待人工操作: 人—机矛盾

机器速度	程序要求处理时间	人工操作时间	操作时间与机器有效运行时间之比
1万次/秒	1小时	3分钟	1:20
60万次/秒	1分钟	3分钟	3:1

- 用户独占全机
- 串行性

杭州电子科技大学

P26 1. 无操作系统的计算机系统

### 2. 脱机输入输出技术

脱机输入: 纸带机 → 外围机 → 磁带 → 主机 → 磁带 → 打印机

脱机输出: 主机 → 磁带 → 外围机 → 打印机

外围机又称为卫星机

优点: 减少主机CPU空闲时间; 提高I/O速度。

杭州电子科技大学

## • 单批道处理系统

P27 1.2 操作系统的发展过程

### 二. 单道批处理系统

#### 1. 作业的概念:

- 是用户定义的、由计算机完成的一个工作单位。作业由不同的顺序相连的作业步组成。
- 组成: 作业=程序+数据+作业说明书
- 应用: 早期批处理系统, 现代大型机、巨型机系统

后台作业:

Linux的shell中: 命令 &

Windows命令行界面: start /b <命令名>

杭州电子科技大学

P28 二. 单道批处理系统

### 2. 基本原理

用户作业 → 提交 → 分界线 → 常驻监控程序

常驻监控程序: 中断处理, 设备驱动程序, 作业序列, 控制语言解释器

用户程序区域: 作业3, 作业2, 作业1

关键技术: 脱机+作业自动过渡

脱机输出: 程序+数据+作业说明书

主机: 磁带, 监督程序, OS雏形, 作业运行结果

杭州电子科技大学

P29 二. 单道批处理系统

### 3. 单道批处理系统的性能分析

作业A: 计算 → 输入 → 计算

CPU: 作业A运行

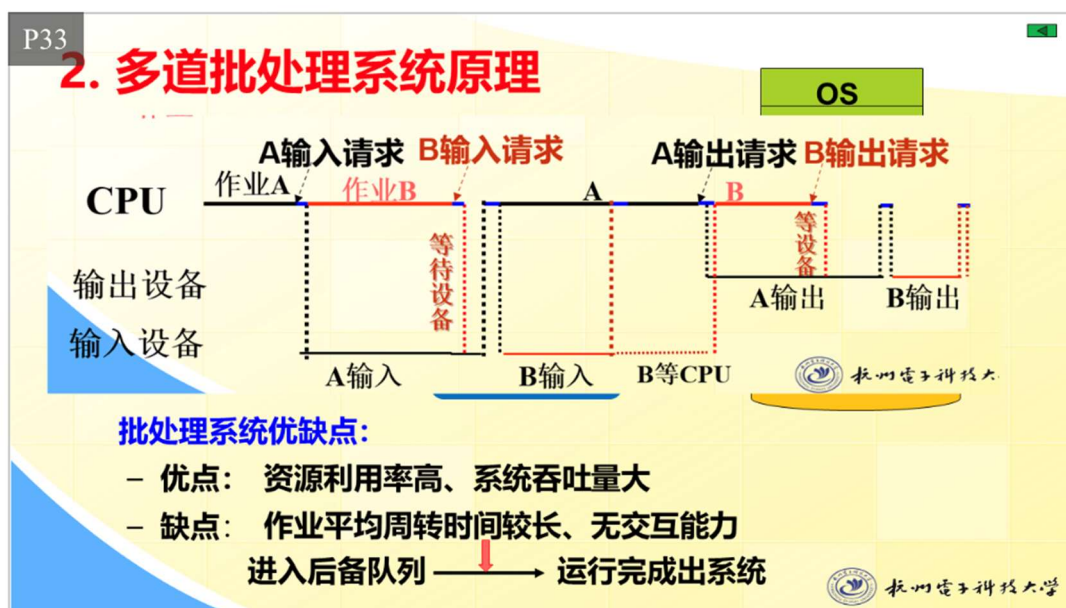
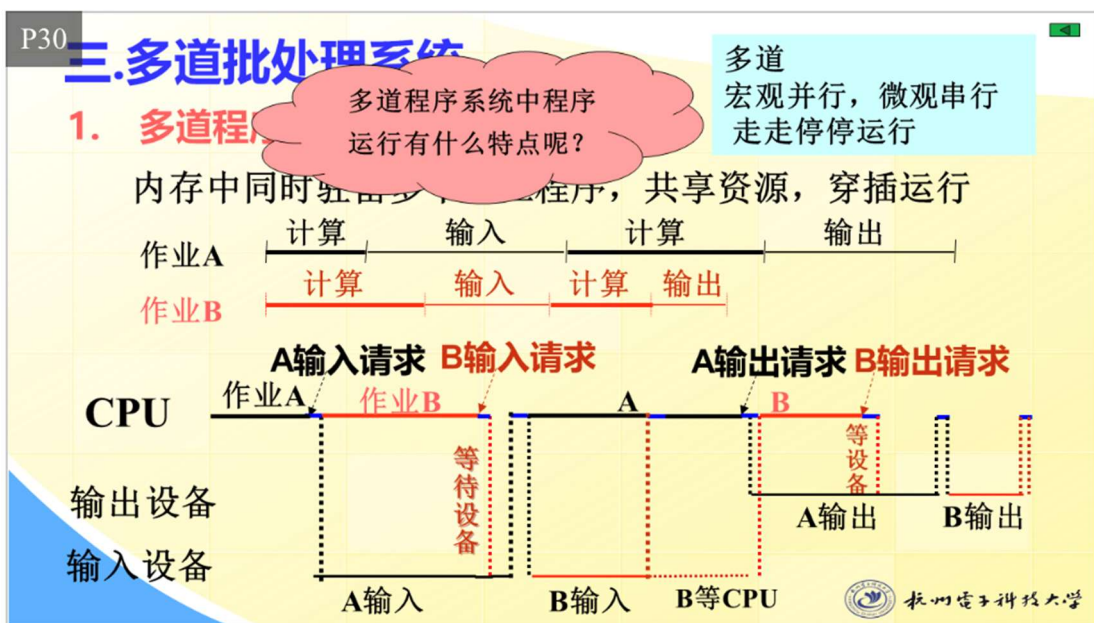
外设: I/O操作

