

# 操作系统原理与实践

---

授课教师：丁熠

电子邮箱：yi.ding@uestc.edu.cn



# 教师介绍

## 丁 熠 博士

- 职称：教授
- 研究方向：人工智能、医工结合、软件工程
- 办公室：沙河校区主楼中424
- Email: yi.ding@uestc.edu.cn
- 最新课件下载及作业提交：

电子科技大学网络学堂

<https://study.uestc.edu.cn/wlxt/index.aspx>



# 课程介绍

- 操作系统课程是计算机教育的**必修**内容
- 计算机专业的**核心**课程
- 计算机技术开发的**理论**基础
- 计算机相关专业的**考研**科目



# 课程目标

- C01：要求学生从系统级上**掌握分析和设计计算机操作系统的基本思路**，对内核、进程、线程等核心概念具有系统级的认识与理解。
- C02：要求学生**理解操作系统的设计原理、框架、目标、各部分的工作原理**，能够从系统级上应用内核、进程、线程等核心概念对操作系统进行设计并改进等。
- C03：要求学生能够**理解各类操作系统中的经典问题和典型应用**，并搭建Linux环境，开展相应的实验，实现这些经典问题及应用。
- C04：要求学生理解计算机操作系统的关键功能，并搭建Linux环境，开展相应的实验，实现这些关键功能，**培养系统能力和解决复杂工程问题的能力**。



# 课程指标点

- GR1.3：理解系统的概念及其在软件工程领域的体现，能对复杂软件工程问题的解决方案进行分析，并尝试改进
- GR4.1：能够采用科学的方法对复杂工程问题中的关键环节，设计相应的实验方案，搭建实验环境，开展实验



# 课程介绍







# 考核方式

# 课程要求

## ■ 理论和综合应用

基础课程

考研课程

应用价值

## ■ 成绩构成

期末考试占70%

考勤和作业占10%

中期考试占10%

实验占10%

## • 课后作业

作业每章提交一次（本章结束后的二周内提交到网络学堂）

## • 实验

实验报告期末统一提交

## • 学习方法:

在学习每一部分内容时,不仅要记住给出的结论,还要认真思考所讨论问题的由来、环境、意义、理论依据和应用背景,并结合实例操作系统加深理解,做到举一反三。



# 教材&参考书籍

## ■ 课程教材：

- 计算机操作系统，汤小丹、汤子瀛等,西安电子科技大学出版社（第四版）
- Linux操作系统应用编程，刘玓、陈佳、肖堃、丁熠，人民邮电出版社

## ■ 参考书目：

- 现代操作系统，Andrew S.Tanenbaum，  
陈向群译，机械工业出版社，2009.7
- Operating System Internals and Design principles, William Stallings , TsingHua University Press
- Applied Operating System Concepts , Abraham Silberschatz *etc.* , Higher Education Press





# 课程构成

▪ 总学时：80（上课56学时、上机24学时）

▪ 第一章	操作系统引论	2学时
▪ 第二章	进程管理	10学时
▪ 第三章	处理器调度与死锁	8学时
▪ 第四章	存储器管理	10学时
▪ 第五章	输入输出系统	2学时
▪ 第六章	文件及文件系统	4学时
▪ 第七章	UNIX/Linux系统入门	2学时
▪ 第八章	shell的交互功能与shell程序设计	4学时
▪ 第九章	文件操作与权限管理	4学时
▪ 第十章	进程运行与监控	2学时
▪ 第十一章	操作系统接口与应用开发	6学时
▪ 第十二章	线程与线程控制	2学时



# 讨论

- 你知道**哪些操作系统**的名字?
- 你觉得**操作系统**是什么?



# 讨论

## ■ 你知道哪些操作系统的名字？



Windows旗帜



Linux企鹅



MAC OS



Ubuntu环



Opensuse变色龙



Sun的Solaris



谷歌的Android

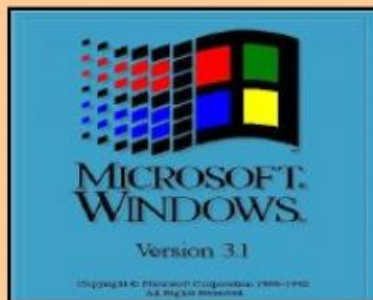


# 讨论

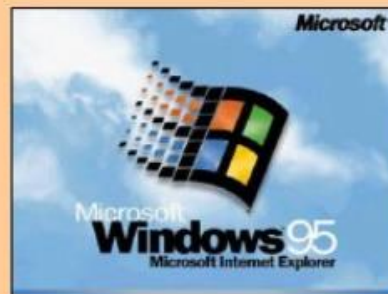
## ■ 微软家族



Window 1.0



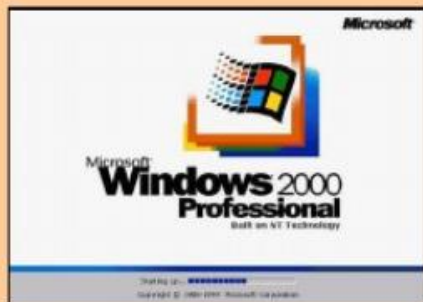
Windows 3.1



Windows 95



Windows 98



Window 2000



Windows XP



Windows Vista



Windows 7



# 讨论

- 熟知的操作系统
  - UNIX
    - Unix是一个多用户、多任务操作系统。最早由贝尔实验室编写
    - 73年使用C语言重写内核





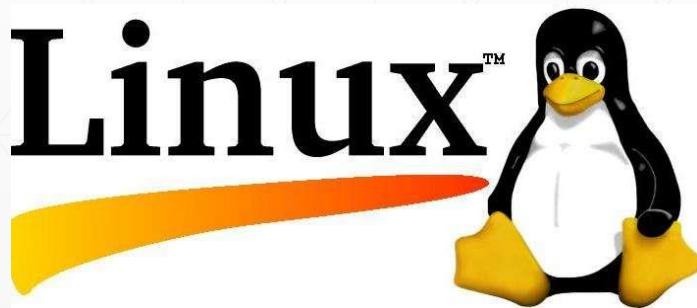


# 讨论

## ■ 熟知的操作系统

### ■ LINUX

- Linux操作系统诞生于1991年10月5日
- Linux是一套免费使用和自由传播的类UNIX操作系统，是一个多用户、多任务、支持多线程和多CPU的操作系统。



redhat.







