
为什么要学习系统编程

宿船长

操作系统的概念/定义

举个栗子，

一台电脑的诞生过程：

First，厂家组装一台裸机；

Second，安装操作系统；

Third，用户安装应用程序（eg. 微信）；

Forth，用户使用微信聊天。

应用程序：微信、QQ等

eg.windows10

裸机：包括CPU、内存、
硬盘、主板等



计算机系统层次结构

(ps. 与计算机组成原理中的分层不同)

操作系统的概念/定义

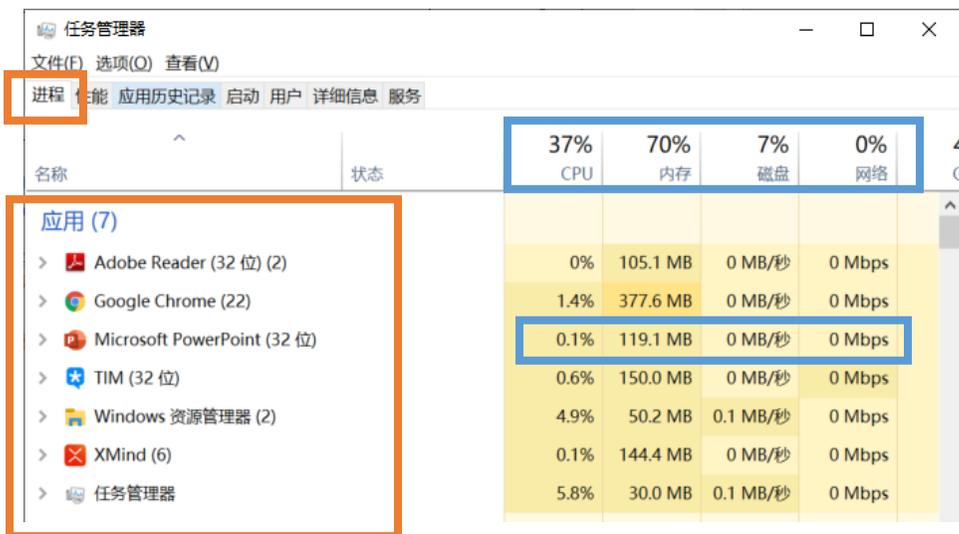
操作系统（Operating System, OS）是指控制和管理整个计算机系统的硬件和软件资源，并合理地组织调度计算机的工作和资源的分配；以提供给用户和其他软件方便的接口和环境；它是计算机系统中最基本的系统软件。

