

教材

- 《计算机操作系统原理分析》(第三版)

主要参考书

- [1]Andrew S. Tanenbaum,Albert S. Woodhull著, Operating Systems Design and Implementation (Second Edition), 北京: 清华大学出版社, 1997.9。
- [2]汤小丹、梁红兵、哲凤屏、汤子瀛编著, 计算机操作系统(第三版), 陕西西安: 西安电子科技大学出版社, 2010.8。
- [3]张尧学、史美林、张高编著, 计算机操作系统教程(第3版), 北京: 清华大学出版社, 2006. 7。

前言

一、计算机操作系统的意义

- 1.作为专业基础课，拥有计算机专业的大量名词和术语，理解和掌握这些专业名词，是专业素质的一个体现
- 2.学习和理解操作系统管理的思想和原理，可以提高分析问题、解决问题的能力
- 3.用户通过操作系统使用计算机，学习、了解操作系统可以为工作建立扎实基础
- 4.计算机专业的许多级别的升学考试都要求操作系统。为将来进一步深造多一种选择机会

二、学习操作系统的基本方法

平时积累： 操作系统概念、原理、思想和方法，需要在课堂上认真听讲，课后复习巩固，在理解的基础上日积月累。

重视实践： 操作系统实例(特别是UNIX、Linux)的使用和编程实践。

自觉学习： 只有自觉学习才能把操作系统的思想和方法转化为自己的知识。

第一章 引论

- 一、操作系统定义

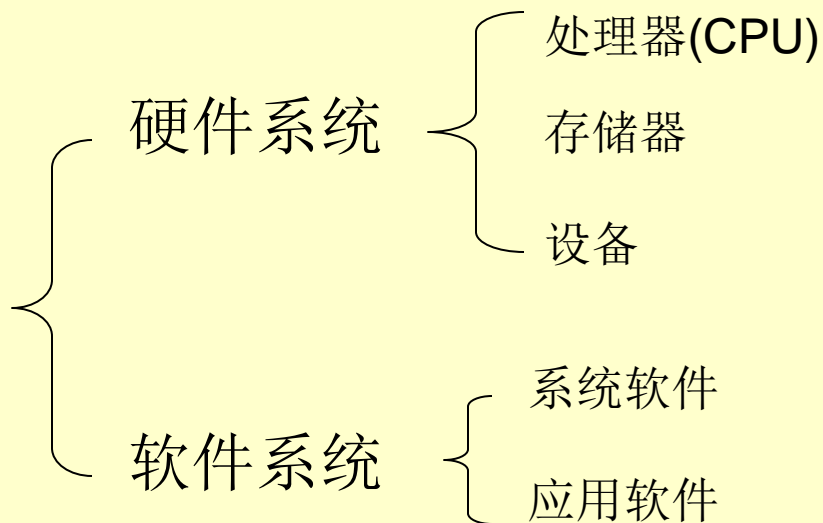
- 1.什么是操作系统（Operating Systems,OS）

计算机硬件系统上配置的第一个大型软件，称为计算机操作系统，如果该软件满足：

- 1)管理计算机系统的硬件和软件；
 - 2)控制计算机系统的工作流程；
 - 3)为其他软件 and 用户提供安全、方便的运行、操作环境；
 - 4)提高计算机系统的效率。

或者说，计算机操作系统是在研究计算机系统的工作方式和使用方式基础上，提出对计算机系统进行管理、控制的原理和方法，让计算机能够更好地为人们的学习、工作和生活服务。

2.计算机系统的组成(资源)



通过系统中各要素相互作用实现系统的整体功能。系统仅仅依靠要素还不够，因为各自独立的要素不会自动完成系统的目标，只有通过管理、控制，使之有机地结合在一起，系统才能正常地运行与活动，有效地发挥系统的功能。计算机系统就是在操作系统的管理、控制下完成系统的功能。

3.计算机系统的层次结构



图 1-1 计算机系统的层次结构

层次结构的概念

- 层
- 接口
- 单向依赖
- 隐藏性

操作系统是对硬件层的第一次扩充，同时又作为其他软件运行和用户操作的基础。起了“承下启上”的作用。

- 二、操作系统的形成

- 1.多道程序设计 with 操作系统

多道程序设计是指：在内存同时存放多道程序，这些程序可以并发执行。

多道程序的**并发执行(Concurrence)**，是指：在多道程序设计环境下，处理器在开始执行一道程序的第一条指令后，在这道程序完成之前，处理器可以开始执行另一道程序、甚至更多的其他程序。