# 讨论

## ■ iOS



## ■ Android





# 讨论

- 你觉得操作系统是什么?
- **当前对于OS还没有一个统一的、适用的定义!**



www.BANDICAM.com



00:00 / 00:58

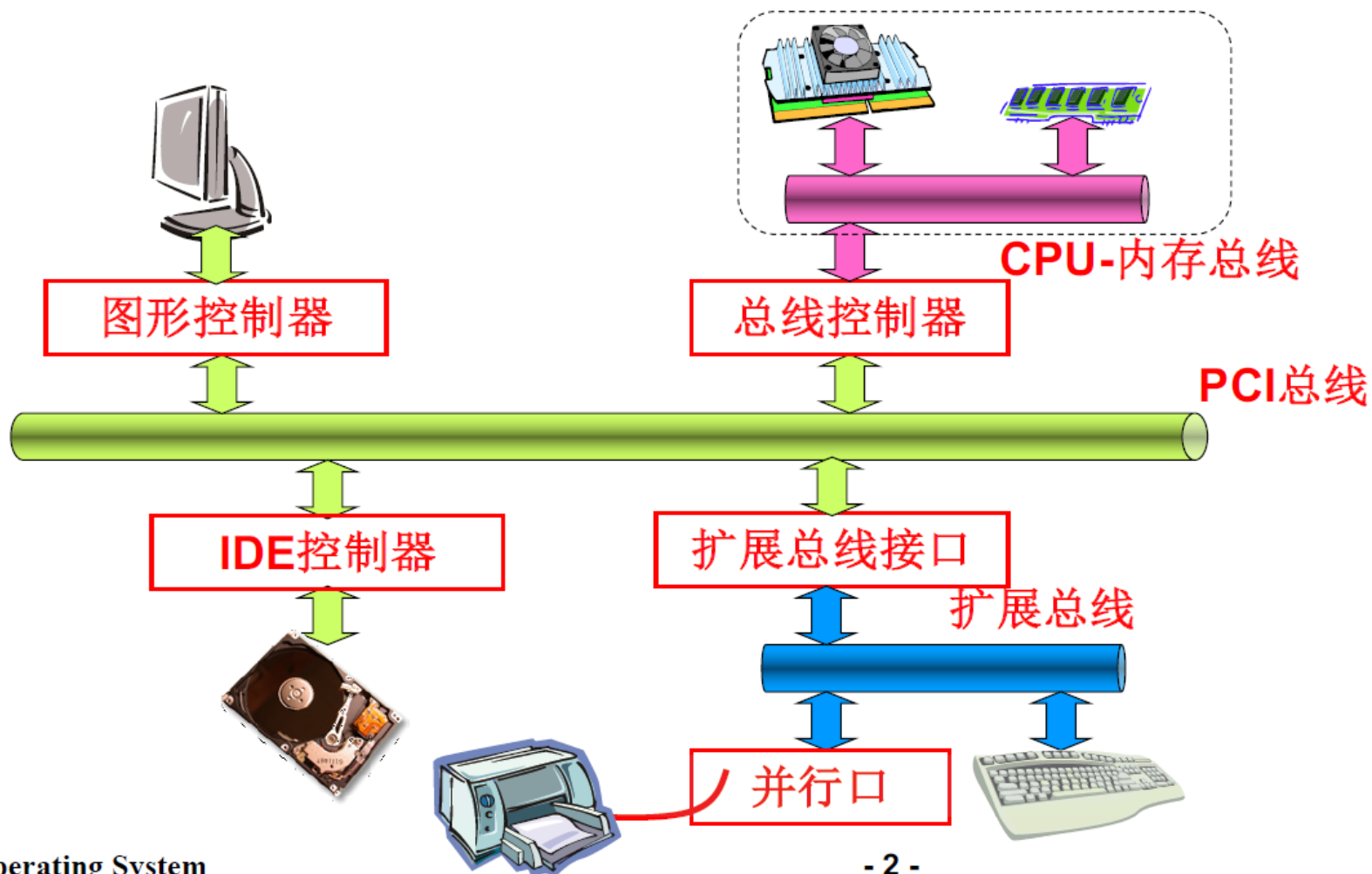
倍速

超清





# 这是什么?

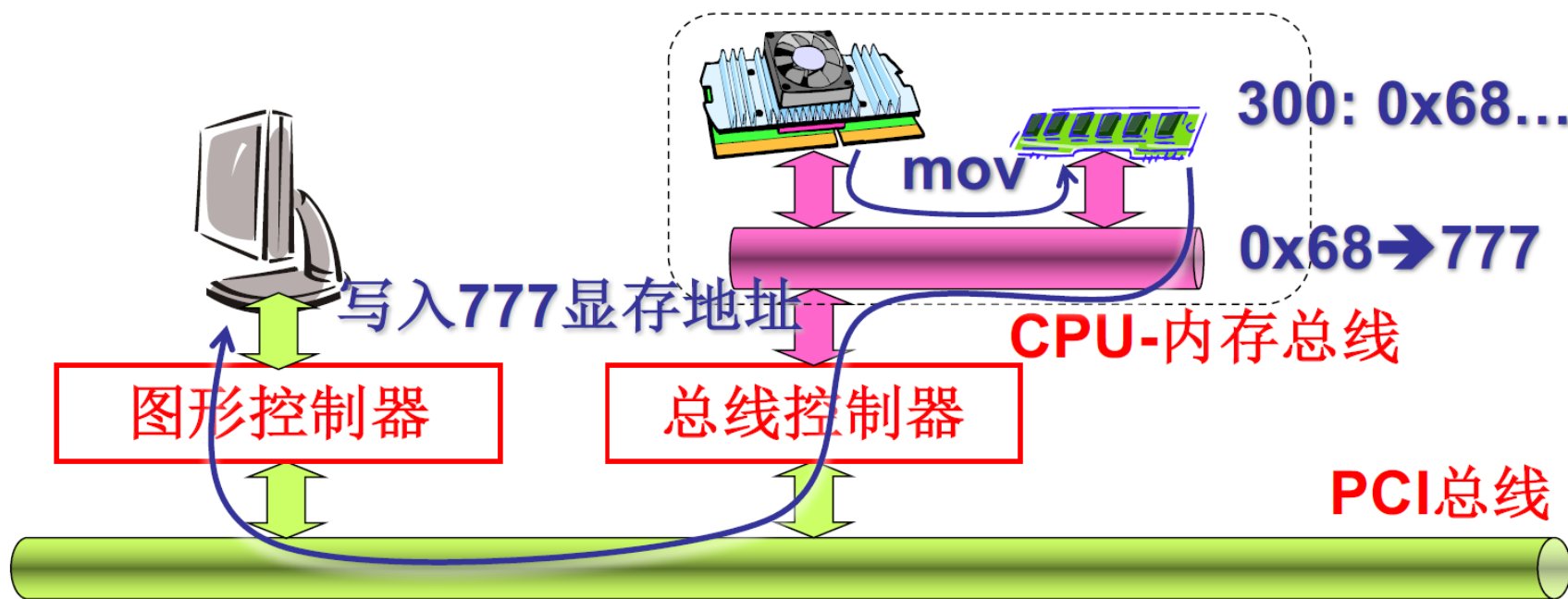






# 用计算机做什么？

- 计算机有了那就解决这个问题吧：屏幕上输出“hello!”

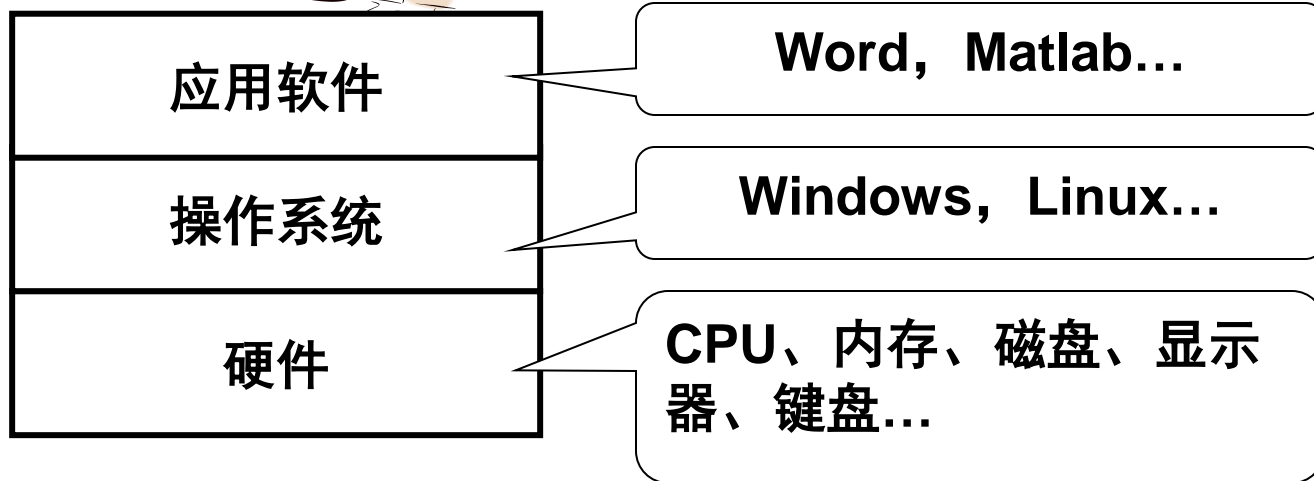




# 安装了操作系统的计算机



```
printf( "hello!" )
```

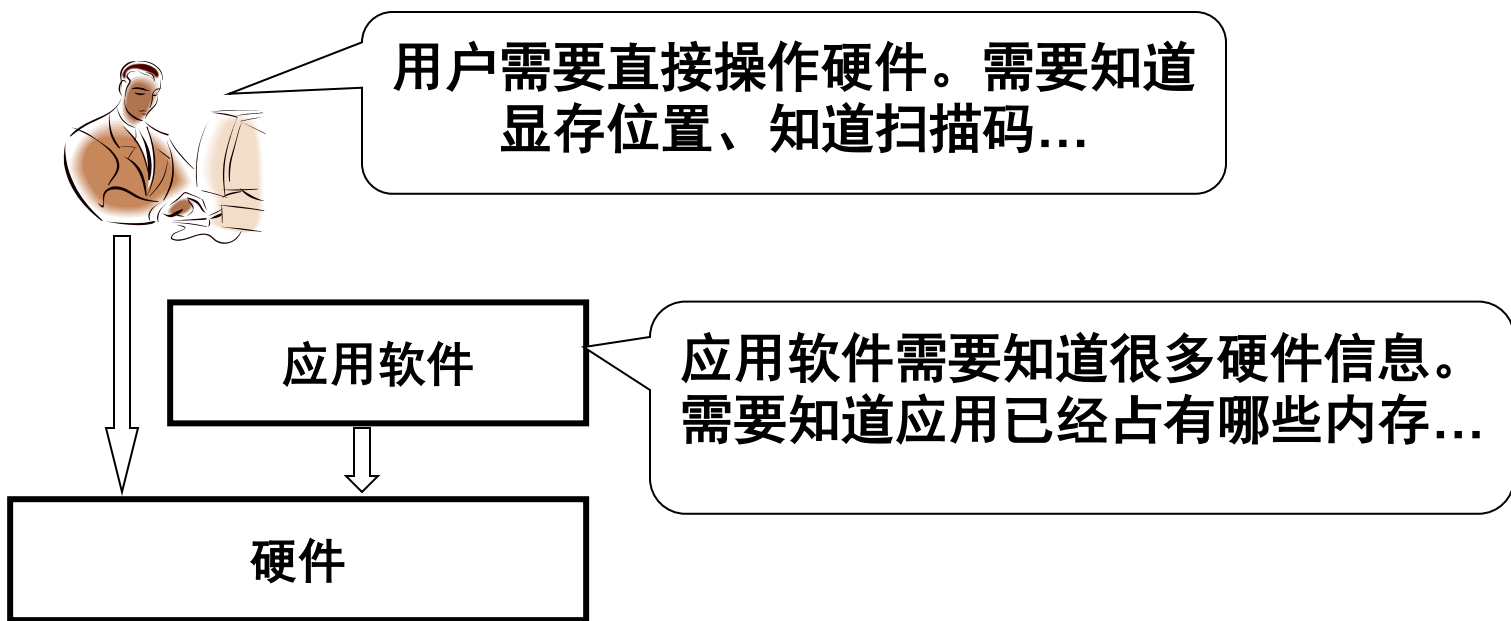






# 为什么操作系统放在那里？

## ■ 如果那里没有操作系统



- 操作系统可以：屏蔽细节、统一管理硬件、防止违法操作...

使计算机的使用更加方便、简单、高效...



# 什么是操作系统?



应用软件

操作系统

硬件

- 是计算机硬件和应用之间的一层软件
  - 方便我们使用硬件，如使用显存...
  - 高效的使用硬件，如打开多个终端
- 管理哪些硬件?

CPU管理

内存管理

终端管理

磁盘管理

文件管理

网络管理

电源管理

多核管理



# 操作系统要做什么？



应用软件

操作系统

硬件



## ■ 管理所有硬件资源

- 冯·诺依曼认为，计算机由五大部件组成:输入设备、输出设备、存储器、运算器、控制器

- OS需要管理CPU(运算器、控制器)

进程管理

- OS需要管理memory(内存)

内存管理

- OS需要管理disk(外存)

文件系统

- OS需要管理IO(输入/输出设备)

I/O系统



# 操作系统要做什么？



## ■ 向用户和应用软件提供使用接口

- 用户输入ls或dir时，OS需要将当前目录下的文件列出
- 应用程序调用malloc()时，OS需要分配内存

## ■ 监控用户对计算机的使用

等等.....

- 用户按下ctrl+c时，该用户的当前任务将被kill
- 用户写一个文件时，OS需检查是否有权限



## “操作系统做什么”是动态变化的

三种场合下OS所做的工作会有很大差别...

### ■ 操作系统的任务会随环境而变化

- 如实时操作系统—任务响应需满足一定的时限要求
- 某些场合要求很严格的时限，如导弹控制 **硬实时**
- 某些场合要求不能太久，如键盘响应 **软实时**
- 某些场合没有时限要求，如屏保 **不需要考虑实时**

### ■ 操作系统的任务会随时间而变化

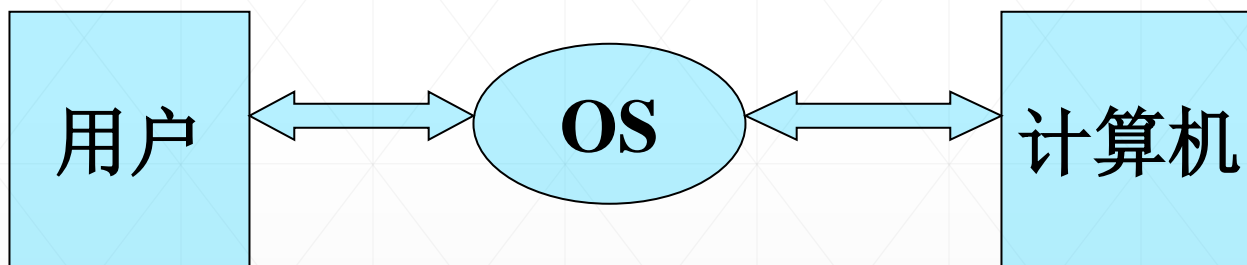
- Moore定律表明：设备体积迅速变小、能力迅速增强
- 出现了嵌入式设备和嵌入式操作系统

对EO能源管理模块很重要



# 讨论：操作系统的概念

- **操作系统定义：**操作系统是一组控制和管理计算机软硬件资源、合理地各类作业进行调度以及方便用户使用的程序集合。



**OS是计算机系统的灵魂！**





# 其它定义 (汤 第三版)

- **操作系统定义 (Operating System, OS) :** 操作系统是位于硬件层(HAL)之上, 所有其它系统软件层之下的一个**系统软件**, 使得管理系统中的各种软件和硬件资源得以充分利用, 方便用户使用计算机系统。