①是系统资源的管理者

③是最接近硬件的一层软件

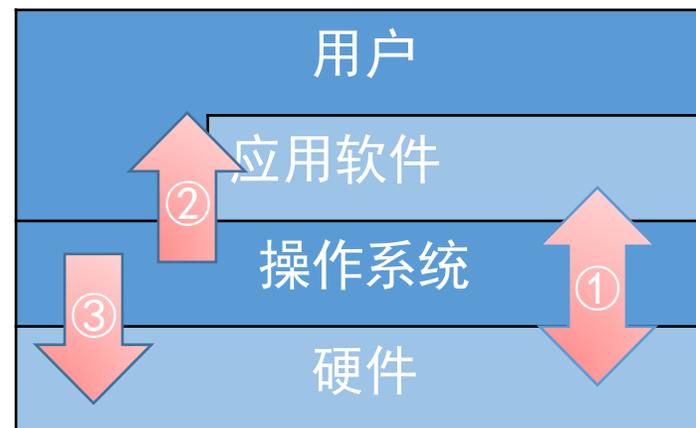
②向上层提供方便易用的服务

直观的例子：打开Windows操作系统的“任务管理器”。



对软件的管理

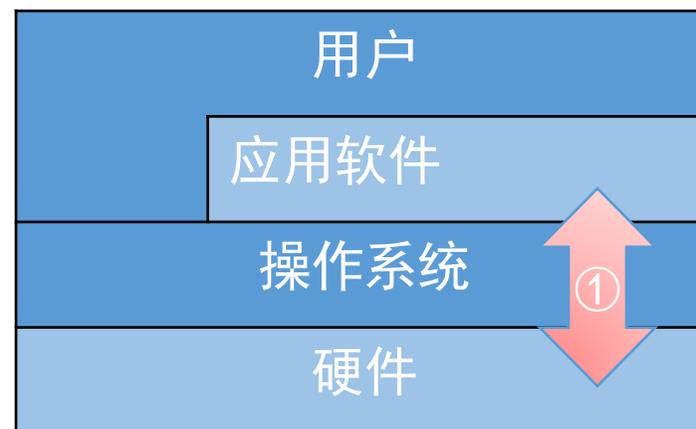
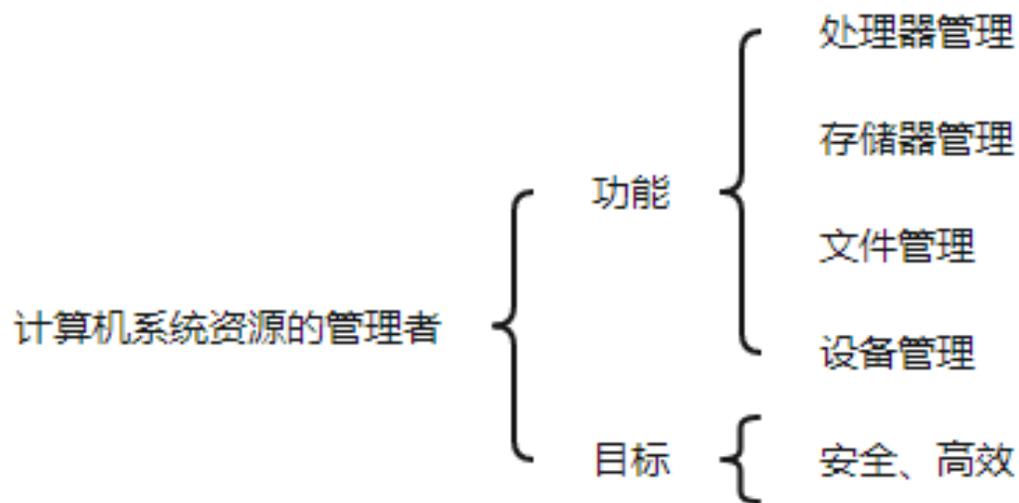
对硬件的管理



操作系统的目标和功能——作为系统资源的管理者

我们把用户比作雇主，操作系统比作工人（用来操作机器），计算机是机器（处理机+存储器+设备+文件等）。

工人控制和协调各个部件的工作 ↔ 操作系统对资源（硬件+软件）的管理



操作系统的目标和功能——向上层提供方便简易的服务

我们把用户比作雇主，操作系统比作工人（用来操作机器），计算机是机器（处理机+存储器+设备+文件等）。

工人通过包装为雇主提供服务 ↔ 操作系统为用户提供接口

“封装”思想：操作系统把一些丑陋的硬件功能封装成简单易用的服务，使用户能更方便地使用计算机，用户无需关心底层硬件的原理，只需要对操作系统发出命令即可。

用户与计算机系统之间的接口 {
命令接口
程序接口
GUI



操作系统的目标和功能——向上层提供方便简易的服务

命令接口

- 联机命令接口，又称**交互式**命令接口，适用于分时或实时系统接口。
- 特点：用户说一句，系统跟着做一句。
- 脱机命令接口，又称**批处理**命令接口，适用于批处理系统。
- 特点：用户说一堆，系统跟着做一堆。

```
~ 余杭 🐱 echo "Hello HaiZeiX 🍷"
Hello HaiZeiX 🍷
~ 余杭 🐱
```