这种工作流程的外在表现就是**多任务**，现代的计算机操作系统都采取了并发执行的工作流程。

顺序执行是指：处理器在开始执行一道程序后，只有在这道程序执行结束(程序指令运行完成，或程序执行过程出现错误而无法继续运行)，处理器才能开始执行下一道程序。

这种工作流程的外在表现就是**单任务**，早期的计算机系统是采用顺序执行的工作流程。

例1-1：假定某计算机系统需要执行两道程序A、B，程序A、B的任务描述如下：

程序A：

2ms CPU

10ms I/O

2ms CPU

程序B：

12ms CPU

5ms I/O

2ms CPU

在同样假定程序A先运行的情况，如果分别按顺序执行和并发执行的工作方式，那么，系统的工作过程怎样？

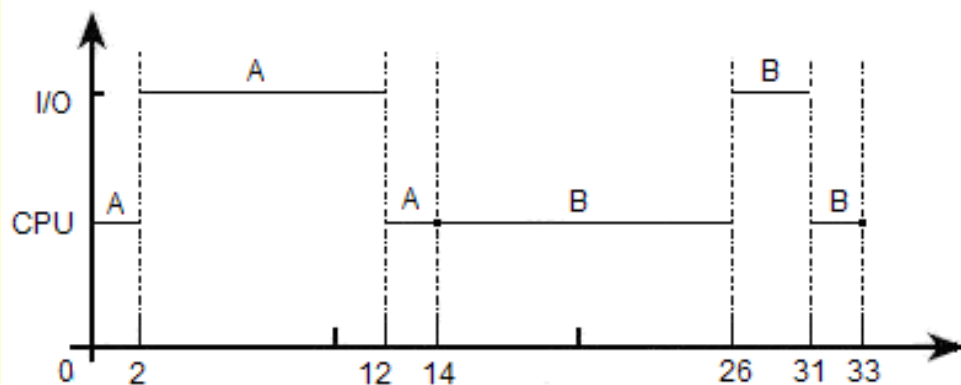


图1-2 顺序执行调度图的例子

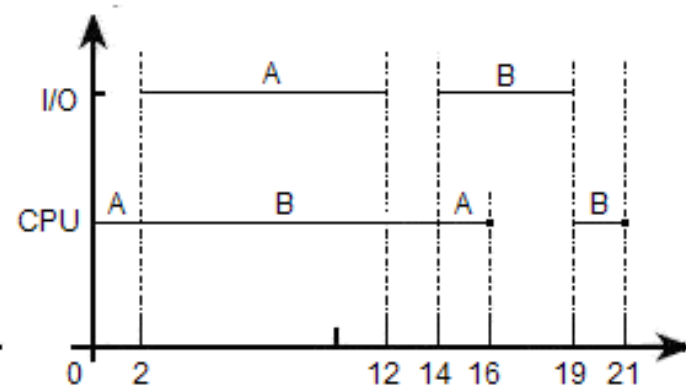


图1-3 并发执行调度图的例子

一个资源在指定时间段的利用率定义：

$$\frac{\text{指定时间段中具体工作时间的总和}}{\text{指定时间段时间长度}}$$

如图1-2所示的顺序执行方式，CPU的利用率为：

$$(2+2+12+2)/33 \approx 54.5\%$$

如图1-3所示的并发执行方式，CPU的利用率为：

$$(2+12+2+2)/21 \approx 85.7\%$$

多道程序是如何提高CPU的利用率？

在以后的学习中，还将介绍多道程序并发执行工作方式的其他优点，但是，多道程序的并发执行是一种复杂的工作方式，甚至导致一些程序运行后出现错误的结果。可是，人们正是对多道程序并发执行工作方式的复杂性和存在问题的研究及解决，形成了操作系统原理的核心内容。