# 第一章 操作系统引论

**1.1 操作系统的目标和作用**

**1.2 操作系统的发展过程**

**1.3 操作系统的基本特征**

**1.4 操作系统的主要功能**

**1.5 操作系统的结构设计**



# 1.1.1 操作系统的目标

- **方便性:**操作系统使计算机更易于使用。
- **有效性:**操作系统允许以更有效的方式使用计算机系统资源。
  - 提高系统资源利用率
  - 提高系统的吞吐量
- **可扩展性:**在操作系统中,允许有效地开发, 测试和引进新的系统功能。
- **开放性:**实现应用程序的可移植性和互操作性,要求具有统一的开放的环境。



这两个目标  
最重要



## 1.1.2 OS的作用

### 1. OS作为用户与计算机硬件系统之间的接口

- 计算机用户需要的**用户命令**：由OS实现的所有用户命令所构成的集合常被人们称为OS的**Interface(用户接口)**；有时也称为**命令接口**。

**命令的表示形式：**

- 字符形式：较灵活但因繁琐而难记；
  - 菜单形式：试图在字符终端上提供友好的用户界面
  - 图形形式：因直观而易记但不灵活。
- 应用软件需要的**System Call(系统调用)**：由OS实现的所有系统调用所构成的集合被人们称为**程序接口**或**应用编程接口(Application Programming Interface, API)**。



## 1.1.2 OS的作用

### 1. 作为用户与计算机硬件系统之间的接口

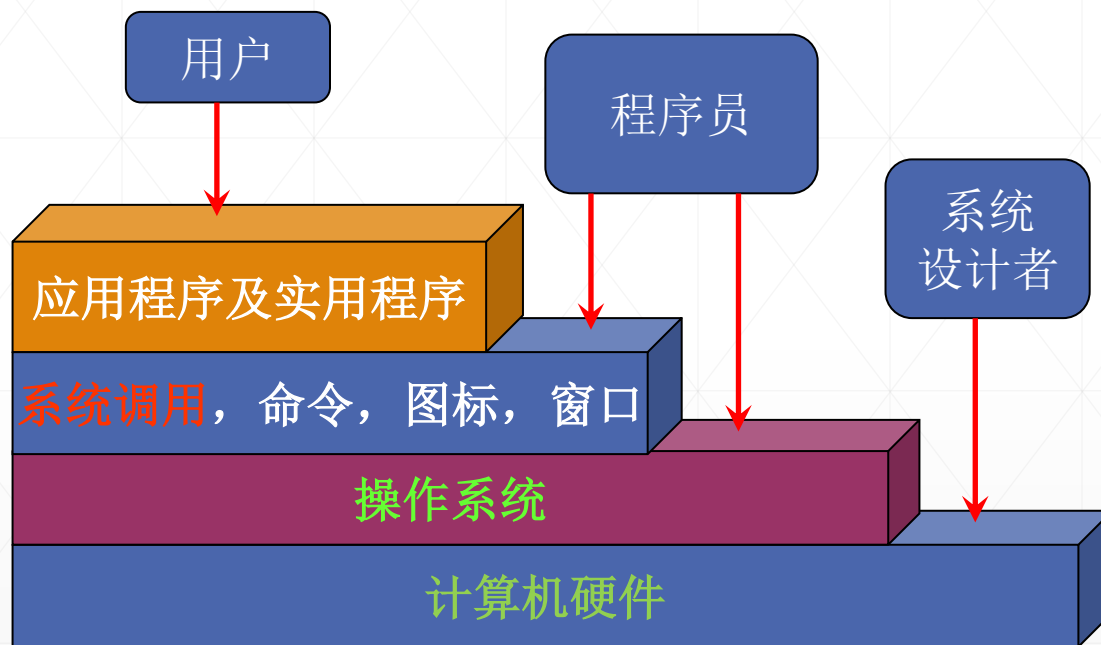


图1-2 OS作为接口的示意图



```
cst:/home/lizhijun# ./output "hello"
ECHO:hello
cst:/home/lizhijun# ./output "OS students"
ECHO:OS students
```

## ■ 命令是什么？命令输入后发生了什么？

```
#include <stdio.h>
int main(int argc, char * argv[])
{ printf("ECHO:%s\n", argv[1]); }
```

output.c程序，经过gcc编译后得到一个可执行文件

```
gcc -o output output.c
```

也是一段程序：shell，即 /bin/sh

```
./output "hello"
```

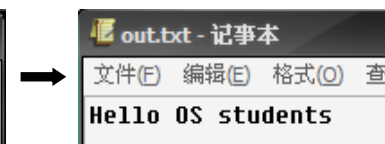
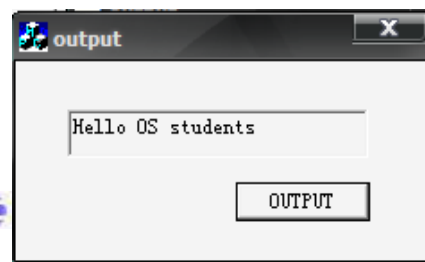
```
int main(int argc, char * argv[])
{ char cmd[20];
  while(1) { scanf("%s", cmd);
              if(!fork()) {exec(cmd);}
              else {wait();} } //while(1)
}
```

■ 命令得以工作的原因：  
OS提供了printf,fork,  
exec等函数(接口)

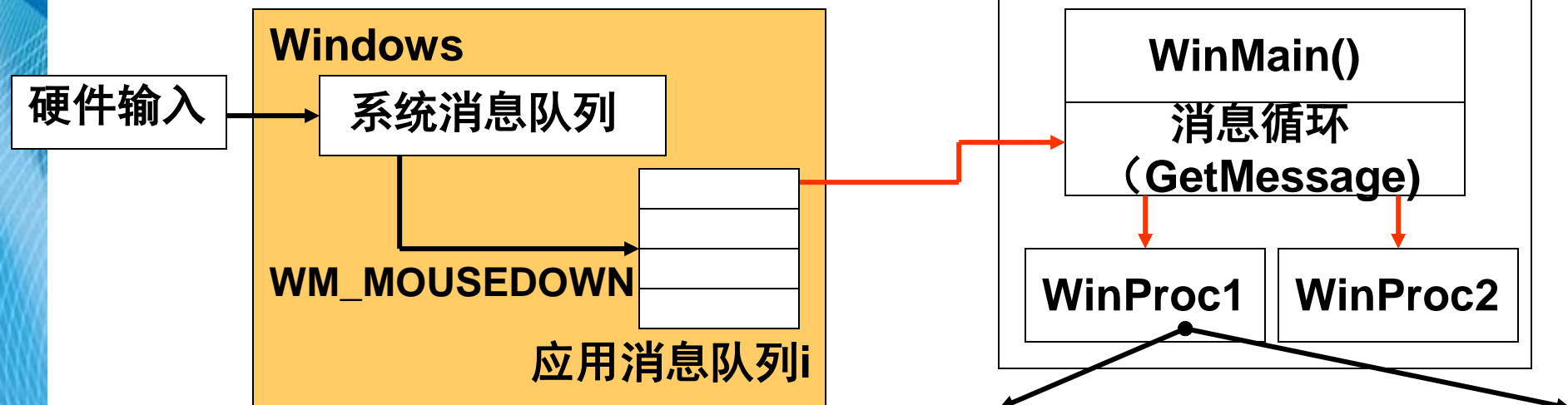




# GUI



## ■ 鼠标点击OUTPUT以后...



```
void COutputDlg::OnOK()  
{ GetDlgItemText(IDC_EDIT1,m_outStr);  
  FILE *fp=fopen("d:\\out.txt","w");  
  fprintf(fp,m_outStr,m_outStr.GetLength());  
  fclose(fp); }
```

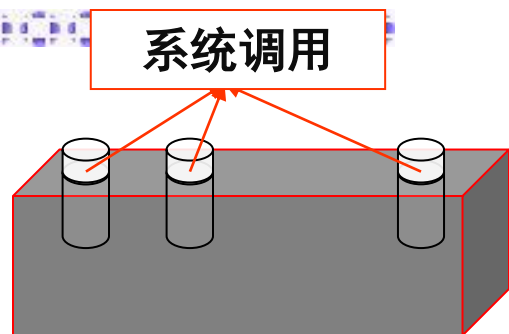
```
OUTPUT按钮的WinProc()  
{case WM_MOUSEBUTTONDOWN:  
  OnOK(); }
```

- 使GUI工作，OS提供消息框架和相关API



# 接口总结 系统调用

- 命令行: 命令程序+shell+字符显示
- GUI: 消息处理程序+消息框架+图形显示
- 应用程序: 将上述部分组成一个整体...



■ 用户使用计算机总结: 应用操作系统提供的接口  
编写程序; 应用这些程序解决问题

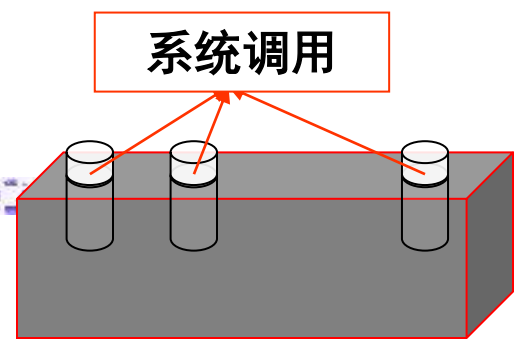
- 用户通过OS接口使用计算机; OS接口影响计算机的  
使用方式

OS提供了一些重要函数, 这就是接口。接口表现为函数调用, 又由OS提供, 所以又称为系统调用。

- 系统调用是学习操作系统的首要任务...



# 提供哪些系统调用?



- **POSIX: 将接口统一比较好!**
  - **Portable Operating System Interface of Unix**
  - **由IEEE开发, 是一个标准族: 1003.1, 2003...**

分类	POSIX定义	描述
任务管理	fork	创建一个进程
	execl	运行一个可执行程序
	pthread_create	创建一个线程
文件系统	open	打开一个文件或目录
	EACCES	返回值, 表示没有权限
	mode_t st_mode	文件头结构: 文件属性
	S_ISDIR(m)	宏展开, 是否为目录文件



## 1.1.2 OS的作用

- 操作系统应**隐藏**复杂的、困难的、丑陋的、特殊的硬件细节，呈现给程序（程序员）良好、清晰、优雅、一致的抽象

自顶向下的  
观点

- 操作系统的任务就是创建好的抽象，并实现和管理它所创建的抽象



## 1.1.2 OS的作用

### 2. OS作为计算机系统**资源的管理者**(软硬件资源)

- ① 处理机管理， 用于分配和控制处理机；
  - ② 存储器管理， 主要负责内存的分配与回收；
  - ③ I/O设备管理， 负责I/O设备的分配与操纵；
  - ④ 文件管理， 负责文件的存取、共享和保护。
- 对于**多用户系统**， 需要管理共享资源， 避免发生冲突。



## 1.1.2 OS的作用

- 操作系统的任务在**相互竞争**的程序之间**有序**地控制对硬件设备的分配

自底向上的  
观点

- 资源管理包括用以下两种不同的方式实现**多路复用（共享）**资源
  1. **在时间上复用**：当一种资源在时间上复用时，不同的程序“轮流”实用它；例如CPU、打印机等
  2. **在空间上复用**：每个客户都得到资源的一部分；例如内存、磁盘等