eg.Linux系统命令解释器

```
1 #!/usr/bin/env bash
2
3 # This script is executed by GitHub Actions when a PR is merged (i.e. in the `deploy` step).
4 set -ex
5
6 function initialize {
7   if [ -z "$TLDRHOME" ]; then
8     export TLDRHOME=${GITHUB_WORKSPACE:-$(pwd)}
9   fi
10
11   export TLDR_ARCHIVE="tldr.zip"
12   export SITE_HOME="$HOME/site"
13   export SITE_REPO_SLUG="tldr-pages/tldr-pages.github.io"
14
15   # Configure git.
16   git config --global user.email "actions@github.com"
17   git config --global user.name "GitHub Actions"
18   git config --global push.default simple
19   git config --global diff.zip.textconv "unzip -c -a"
20
21   # Decrypt and add deploy key.
22   eval "$(ssh-agent -s)"
23   echo "${DEPLOY_KEY}"> id_ed25519
24   chmod 600 id_ed25519
}
~/tldr/tldr/scripts/deploy.sh [FORMAT=unix] [TYPE=SH] [POS=21,1][45%] 17/08/21 - 08:17
```

使用Linux系统的find命令，搜索用户家目录中的*.sh文件，用vim任意打开一个

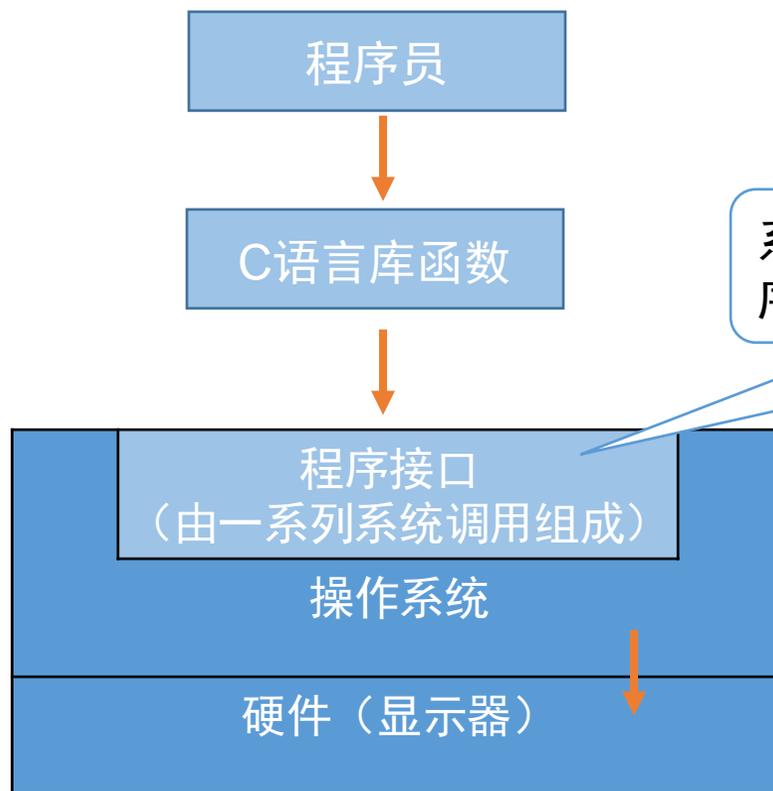
```
find . -name "*.sh"
```

操作系统的目标和功能——作为系统资源的管理者

程序接口

- 程序接口由一组**系统调用**（也称广义指令）组成。用户通过在程序中使用这些系统调用来请求操作系统为其提供服务，如使用各种外部设备、申请分配和回收内存及其他各种要求。

如：写C语言“Hello world”程序时，在printf函数的底层就使用到了操作系统提供的显式相关的“系统调用”。



系统调用类似于函数调用，是应用程序请求操作系统服务的唯一方式

操作系统的目标和功能——向上层提供方便简易的服务

GUI: 图形化用户接口 (Graphical User Interface)