- CPU和I/O设备串行工作

那怎么解决这个问题呢?

杭州电子科技大学

## • 多批道处理系统

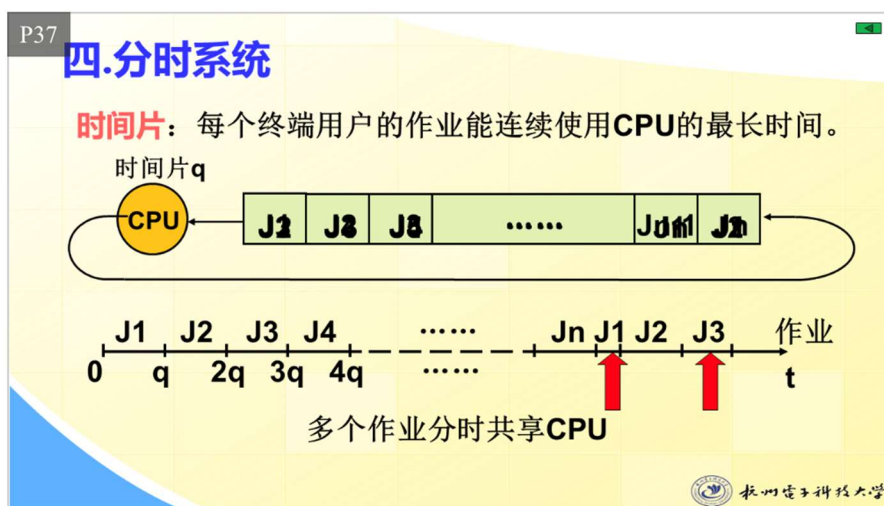
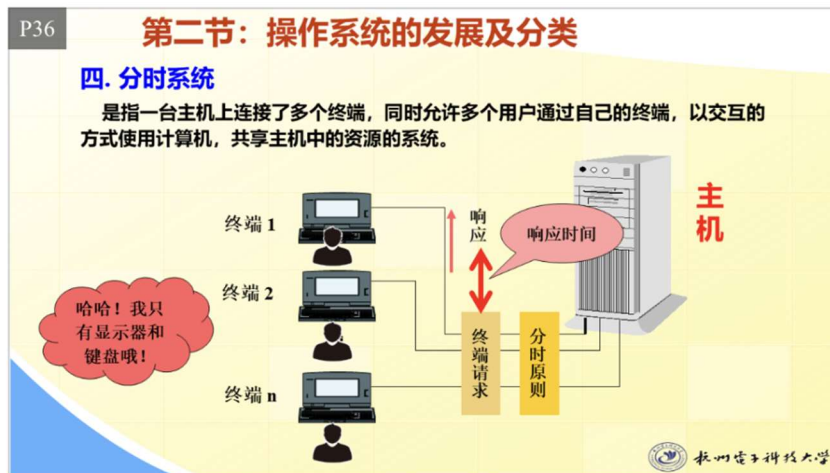