## 1.1.2 OS的作用

### 3. OS用作扩充机器

- **裸机：完全无软件的计算机系统。**
- 操作系统用作扩充机器功能，使其便于使用
- 在裸机上覆盖管理软件，实现对设备的操作，并向上提供一组操作命令。
- 隐藏对设备操作的具体细节，实现对硬件操作的多个层次的抽象。
- 通常把覆盖了软件的机器称为扩充机器或虚机器。

## 1.2 操作系统的发展过程

- 无操作系统
  - 单道批处理操作系统
  - 多道批处理操作系统
  - 分时操作系统(time-sharing system)
  - 实时操作系统(real time system)
-



# 1.2.1 未配置操作系统的计算机系统

## 1、人工操作方式

**一台计算机的所有资源由用户独占，降低了计算机资源利用率，人操作慢，出现了严重的人机矛盾。**

**缺点：**

**(1)用户独占全机**

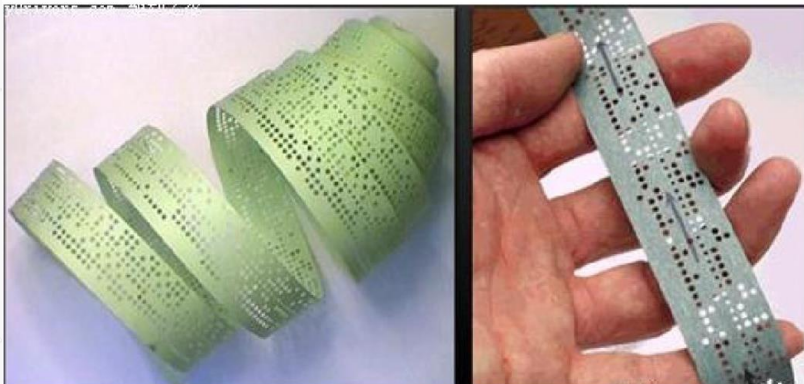
**(2)CPU等待人工操作**



## 1.2.1 未配置操作系统的计算机系统

### 1、人工操作方式

输入输出：穿孔卡片



### ENIAC计算机

运算速度：1000次/每秒, 数万个真空管, 占地100平方米





## 人工操作方式缺点:

机器速度	作业在机器上计算所需时间	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

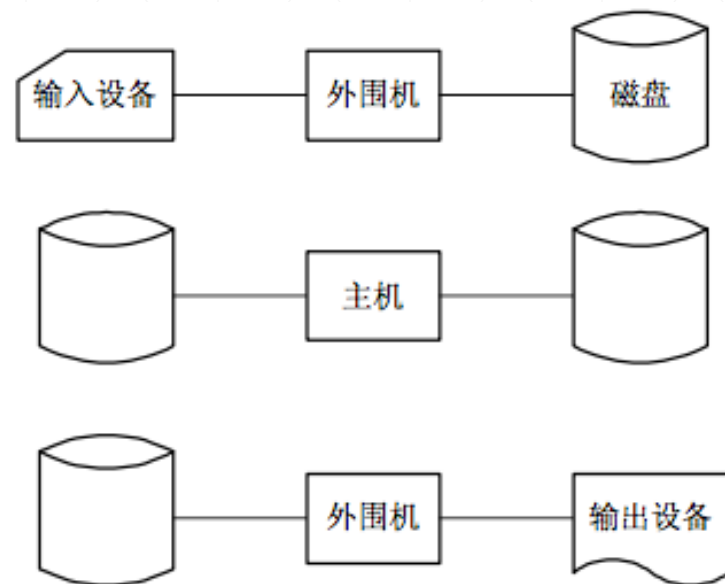
- **主要矛盾:**  
人机矛盾、CPU和I/O设备速度不匹配
- **怎么解决?**
  - **提高效率的途径: 脱机I/O、批处理**



## 1.2.1 未配置操作系统的计算机系统

### 2、脱机输入输出方式

- 在外部计算机的控制下，实现输入输出。
- 主要解决了CPU与设备之间不匹配的矛盾，提高了I/O速度



脱机 I/O 示意图





## 1.2.2 单道批处理系统

50年代末 ~ 60年代中（晶体管）

把一批作业以脱机输入方式输入到磁带/磁盘

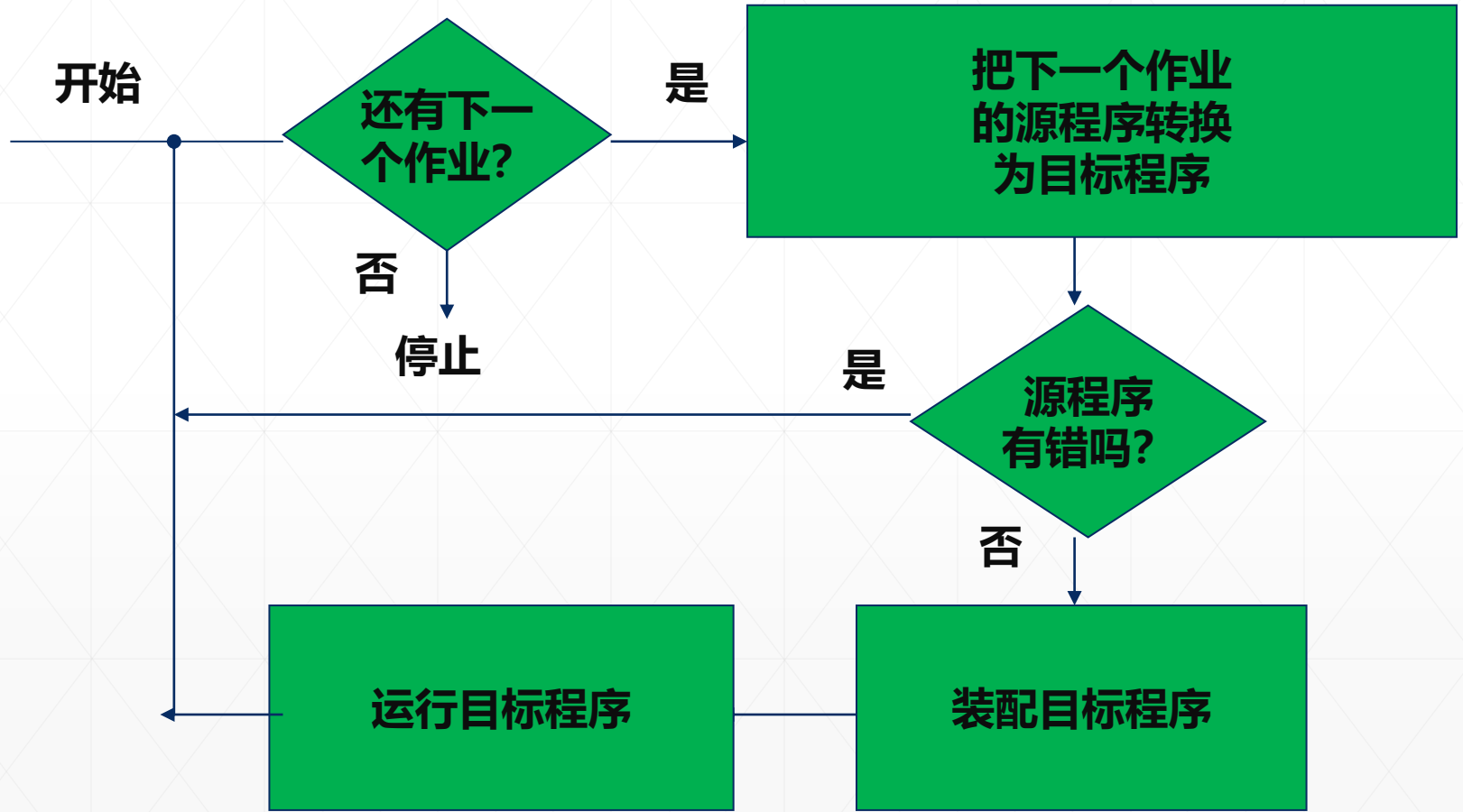
利用磁带或磁盘把任务分类编成作业顺序执行

每个批作业由专门监督程序（Monitor）自动依次处理

- 系统对作业的处理都是成批进行的，且在**内存中始终仅存一道作业运行**，运行结束或出错，才自动调另一道作业运行，故称为**单道批处理系统**。

# 单道批处理系统的处理流程

一批作业输入磁带





## 1.2.2 单道批处理系统

- 1、**单道批处理系统主要特征**：自动性、顺序性、单道性。
- 2、单道批处理系统**主要优点**：减少人工操作，解决了作业的自动接续。
- 3、单道批处理系统**主要缺点**：平均周转时间长，没有交互能力。



## 1.2.3 多道批处理系统

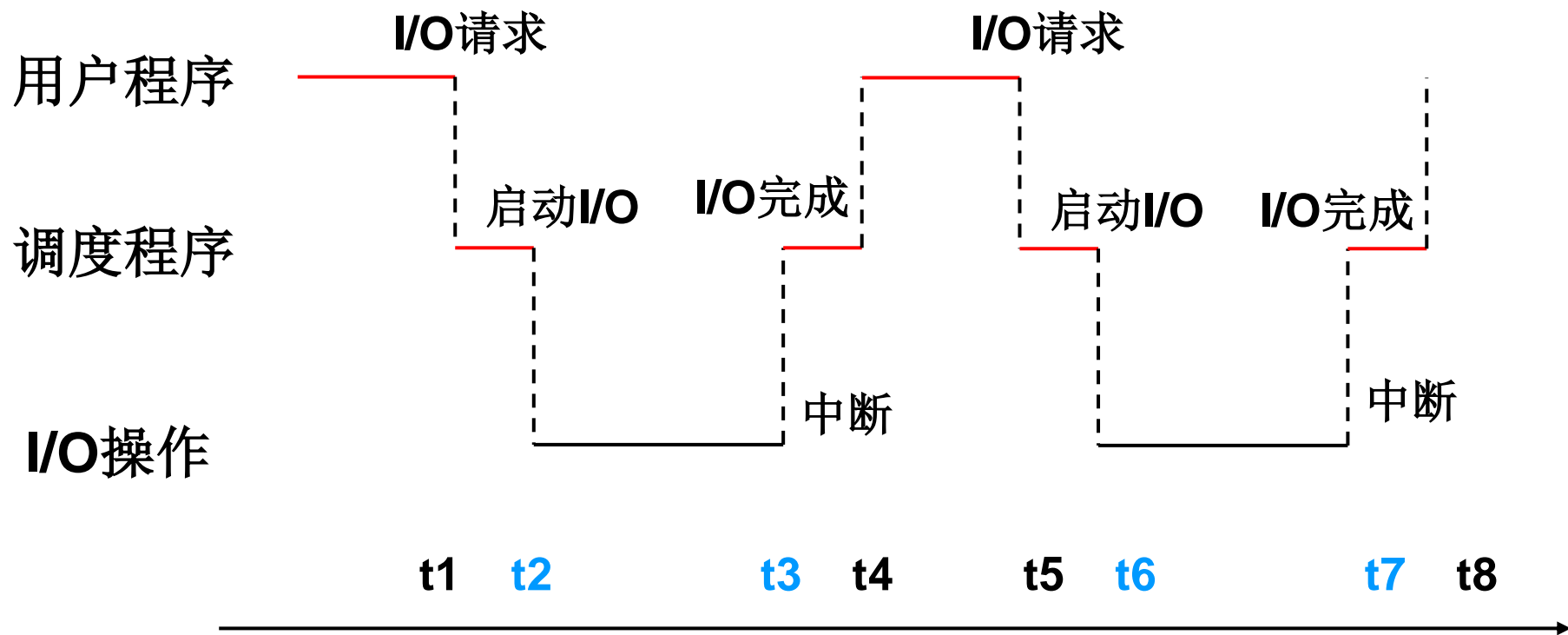
一、**多道程序的概念**：在内存中存放多道作业运行，运行结束或出错，自动调度内存中的另一道作业运行。