2.OS的形成（20世纪60年代中期）

OS形成的条件：在硬件上具有中央处理器与设备、设备与设备并行工作的能力，相应地，在软件上，提出了多道程序设计技术，两者奠定操作系统形成的基础。

OS形成的标志：操作系统的基本类型，即批处理系统、分时系统和实时系统

3.OS的发展（从多计算机观点看）

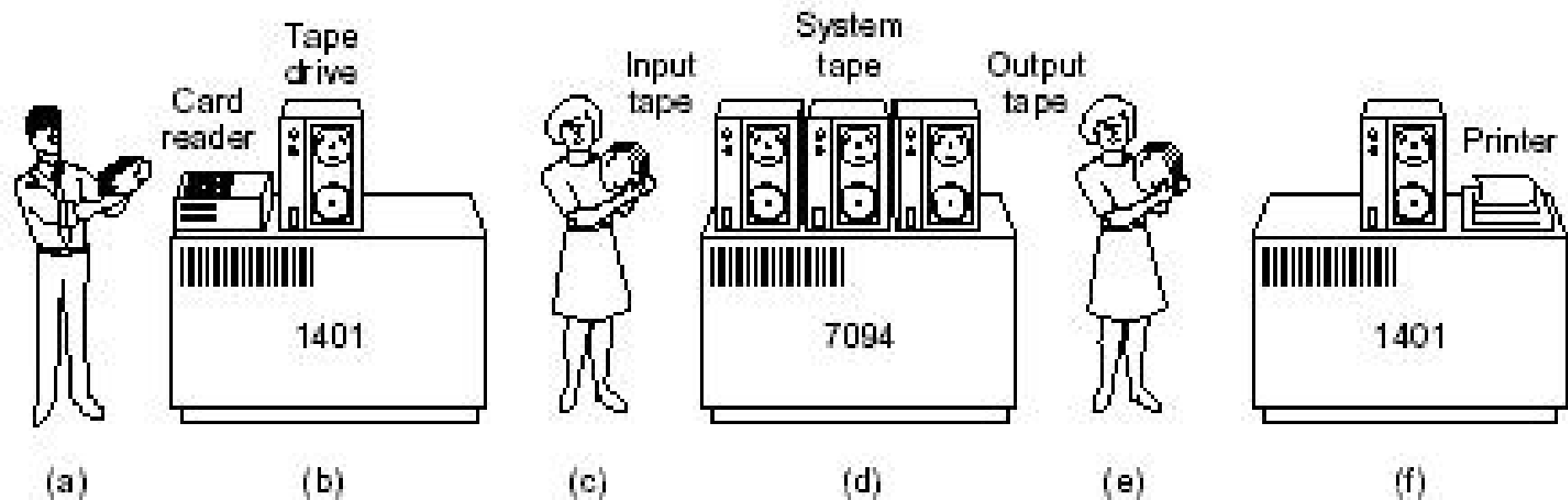
- ✓ 个人微机操作系统
- ✓ 网络操作系统
- ✓ 分布式操作系统

• 三、OS的基本类型

1.批处理系统及其特征

作业和作业步、作业流， 程序员和操作人员
操作控制语言(JCL)和作业说明书

作业经过提交、后备、执行和完成等四个阶段



单道批处理系统和多道批处理系统、脱机批处理系统和联机批处理系统

- SPOOLing技术（假脱机批处理系统）
Simultaneous Peripheral Operation On Line

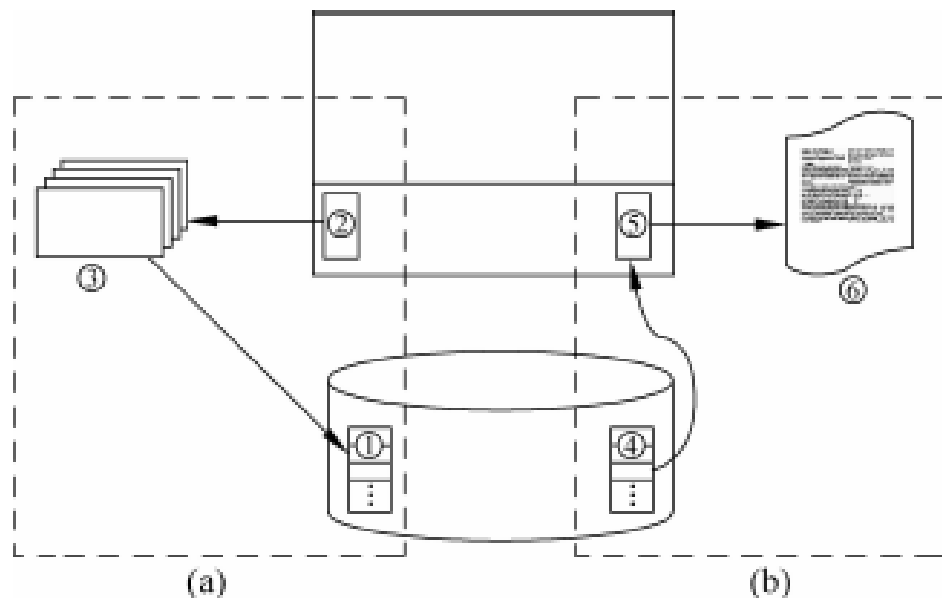


图 1-5 联机批处理系统

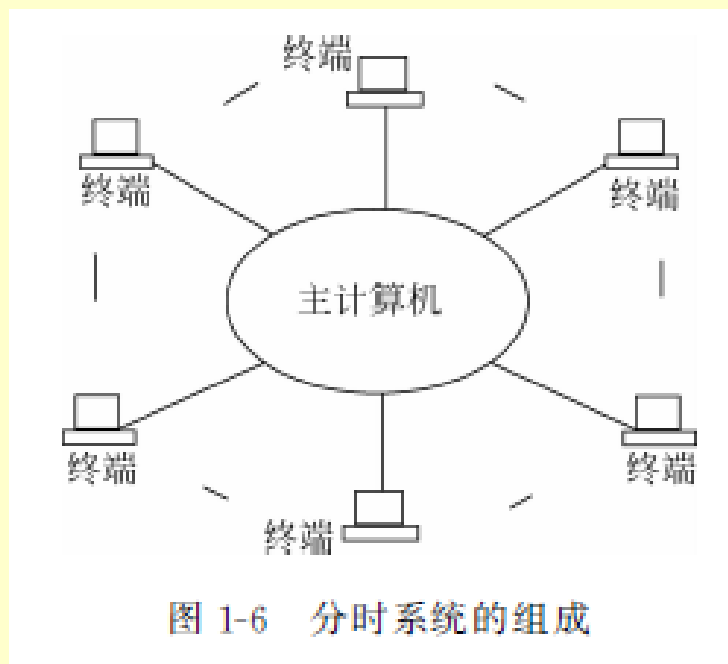
SPOOLing组成

- 输入井
- 输出井
- 预输入程序
- 缓输出程序

批处理系统的特征

- 批量处理，减少手工操作
- 自动执行，资源利用率高
- 缺少人-机交互能力

2.分时系统及其特征



时间片(Timeslice)

分时系统具备如下四个特征

- 同时性
- 独立性
- 及时性
- 交互性

3.实时系统及其特征

定期或者随时产生事件及时响应并且在严格规定的时间范围内处理完成。

实时系统分为：

- 实时过程控制系统
- 实时信息处理系统