用户可以使用形象的图形界面进行操作，而不再需要记忆复杂的命令、参数。



操作系统的目标和功能——作为最接近硬件的层次

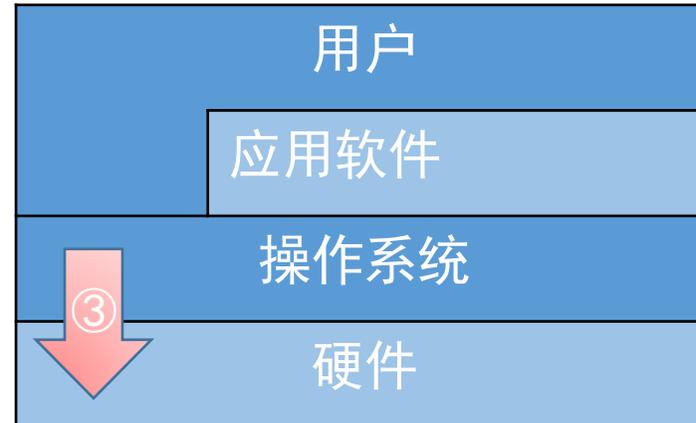
我们把用户比作雇主，操作系统比作工人（用来操作机器），计算机是机器（处理机+存储器+设备+文件等）。