●**多道程序带来的好处**：

- 1、提高CPU的利用率。
- 2、提高内存和I/O设备利用率。
- 3、增加系统吞吐率。

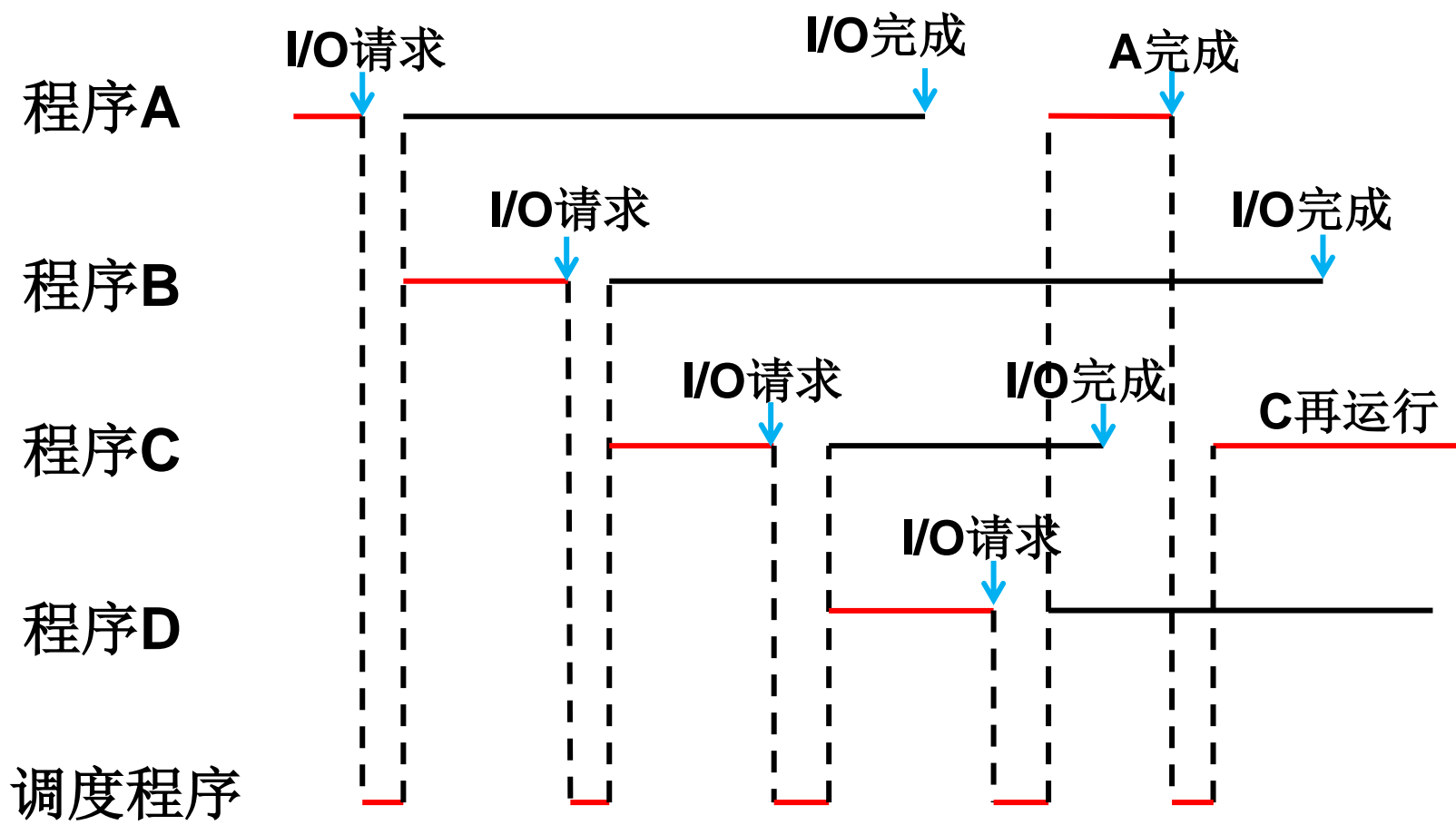
IBM S/360





表示获得CPU

(a)单道程序运行情况



—— 表示获得CPU

—— 表示I/O操作

(b)四道程序运行情况





## 举例:

- 设内存中有三道程序A、B和C，它们按A、B、C的优先次序执行。它们的计算和I/O操作时间如下表所示。

假设三道程序使用能够相同的设备进行I/O操作，即程序以串行方式使用设备，试画出单道运行和多道运行的时间关系图（调度程序的执行时间忽略不计）。在两种情况下，完成这三道程序各要花多少时间？

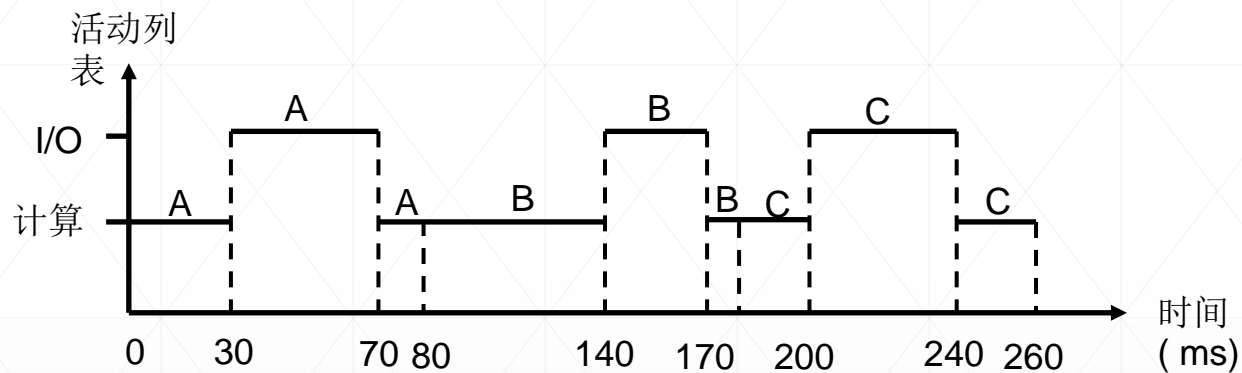
	A	B	C
计算	30ms	60ms	20ms
I/O	40ms	30ms	40ms
计算	10ms	10ms	20ms



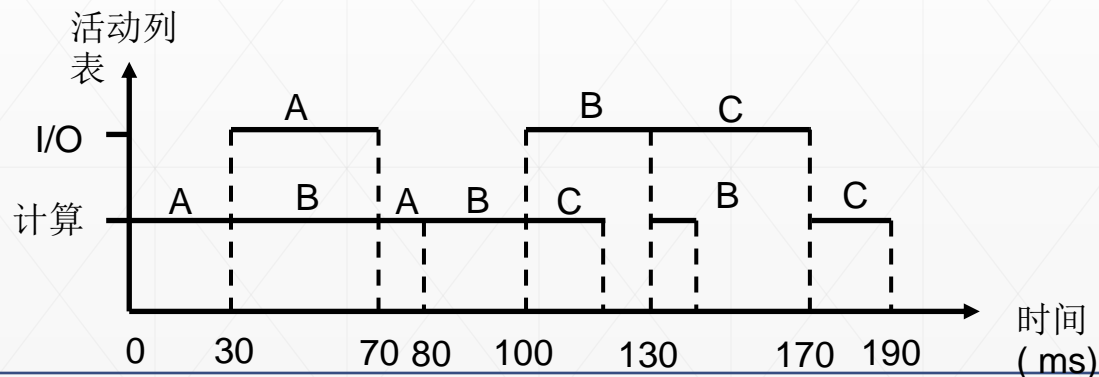
# 举例:

	A	B	C
计算	30ms	60ms	20ms
I/O	40ms	30ms	40ms
计算	10ms	10ms	20ms

## ■ 单道运行的甘特图



## ■ 多道运行的甘特图

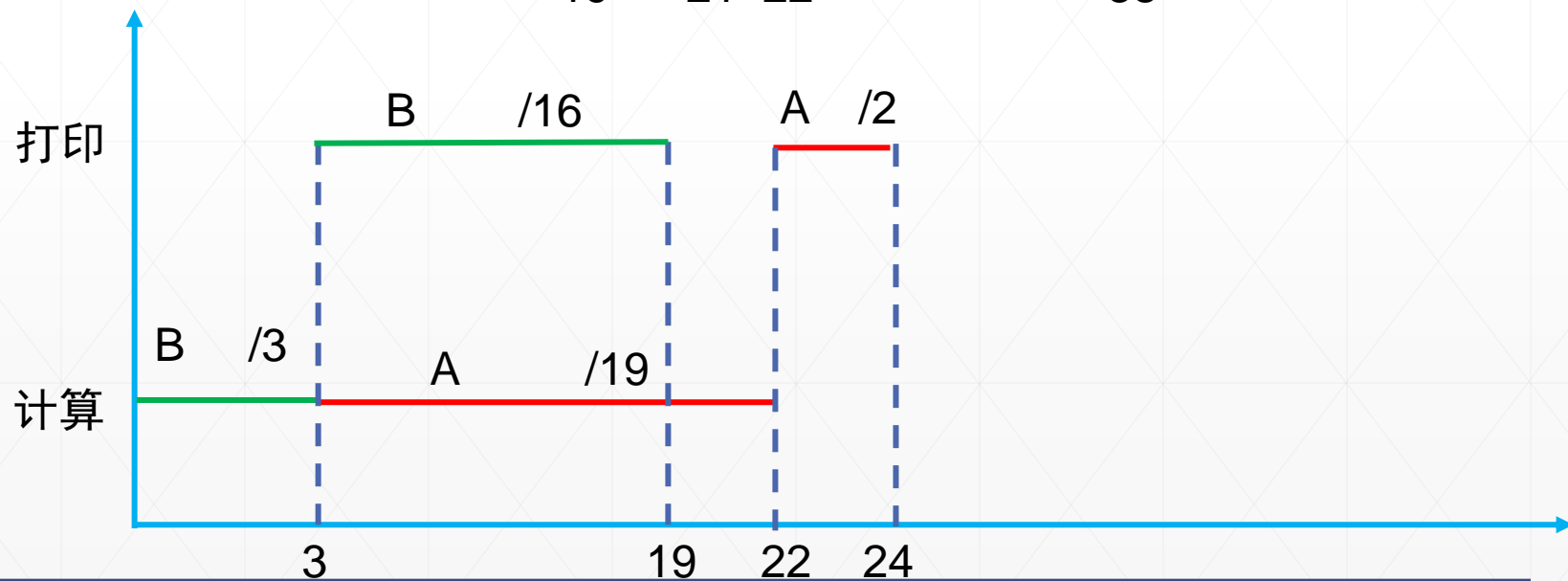
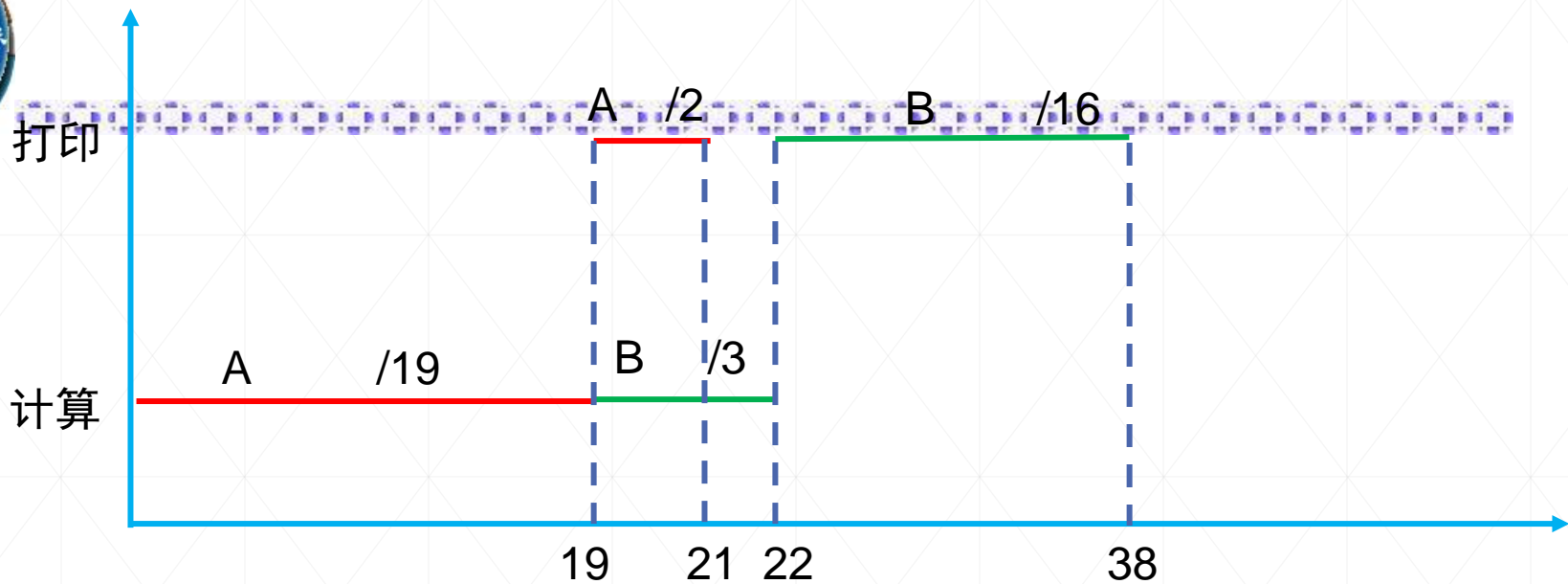