优点: 资源利用率高，系统吞吐量大。

缺点: 作业的平均周转时间长（周转时间: 指从作业装入系统开始，到运行完成并退出系统为止所经过的时间），并且无交互能力。



- 分时系统



特征：多路性；独立性；及时性；响应时间；交互性（最重要的特征之一）

- 实时系统

P49 1.2 操作系统的发展及分类

### 五. 实时系统

#### 1. 实时系统应用需求

- 实时控制：生产过程控制、武器控制、航空航天控制
- 实时信息处理：订票系统、情报检索系统

P50 1.2 操作系统的发展及分类

### 五. 实时系统

#### 1. 实时系统应用需求

- 实时控制：生产过程控制、武器控制、航天器控制

P51 1.2 操作系统的发展及分类

### 五. 实时系统

#### 1. 实时系统应用需求

- 实时控制：生产过程
- 实时信息处理：订票系统

P52 1.2 操作系统的发展及分类

### 五. 实时系统

#### 2. 实时系统概念

系统能及时响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行

概念：os 能及时响应外部事件的要求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行

特点：及时性和可靠性都很高，交互性比分时系统弱；

多路性；

独立性；

及时性；

典型的实时信息处理系统：机票订购系统、银行财务系统、情报检索系统。

• 微机操作系统

P56 1.2 操作系统的发展及分类

### 六. 微机操作系统的发展

用户数量	单用户，多用户
任务数量	单任务，多任务

- 单用户单任务操作系统  
DOS
- 单用户多任务操作系统  
Windows单机版
- 多用户多任务操作系统  
Unix、Linux  
Windows server

## • 网络操作系统

P57 1.2 操作系统的发展过程

### 七. 网络操作系统

(1) 计算机网络的定义

由线路将一组**独立自主**的计算机相互连接形成的一个集合体

(2) 计算机网络的特点

- 多个独立计算机
- 无公共内存
- 具备消息通信机制



杭州电子科技大学


P58 七. 网络操作系统

(3) 网络操作系统概念

- 一般操作系统功能模块
- 网络通信模块（通信接口中断处理程序、通信控制程序以及各级网络协议软件）

(4) 计算机网络的局限性

- 不能支持透明的资源存取
- 不能支持合作计算
- 不能对网络资源进行有效、统一的管理



杭州电子科技大学

## • 分布式操作系统

P59 1.2 操作系统的发展过程

### 八. 分布式操作系统

(1) 分布式系统概念

- 包含多个分布的通用资源部件，并经过通信网络相互作用
- 有一个分布式操作系统对资源进行全局和动态的管理控制
- 系统对用户是透明的
- 任务可以分布处理

(2) 分布式系统的特点

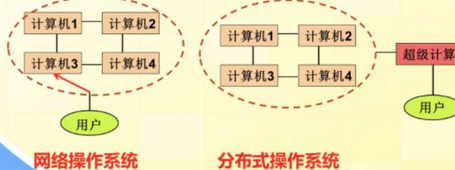
- 可扩展性
- 增加性能
- 高可靠性



杭州电子科技大学

P60 八. 分布式操作系统

(3) 分布式操作系统与网络操作系统的区别



网络操作系统      分布式操作系统

杭州电子科技大学

## • 嵌入式操作系统 EOS

P61 1.2 操作系统的发展过程

### 九. 嵌入式操作系统EOS

(1) 概念

- 将应用程序和OS与计算机硬件集成在一起的系统。
- 负责嵌入式系统的全部软、硬件资源的分配、调度工作、控制协调并发活动等。

(2) 嵌入式操作系统举例

- 手机操作系统：Android、iOS、黑莓等。
- VxWorks：是目前嵌入式系统领域中使用最广泛、市场占有率最高的系统。它支持多种处理器，如x86、1960、Sun Sparc、Motorola MC68xxx、MIPS RX000、POWER PC等等。大多数的VxWorks API是专有的。采用GNU的编译和调试器。

杭州电子科技大学

P62 九. 嵌入式操作系统

(2) 嵌入式操作系统举例

- **psOS**：该系统是一个模块化、高性能的实时操作系统，专为嵌入式微处理器设计，提供一个完全多任务环境，在定制或是商业化的硬件上提供高性能和高可靠性。可以让开发者根据系统的功能和内存需求定制成每一个应用所需的系统。开发者可以利用它来实现从简单的单个独立设备到复杂的、网络化的多处理器系统。
- **QNX**：是一个实时的、可扩充的操作系统，它提供了一个很小的微内核以及一些可选的配合进程。其内核仅提供4种服务：进程调度、进程间通信、底层网络通信和中断处理，其进程在独立的地址空间运行。所有其它OS服务，都实现为协作的用户进程，因此QNX内核非常小巧(QNX4.x大约为12Kb)而且运行速度极快。
- **LynxOS**：是一个分布式、嵌入式、可规模扩展的实时操作系统，支持线程概念；提供一些传统的、非实时系统的服务特征；包括基于调用需求的虚拟内存，一个基于Motif的用户图形界面，与工业标准兼容的网络系统以及应用开发工具。