在实时系统中，计算机对一个任务处理的正确性，不仅要求计算结果是正确的，还要求在规定的时间内得到结果

实时系统的特征

- 高及时性
- 高可靠性

• 四、OS的研究内容

1.OS理论

研究计算机系统的工作方式、用户使用方式，以便计算机系统更加方便、有效、安全地为人们的学习、生活和工作服务

2.OS软件

依据操作系统理论，对指定计算机系统实现管理的一组程序和数据集合

2.操作系统的主要功能

- 用户接口及作业管理
- 处理器管理
- 存储器管理
- 文件系统
- 设备管理

第2章 操作系统的接口

- 一、操作系统内核

- 1.处理器指令

- 特权指令和非特权指令

- 2.处理器工作模式

- 核心态和用户态

- 3.内核的主要组成

- 与硬件密切相关的操作
 - 关键数据结构
 - 基本中断处理程序
 - 使用频繁的功能模块

- 4.内核基本特点

- 常驻内存
 - 运行在核心态

- 二、操作系统的启动

- 1.固件(Firmware)及其基本功能

- 固件是硬件平台与操作系统之间的接口

- 2. BIOS系统

- BIOS组成

- POST自检程序
 - 基本启动程序
 - 基本硬件驱动程序及其中断处理程序

- BIOS磁盘分区管理-主引导记录 (MBR) 结构

表2-1 MBR结构		
偏移量	字节数	内容
0	440	主引导程序(MBR)
440	4	磁盘签名(signature)
444	2	[未定义]
446	16	第1个分区表DPT1
462	16	第2个分区表DPT2
478	16	第3个分区表DPT3
494	16	第4个分区表DPT4
510	2	结束标志符(BRID): 55H AAH(Magic Number)