## 课堂练习:

- 现有两道作业，其中一道可能存在的操作有：计算19分钟，打印2分钟。另一道可能存在的操作有：计算3分钟，打印16分钟。那么在单道程序系统中，两道作业的执行总时间至少为多少分钟？而在多道程序系统中，这一时间至少为多少分钟？

40  
24





## 1.2.3 多道批处理系统

### 二、多道批处理系统主要特征：

多道性、无序性、调度性（进程调度和作业调度）

### 三、多道批处理的主要优点：提高了资源利用率和吞吐能力

多道批处理的**主要缺点**：平均周转时间长，没有交互能力

### 四、多道批处理系统需要解决的5个问题

- 1、处理机管理：分配和控制CPU
- 2、存储器管理：内存分配与回收
- 3、I/O设备管理：I/O设备的分配与操纵
- 4、文件管理：文件的存取、共享和保护
- 5、作业管理：如何组织作业运行



## 1.2.4 分时操作系统

### 一、分时系统的产生

用户需要：人机交互、共享主机、便于用户上机

### 二、分时系统实现中的关键问题：

及时接收：实现多个用户的信息及时接收。

及时处理：及时控制作业的运行。

### 三、分时系统实现的方法

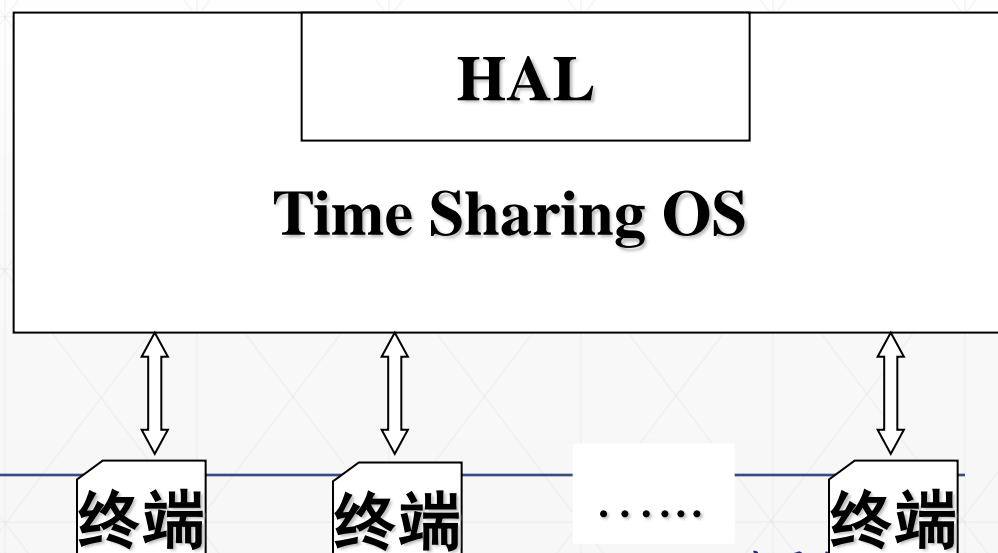
- 简单分时系统
- 具有“前台”和“后台”的分时系统
- 多道分时系统





## 1.2.4 分时操作系统

- 特点：
  - 多路性：一个主机与多个终端相连；
  - 独立性：彼此独立操作，互不干扰；
  - 及时性：系统能在很短的时间得到回答；
  - 交互性：能实现人机对话（区别于批处理系统）；
- 典型系统：
  - Multics (MIT)
  - UNIX
  - Linux

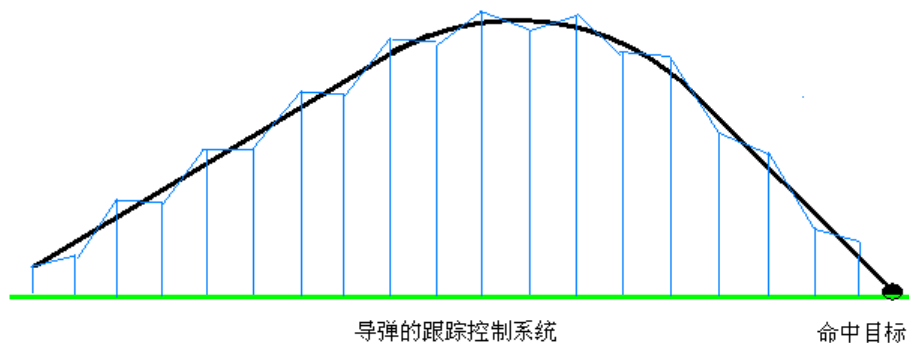


分时处理  
终端请求



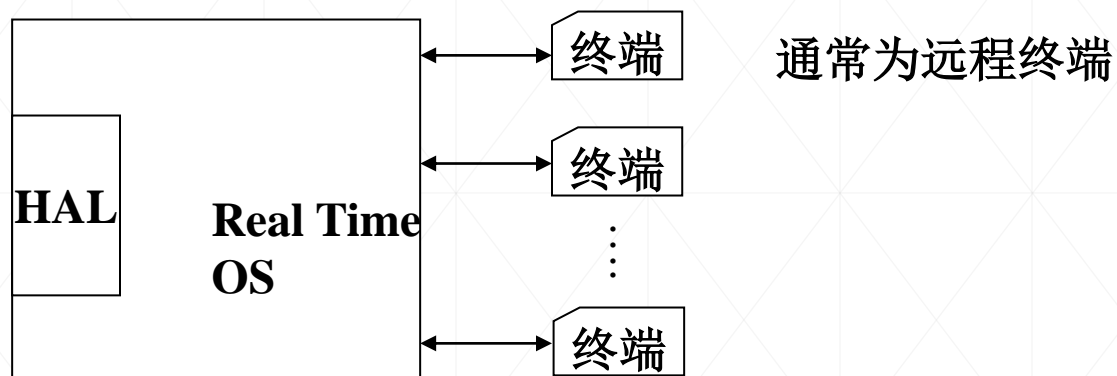
## 1.2.5 实时系统

- 所谓实时系统：是计算机**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，并控制所有实时设备和实时任务协调一致的运行。
  - 1、实时控制系统：工业控制，军事控制，医疗控制，.....
  - 2、实时信息处理系统：航班定票，联机情报检索，.....





# 实时信息处理

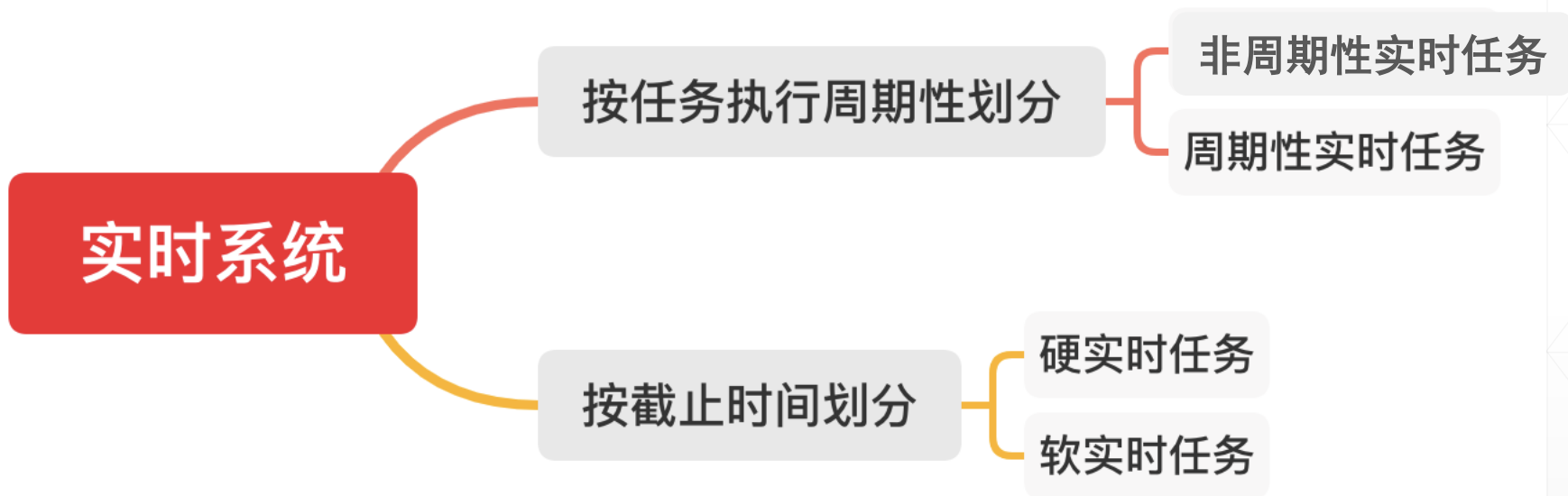


特点:

- (1) 响应及时 (prompt response)
- (2) 可靠性高 (high reliability)



## 二、实时任务的类型





### 三、实时系统的特征

- 1、**多路性**：能对多个对象进行控制。
- 2、**独立性**：独立运行，不混淆，不破坏。
- 3、**交互性**：仅限于访问系统中某些特定的专用服务程序。
- 4、**可靠性**：高可靠性，应具有多级容错防护能力。
- 5、**及时性**：不同的系统要求不一样，控制对象必须在截止时间内完成。



## 1.3 操作系统的基本特征

- 现代OS的四个基本特征：
  - 1、并发性（最重要的特征）
  - 2、共享性
  - 3、虚拟性
  - 4、异步性
- 并发是最重要的特征，其它特征都以并发为前提。

考研题目：现代操作系统的两个基本特征是（ ）  
和资源共享。

- A.多道程序设计      B.中断处理  
C.程序的并发执行    D.实现分时与实时处理





## 1.3.1 并发

### 1. 并发——并行性和并发性，并发执行的过程。

- - 并行性：是指两个或多个事件在**同一时刻**发生。
- - 并发性：是指两个或多个事件在**同一时间间隔内**发生。
- **任务共行**
  - 从宏观上看，任务共行是指系统中有多个任务同时运行
  - 从微观上看，任务共行是指单处理机系统中的**任务并发** (Task Concurrency: 即多个任务在单个处理机上交替运行) 或多处理机系统中的**任务并行** (Task Parallelism: 即多个任务在多个处理机上同时运行) 。



## 1.3.1 并发

### 2. 引入进程 **目的：为了使多个程序能并发执行。**

- 程序：静态实体
- 进程：系统中能独立运行并作为资源分配的基本单位，由一组机器指令、数据和堆栈等组成的独立运行的活动实体。

### 3. 引入线程

- 进程作为资源分配的基本单位
- 线程作为独立运行和调度的基本单位



## 1.3.2 共享

- **共享**：是指系统中的资源可供内存中多个**并发执行的**进程共同使用。

### 1、互斥共享方式：

- 把在一段时间内只允许一个进程访问的资源，称为**临界资源**。如打印机、栈、表格等
- 系统中的临界资源可以提供给多个进程使用，但**一段时间内仅允许一个进程**使用，称为**互斥共享**方式。



## 1.3.2 共享

### 2、同时访问方式:

- 从宏观上看，资源共享是指多个任务可以**同时**使用系统中的软硬件资源。
- 从微观上看，多个进程**交替互斥**地使用系统中的某个资源。例如磁盘。
- **并发和共享是操作系统的两个最基本的特征，它们又是互为存在的条件**



## 1.3.3 虚拟性

- **虚拟**：是指通过某种技术把一个物理实体变为（映射为）若干个逻辑上的对应物。

### 1.时分复用技术

- 虚拟处理机：分时实现
- 虚拟设备：SPOOLING技术（**自学**）

### 2.空分复用技术

- 虚拟磁盘技术：逻辑分区
- 虚拟存储器：虚拟存储管理实现



## 1.3.4 异步性

- 1、**执行结果不确定**，程序不可再现。
- 2、**异步性**，多道程序环境下程序（进程）以异步的方式执行，每道程序在何时执行、各自执行的顺序、完成时间都是不确定的，也是不可预知的。

进程以人们不可预知的速度向前推进。

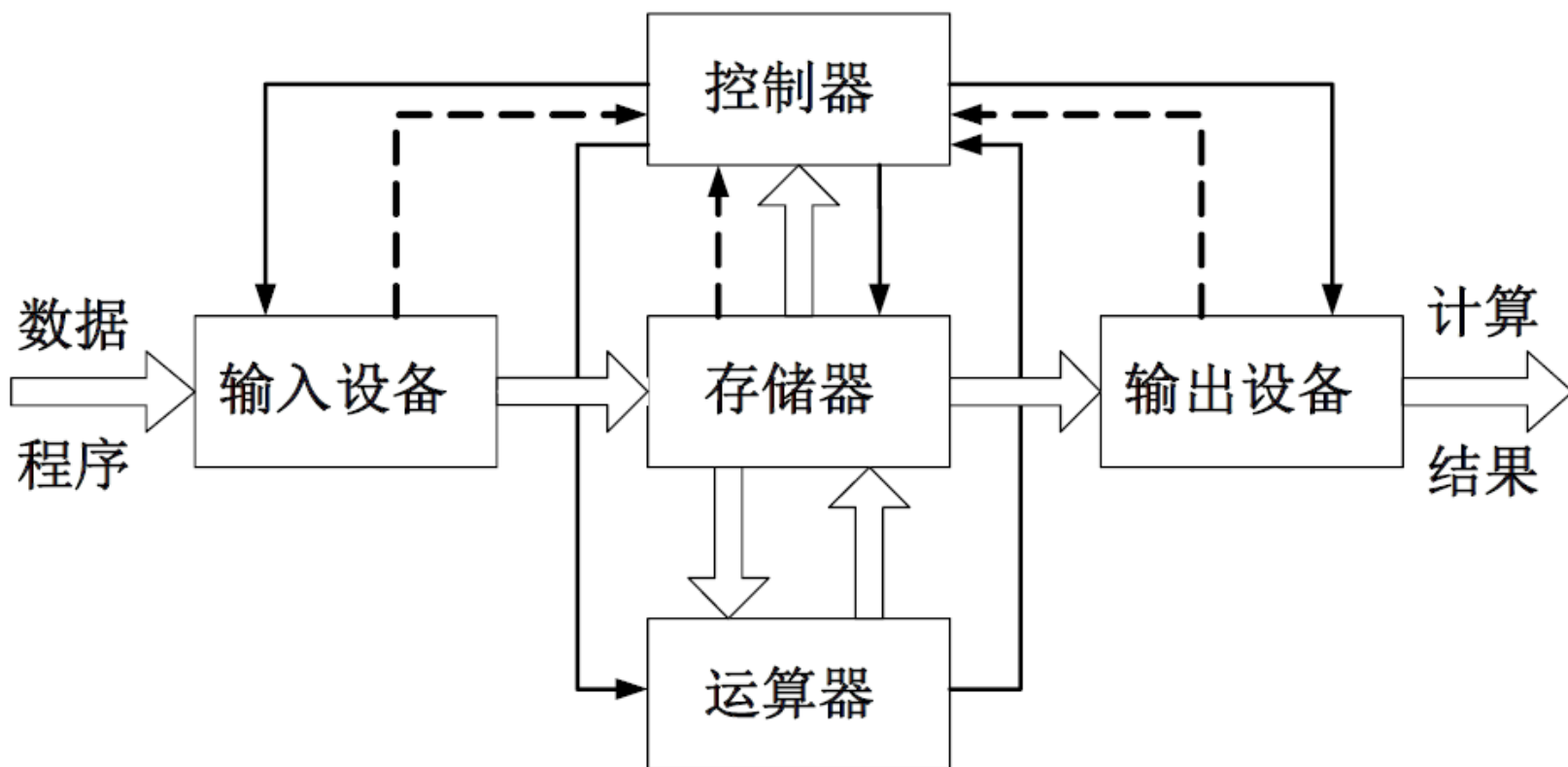


运行结果如何保证正确？





# 存储程序计算机的结构





## 1.4 操作系统的主要功能

- 操作系统的主要任务：
  - 为多道程序的运行提供良好的运行环境，以保证多道程序能有条不紊地、高效地运行，并能最大程度地提高系统中各种资源的利用率和方便用户的使用。
- 操作系统应具有五方面的功能：
  - 1、处理机管理 (CPU)
  - 2、存储器管理
  - 3、设备管理
  - 4、文件管理
  - 5、方便用户使用的用户接口。



## 1.5 OS结构设计

- 操作系统是一个大型系统软件，其结构已经历了四代的变革：
  - 第一代的OS是无结构的；
  - 第二代OS采用了模块式结构；
  - 第三代是层次式结构
  - 现代OS结构是微内核结构；



## 传统的操作系统结构

- 操作系统中增加了越来越多的功能，并且随着底层硬件更高的性能，更加通用，操作系统的大小和复杂性也随着增加。
- 为了控制该软件的复杂性，在开发OS时，先后引入了分解、模块化、抽象和隐蔽等方法。开发方法的不断发展，促进了OS结构的更新换代。



# 1.5.1 传统操作系统结构

## 1. 无结构操作系统

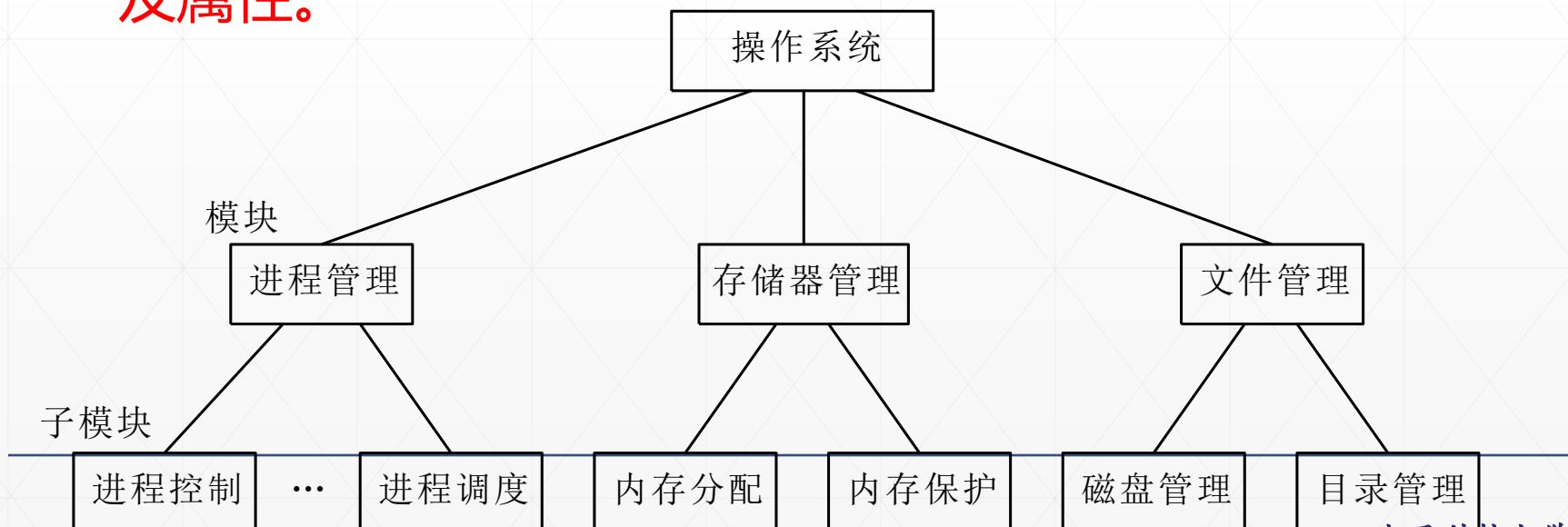
- 在早期开发操作系统时，设计者只是把他的注意力放在功能的实现和获得高的效率上，缺乏首尾一致的设计思想。
- OS是为数众多的一组过程的集合，各过程之间可以相互调用，在操作系统内部不存在任何结构，因此，有人把它称为整体系统结构。
- **缺陷：**
  - 设计出的操作系统既庞大又杂乱，缺乏清晰的程序结构。
  - 编制出的程序错误很多，给调试工作带来很多困难；增加了维护人员的负担。