杭州电子科技大学

## 1.2 操作系统的发展过程

2. 现有以下计算机的应用场合，请为其选择适当的操作系统：

- ① 航天器控制系统；
- ② 国家统计局数据处理中心；
- ③ 机房学生上机学习编程；
- ④ 锅炉炉温控制；
- ⑤ 两个不同地区之间发送电子邮件；
- ⑥ 产品组装流水线；
- ⑦ 核反应堆控制；
- ⑧ 自动洗衣机控制系统

小组讨论后  
回答

实时；分布；分时；实时；网络；实时；实时；嵌入

### 2. CPU 的利用率跟哪些相关

（CPU 利用率，最主要的就是说 CPU 它一开始我们操作系统发展的过程中，我们实际当中就是逐步在提升 CPU 的利用率。如果不要让它空闲，那么它这道题是一套综合性的题目。它会涉及到跟内存的关系，跟里面有多少个进程，然后磁盘的读写，以及文件的物理结构都会相关的。所以这道题也是一个关联性的题目）

### 3. 系统调用、中断与 CPU 分级

（系统调用中断和 CPU 的升级。我们大概这个就讲过大概两次，两节课到 3 节课。什么叫系统调用对吧？概念是什么？操作系统为了让用户使用自身的功能，使用操作系统的功能来完成对资源的访问。所以的话专门写了一些函数放在内核里，供用户通过系统调用函数的系统调用号来进行调用）



- 系统调用

概念：在用户程序与硬件设备之间添加了一个中间层。是一种用户在程序一级请求操作系统内核完成某种功能服务的过程调用，每种操作系统都会提供多达几百种的系统调用，每一个系统调用都是完成某种特定内核功能的一个函数。

P22

## 1.4 操作系统的用户接口

### 二. 程序接口：系统调用（以Linux/openEuler为例）

#### 1. 系统调用概念

系统功能调用是用户在程序一级请求操作系统服务的一种手段，它是带有一定功能号的“访管指令”。其功能是由操作系统中的程序完成的。



INT指令  
Dos: INT 21H  
Linux: INT 80H

trap指令

 杭州电子科技大学

系统调用和一般的过程调用（如三角函数  $\cos()$ ）区别：一般的过程调用调用程序与被调用过程运行在相同的状态——系统态或用户态，所以可直接由调用程序转向被调用过程；系统调用调用程序运行在用户态，被调用过程运行在系统态，不允许调用程序直接转向被调用过程，需要通过中断及陷入机制，先由用户态转换到系统态，经内核分析后，才能转向内核执行被调用过程。

系统调用号：os 为每个系统调用都赋予一个唯一的系统调用号

系统调用服务例程：每个系统调用都会完成操作系统内核的某项服务功能，具体是由一个特定的内核函数来实现的。



系统调用入口表：用于关联系统调用号及其对应服务例程的入口地址。

P25

## 二. 系统调用 (以Linux/openEuler为例)

### 5. 系统调用服务例程与系统调用入口表

0	common	read	sys_read
1	common	write	sys_write
2	common	open	sys_open
3	common	close	sys_close
56	common	clone	stub_clone
57	common	fork	stub_fork
58	common	vfork	stub_vfork
59	64	execve	stub_execve
60	common	exit	sys_exit

系统调用名称

系统调用号

64位系统还是32位系统

系统调用服务例程入口地址

杭州电子科技大学

系统调用处理程序：

P26

## 二. 系统调用 (以Linux/openEuler为例)

### 3. 系统调用处理程序：system\_call()

int \$0x80异常的处理程序，是所有系统调用的入口点

内核初始化时调用trap\_init()函数：设置中断描述符相应字段

```
01237: set_system_gate(SYSCALL_VECTOR, &system_call);
```

```
00031: #define SYSCALL_VECTOR 0x80
```

杭州电子科技大学

系统调用参数传递和返回值：

P27

## 二. 系统调用 (以Linux/openEuler为例)

### 4. 系统调用参数传递

- 寄存器传递：
  - eax：系统调用号
  - ebx, ecx, edx, esi和edi按照顺序存放前五个参数
- 对参数的要求：
  - ✓ 每个参数长度不能超过寄存器长度，即32位；
  - ✓ 参数个数实际不能超过5个；