2. 可扩展固件接口(EFI)

英特尔(Intel)公司在1997年为其新推出的高性能处理器，计划设计一种可扩展的，标准化的固件接口规范EFI(Extensible Firmware Interface)，用于计算机系统的启动以及提供与操作系统的接口

2005年 UEFI(Unified EFI)论坛, 2011年4月6日推出UEFI 2.3.1.

可扩展固件接口(EFI)的特点:

- 驱动程序运行环境(DXE, Driver Execution Environment)
- 磁盘管理采用GUID分区方法
- UEFI应用程序(没有操作系统 时的系统管理)

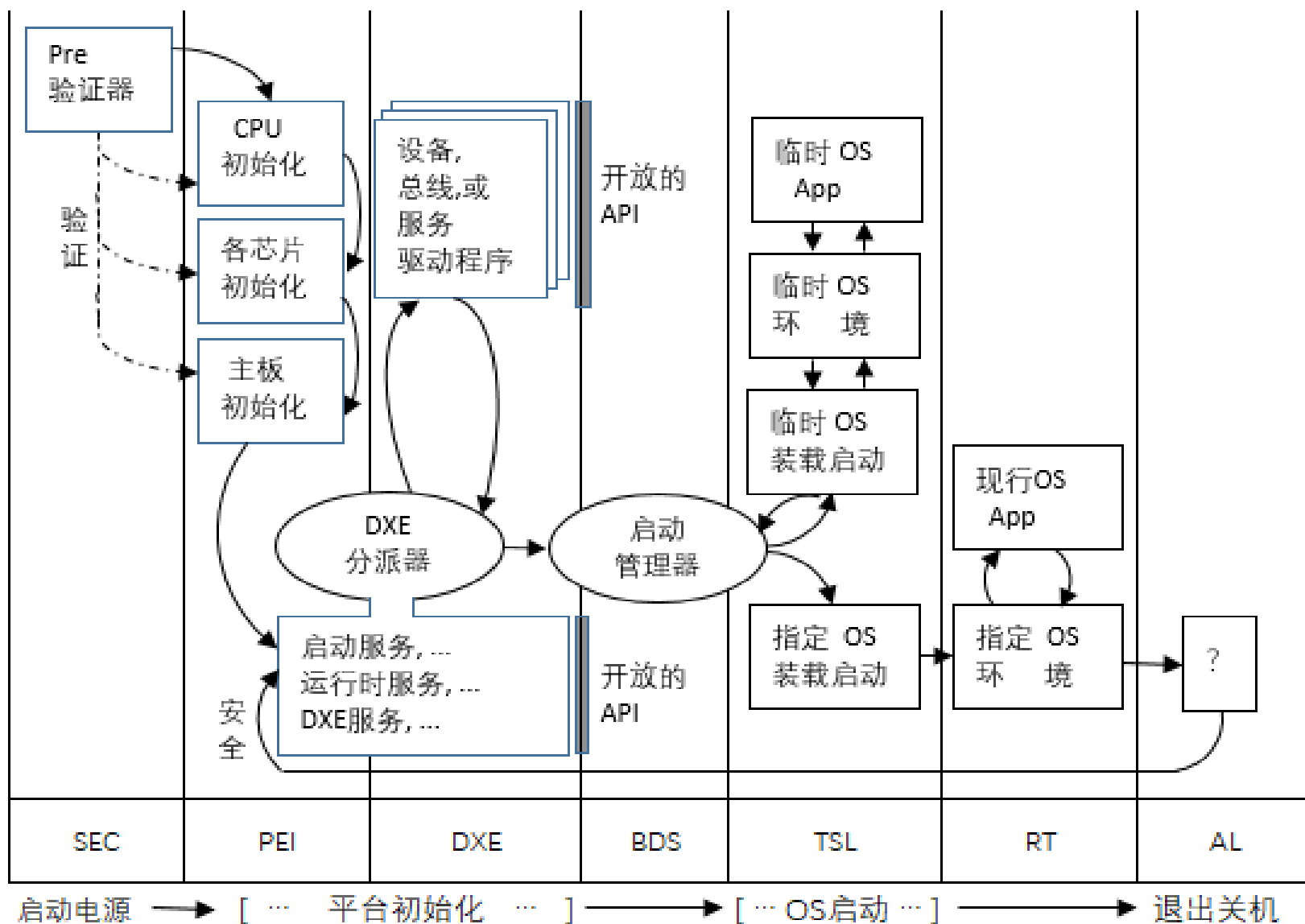


图2-2 UEFI 体系结构

表2-4 GPT表头(HDR)结构

偏移量	字节数	字段	说明
0	8	Signature	签名，表示EFI兼容分区表表头，值为”EFI PART”
8	4	Revision	GPT头版本号
12	4	HeaderSize	PDT表头的大小(字节数)， $92 < \text{HeaderSize} \leq \text{逻辑块长度}$
16	4	HeaderCRC32	GPT表头信息的CRC32校验和
20	4	Reserved	保留，置0
24	8	MyLBA	GDT表头占用的逻辑块数
32	8	AlternateLBA	GDT表备份的逻辑块号
40	8	FirstUsableLBA	可用空间的起始块的逻辑块号
48	8	LastUsableLBA	可用空间的结束块的逻辑块号