## 2 模块化OS结构

### (1) 模块化结构（模块-接口法）

- 使用分块结构的系统包含若干 module（模块）；其中，每一块实现一组基本概念以及与其相关的基本属性。
- 块与块之间的相互关系：
  - 所有各块的实现均可以任意引用其它各块所提供的概念及属性。







## 2 模块化OS结构

### (2) 模块化OS的优缺点

#### 优点:

- ①提高了OS设计的正确性、可理解性和可维护性。
- ②增强了OS的可适应性。
- ③加速了OS的开发过程。

#### 缺点:

- ① 对模块的划分及对接口的规定要精确描述很困难。
- ②从功能观点来划分模块时，未能将共享资源和独占资源加以区别。



## 3 分层式OS结构

- 使用分层系统结构包含若干layer (层) ; 其中, 每一层实现一组基本概念以及与其相关的基本属性。
- 层与层之间的相互关系:
  - 所有各层的实现不依赖其以上各层所提供的概念及其属性, 只依赖其直接下层所提供的概念及属性;
  - 每一层均对其上各层隐藏其下各层的存在。



## 3 分层式OS结构

- 层次的设置时应考虑的几个因素
  - (1) **程序嵌套**。通常OS的每个功能的实现，并非是只用一个程序便能完成的，而是要经由若干个软件层才有可能完成，因此在考虑实现OS时，每个功能可能形成的程序嵌套。
  - (2) **运行频率**。将那些经常活跃的模块放在最接近硬件的层。
  - (3) **公用模块**。把供多种资源管理程序调用的公用模块，设置在最低层，以便调用。
  - (4) **用户接口**。命令接口、程序接口以及图形用户接口。这些接口应设置在OS的最高层，直接提供给用户使用。



# 操作系统层次化体系结构

分  
层  
次

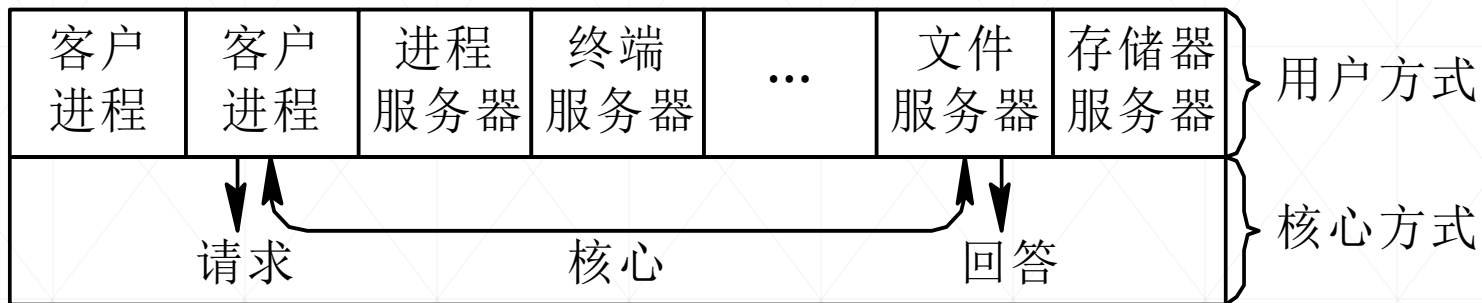


模  
块  
化

**DeltaCORE的体系结构：层次 + 模块结构**



## 1.5.2 客户/服务器模式 (Client-Server Model)



单机环境下的客户/服务器模式

### ■ 优点

- 提高了系统的可扩展性
- 增强了系统的可靠性
- 可移植性好
- 提供了对分布式系统的支持

### ■ 缺点

- 运行效率有所降低:  
消息传递开销 + 模式切换开销

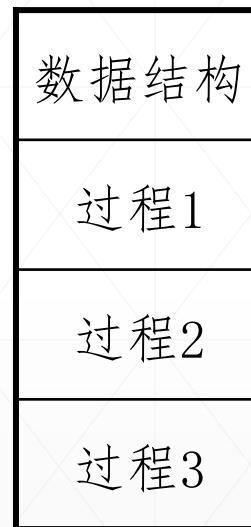


## 1.5.3 面向对象的程序设计技术 (Object-Orientated Programming)

### 1) 面向对象技术的基本概念

所谓**对象**，是指在现实世界中具有相同属性、服从相同规则的一系列事物的**抽象**，而把其中的**具体事物**称为**对象的实例**。

OS中的各类实体如进程、线程、消息、存储器等，都使用了对象这一概念，相应地，便有进程对象线程对象、存储器对象等。



一个对象的示意图





## 1.5.3 面向对象的程序设计技术 (Object-Orientated Programming)

### 2) 面向对象技术的优点

(1) 可修改性和可扩充性。由于隐蔽了表示实体的数据和操作，因而可以改变对象的表示而不会影响其它部分，从而可以方便地改变老的对象和增加新的对象

(2) 继承性。继承性是面向对象技术所具有的重要特性。继承性是指子对象可以继承父对象的属性，这样，在创建一个新的对象时，便可减少大量的时空开销

(3) 正确性和可靠性。由于对象是构成操作系统的基本单元，可以独立地对它进行测试，这样，比较易于保证其正确性和可靠性，从而比较容易保证整个系统的正确性和可靠性



## 1.5.4 微内核OS结构

- 所谓微内核技术，是指精心设计的、能实现现代OS核心功能的小型内核，它与一般的OS(程序)不同，它更小更精炼，它不仅运行在**核心态**，而且开机后常驻内存，它不会因内存紧张而被换出内存。
- 当前比较流行的、能**支持多处理机**运行的OS几乎全部都采用了微内核结构，如Mach OS, windows 2000

- 尽可能把内核功能移到用户空间
- 用户模块间的通信使用消息传递
- 好处: 灵活/安全...
- 缺点: 性能





## 1.5.5 微内核OS结构

- 微内核所提供的功能，通常都是一些最基本的功能，如进程管理、低级存储器管理、中断和陷入处理、进程间通信、低级I/O功能
- 微内核特点
  - 足够小的内核。
  - 基于客户/服务器模式
  - 应用“**机制与策略分离**”原理
  - 采用面向对象技术



## 补充：我国UNIX操作系统的研发过程

- **1978年**：我国科研人员出国学习，开始了解UNIX操作系统的功能特色、应用情况和发展趋势。
- **1981年**：科研人员陆续回国汇报，政府部门逐步确立了我国应该尽快研究应用UNIX操作系统的方针。
- **1982年**：确认因UNIX操作系统是高技术战略性产品，所以“巴黎统筹委员会”禁止向中国出口。
- **1983年**：由国家计委、国家经委、国家科委和财政部四部委联合下达了国家第六个五年计划重大科技攻关项目。



# 补充：我国UNIX操作系统的研发过程

**1984年：**

获得了UNIX V7操作系统的源代码。

**1985年：**

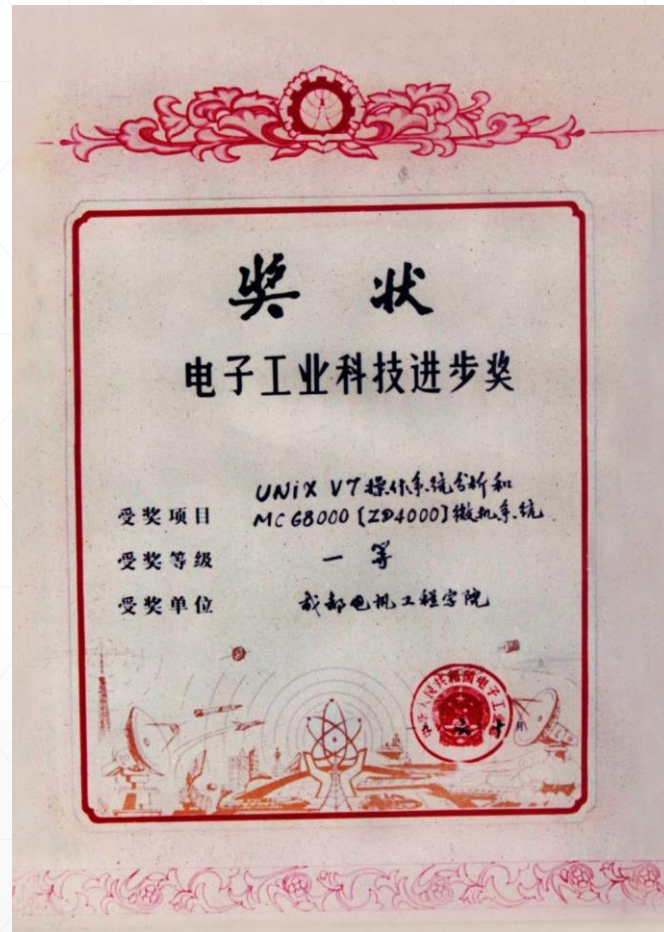
撰写出UNIX V7操作系统的内核分析报告。至此，中国已完全掌握的UNIX操作系统的原理和内部结构，打破了西方国家对中国的封锁。







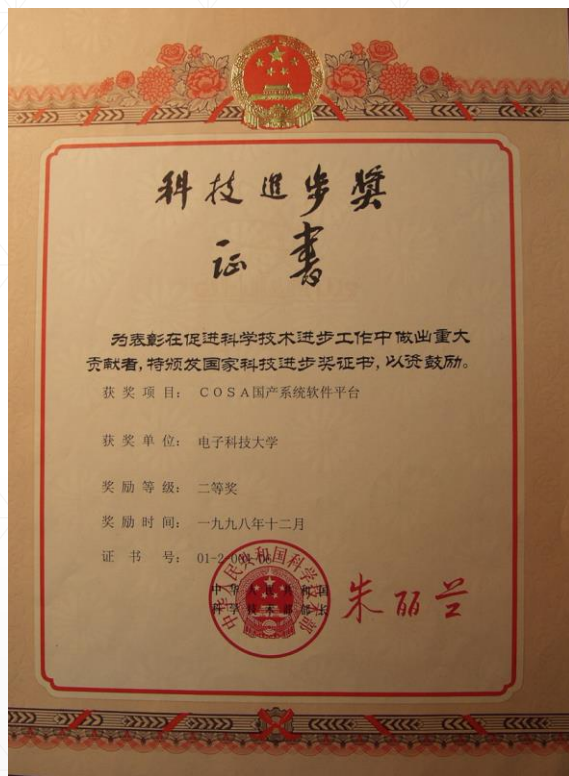
# 补充：我国UNIX操作系统的研发过程







# 补充：我国UNIX操作系统的研发过程



在这个项目中充分体现了我国科技人员在条件有限的环境下，为了国家的强盛和科技繁荣，不畏强权、艰苦奋斗、勇于拼搏、百折不挠的精神。

明白了一个道理：**只有自己掌握了核心技术，站在科技制高点上，才能摆脱对西方国家的依赖，才不会受制于人，才能够真正走自己的科技发展道路。**

## 作业

- 书 第一章 习题

1, 5, 6, 8, 10-11, 15-18

---