在执行int \$0x80之前  
libc库中的封装例程设置

```
fork(){  
    |  
    int 0x80  
    |  
}
```

如果系统调用参数超出5个怎么办？

### 5. 系统调用返回值：

所有系统调用返回一个整数值：

- 正数或0表示系统调用成功结束
- 负数表示出错，返回的负值由封装例程存放在errno变量中返回给应用程序

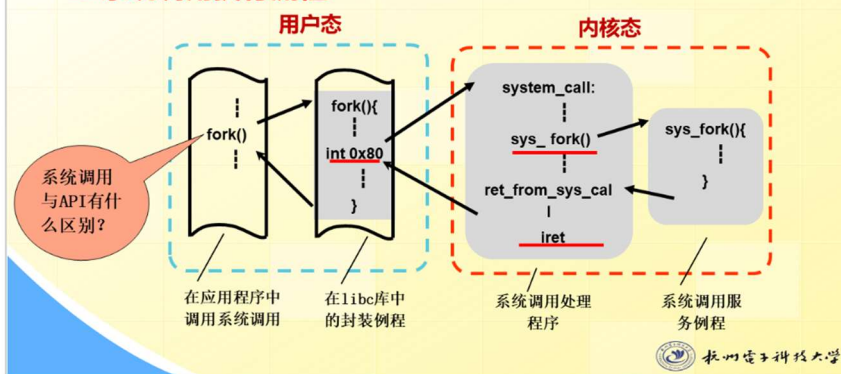
杭州电子科技大学

流程：

P23

## 二. 系统调用 (以Linux/openEuler为例)

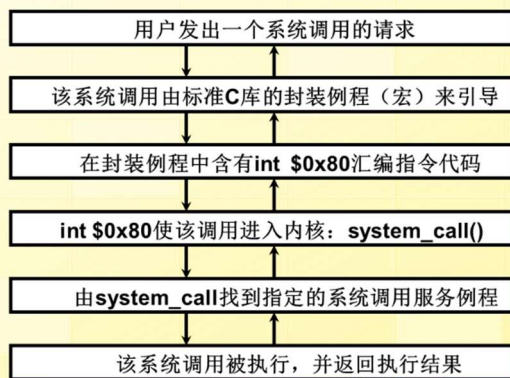
### 2. 系统调用执行流程



P28

## 二. 系统调用 (以Linux/openEuler为例)

### 6. 系统调用执行流程



### • cpu 分级

分为管态和目态

**管态:** 又称系统态、核心态。处理器在这种状态下，可以使用系统中所有资源、访问整个存储区，使用处理器的全部指令，包括一组特权指令。

**目态:** 又称用户态，是用户程序执行时处理器所处的状态，权限低，只能访问自己的存储区域，不能执行特权指令，不能直接取用系统资源，而只能向操作系统提出资源使用申请，由操作系统统一分配。

**特权指令:** 有特权权限的指令，如果使用不当，将导致整个系统崩溃。

如清内存、关闭系统、设置时钟、分配系统资源、修改虚存的段表和页表、修改用户的访问权限、管理设备的 I/O 指令。（读时钟日期不是特权指令）

P3

## 2.1.4处理机的特权级

1. 为什么要区分处理机的运行状态：保护操作系统

系统中两类程序区别

OS: 管理程序	用户程序
管理系统资源	使用资源, 提出申请
控制程序运行	被控制

2. 处理机的状态分类

管态 (系统态)	目态 (用户态)
操作系统的程序执行	用户程序执行
使用全部指令	禁止使用特权指令
使用全部系统资源 (包括整个存储区域)	只允许用户程序访问自己的存储区域

杭州电子科技大学

P4

## 2.1.4处理机的特权级

### 3. 特权指令集

- ① 涉及外部设备的输入/输出指令
- ② 修改特殊寄存器的指令: CPU状态寄存器PS、段地址寄存器、时钟寄存器、I/O设备控制寄存器等
- ③ 改变机器状态的指令: 停机指令、中断返回指令、禁止中断指令等

杭州电子科技大学

## 2.1.4处理机的特权级

### 4. 实例操作系统处理机的状态

- ① DOS系统  
不分态
- ② Windows 系统  
3环 用户态  
0环 系统态  
还有1、2环预留
- ③ UNIX系统、Linux系统  
00 管态 (系统态)  
11 目态 (用户态)

杭州电子科技大学