56	16	DiskGUID	磁盘的GUID
72	8	PartitionEntryLBA	GUID分区入口表(数组)的起始块的逻辑块号
80	4	NumberOfPartitions	GUID入口表中分区数
84	4	SizeOfPartitionEntry	GUID分区入口表项长度(字节数，等于 128×2 的整数倍)
88	4	PartitionEntryArrayCRC32	GUID分区入口表CRC32校验和
92	块尾	Reserved	逻辑块的剩余部分。UEFI保留，置0

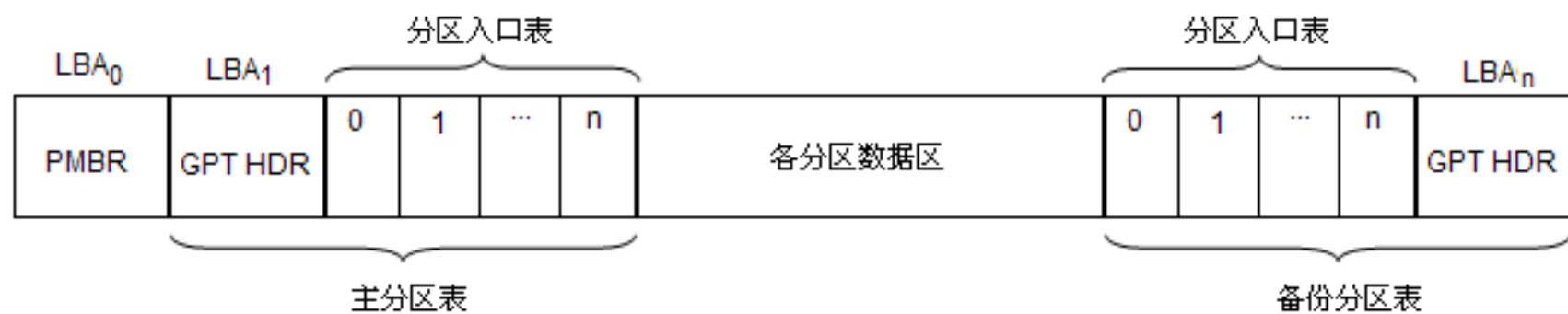


图2-3 GUID分区磁盘结构

表2-5 GPT分区入口表的表项结构

偏移量	字节数	字段	说明
0	16	PartitionTypeGUID	分区类型ID，定义分区的类型，0表示未用
16	16	UniquePartitionGUID	分区GUID
32	8	StartingLBA	分区起始块的逻辑块号
40	8	EndingLBA	分区结束块的逻辑块号
48	8	Attributes	分区属性
56	72	PartitionName	分区名，以null为结束标志的字符串，用户可读取
128	至结束	Reserved	分区表项的剩余部分。UEFI保留，置0

2. OS启动过程

在用户开机后，系统进入BIOS或UEFI的PI部分，进行硬件平台的检测和启动，之后，系统从默认启动设备或用户选择指定的启动设备上，装载操作系统的安装程序(OS Loader)。

BIOS [0000:7C00]

操作系统启动装载程序的执行过程大致如下：

- 系统配置
- 内核的装入和初始化
- 用户登录

三、操作系统的用户接口

1.命令接口

命令接口的按实现分类,分为: 外部命令和内部命令

命令接口的按使用分类,分为: 脱机命令和联机命令

作业控制语言(JCL)属于脱机命令

联机命令又分为: 字符命令、菜单命令和图标命令等三种。其中, 字符命令是基本的命令接口

2.程序接口

什么是系统调用

- 1)一组操作系统设计人员事先编写的子程序, 这些子程序作为内核的一部分;
- 2)程序员使用这组子程序的方法。

访管指令及其功能

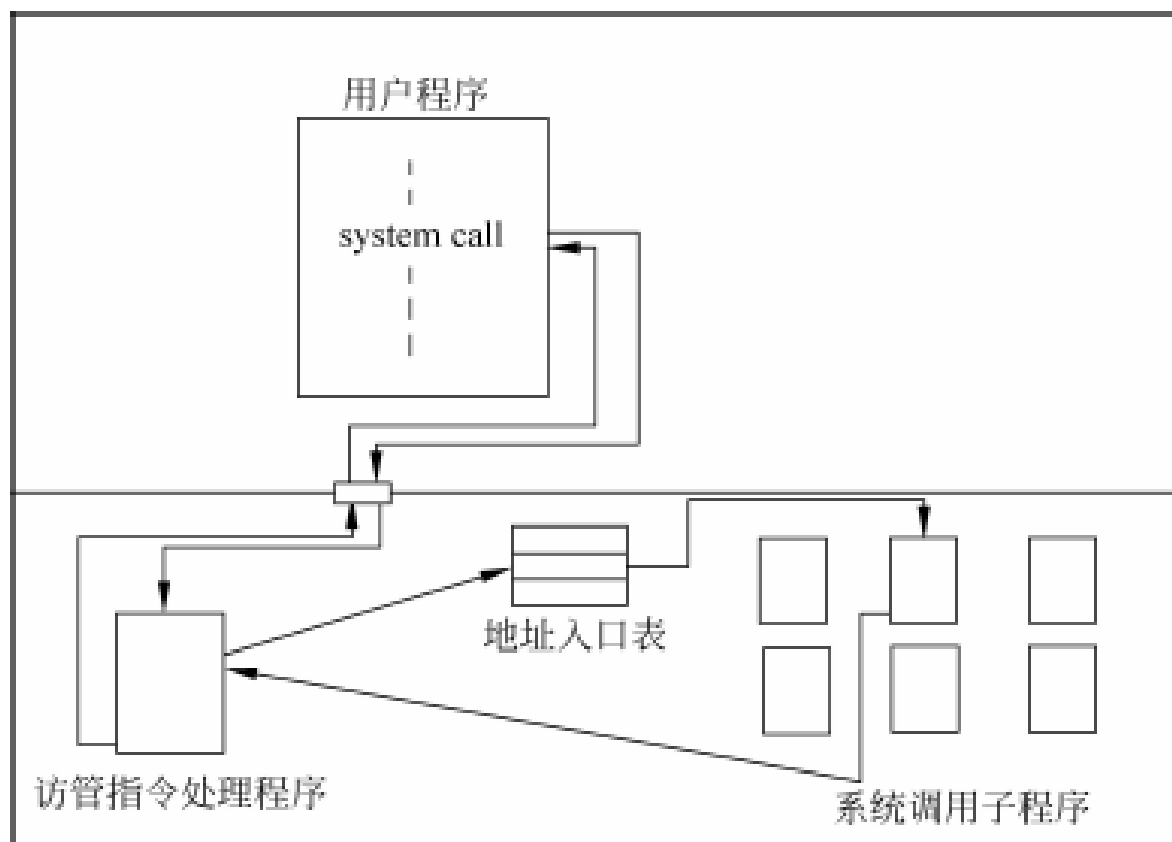


图 2-6 系统调用实现过程

对比项目	系统调用	用户子程序
运行环境	核心态	用户态
中断	访管中断	无
与主程序关系	与主程序分开、独立	同一进程地址空间
共享	不同用户可以共享	同一进程内部调用