有了工人，机器能发挥更大的作用 ↔ 操作系统用作扩充机器

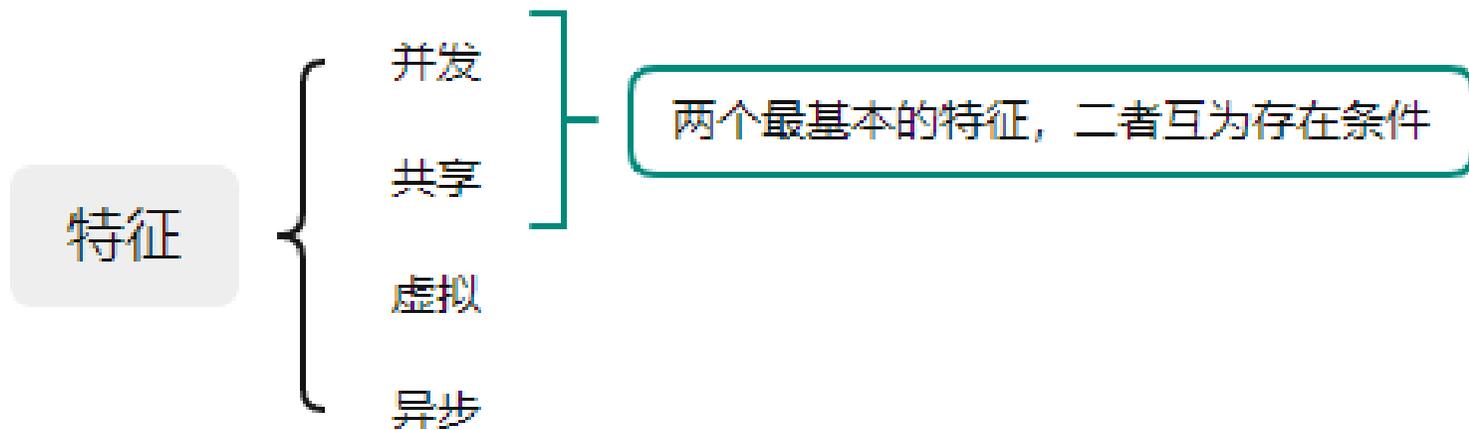
需要实现对硬件机器的拓展

没有任何软件支持的计算机成为裸机。在裸机上安装的操作
系统，可以提供资源管理功能和方便用户的服务功能，将裸
机改造成功能更强、使用更方便的机器

通常把覆盖了软件的机器成为扩充机器，又称之为虚拟机



操作系统的特征



操作系统的特征——并发

并发：指两个或多个事件在同一时间间隔内发生。这些事件**宏观上是同时发生的**，但**微观上是交替发生的**。

同一时间间隔

易混淆概念：

并行：指两个或多个事件在同一时刻同时发生。

同一时刻

操作系统的特征——并发

并发：指两个或多个事件在同一时间间隔内发生。这些事件**宏观上是同时发生的**，但**微观上是交替发生的**。

操作系统的并发性指计算机系统中“同时”运行着多个程序，这些程序宏观上看是同时运行着的，而微观上看是交替运行的。

操作系统就是伴随着“多道程序技术”而出现的。因此，**操作系统和程序并发是一起诞生的**。

注意（**重要考点**）：

单核CPU同一时刻只能执行**一个程序**，各个程序只能**并发地**执行
多核CPU同一时刻可以同时执行**多个程序**，多个程序可以**并行地**执行

比如Intel的第八代i3处理器就是**4核CPU**，意味着可以**并行地执行4个程序**。

即使是对于4核CPU来说，只要有4个以上的程序需要“同时”运行，那么并发性依然是必不可少的，因此**并发性是操作系统一个最基本的特性**



操作系统的特征——共享

共享即资源共享，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。

两种资源共享方式

互斥共享方式

系统中的某些资源，虽然可以提供给多个进程使用，但一个时间段内只允许一个进程访问该资源

同时共享方式

系统中的某些资源，允许一个时间段内由多个进程“同时”对它们进行访问

所谓的“同时”往往是宏观上的，而在微观上，这些进程可能是交替地对该资源进行访问的（即分时共享）

生活实例：

互斥共享方式：使用钉钉和微信视频。同一时间段内摄像头只能分配给其中一个进程。

同时共享方式：使用钉钉发送文件A，同时使用微信发送文件B。宏观上看，两边都在同时读取并发送文件，说明两个进程都在访问硬盘资源，从中读取数据。微观上看，两个进程是交替着访问硬盘的。

操作系统的特征——并发和共享的关系

并行性指计算机系统中同时存在着多个运行着的程序。

共享性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。

通过上述例子来看并发与共享的关系：
使用QQ发送文件A，同时使用微信发送文件B。

1. 两个进程正在并发执行（**并行性**）
2. 需要共享地访问硬盘资源（**共享性**）

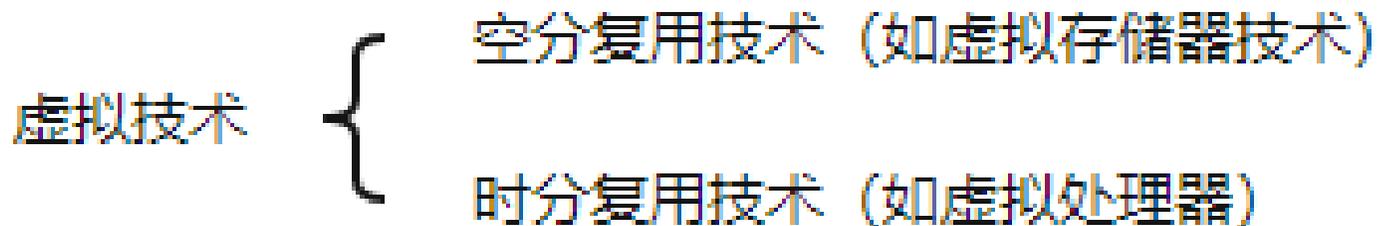
如果失去并行性，则系统中只有一个程序正在运行，则共享性失去存在的意义

如果失去共享性，则QQ和微信不能同时访问硬盘资源，就无法实现同时发送文件，也就无法并行

并行性 ↔ **共享性**
互为存在条件

操作系统的特征——虚拟

虚拟是指把一个物理上的实体变为若干个逻辑上的对应物。物理实体（前者）是实际存在的，而逻辑上对应物（后者）是用户感受到的。



操作系统的特征——虚拟

虚拟是指把一个物理上的实体变为若干个逻辑上的对应物。物理实体（前者）是实际存在的，而逻辑上对应物（后者）是用户感受到的。

空分复用技术（虚拟存储器）

GTA5需要4GB的运行内存
雷需要256MB的内存，网
存……

我的电脑：4GB内存

问题：这些程序同时运行
么为什么它们还可以在我

答：这是虚拟存储器技术。实际只有4GB的内存，在
用户看来似乎远远大于4GB

显然，如果失去了并发性，则
一个时间段内系统中只需运行
一道程序，那么就失去了实现
虚拟性的意义了。因此，**没有
并发性，就谈不上虚拟性。**

时分复用技术（虚拟处理器）

计算机中，用户打开了以下软件。



每个程序需要被分配CPU才能正常执行，
该CPU的电脑中能同时运行这么多个程

序呢？

答：这是虚拟处理器技术。实际上只有一个单核CPU，
在用户看来似乎有6个CPU在同时为自己服务

操作系统的特征——异步

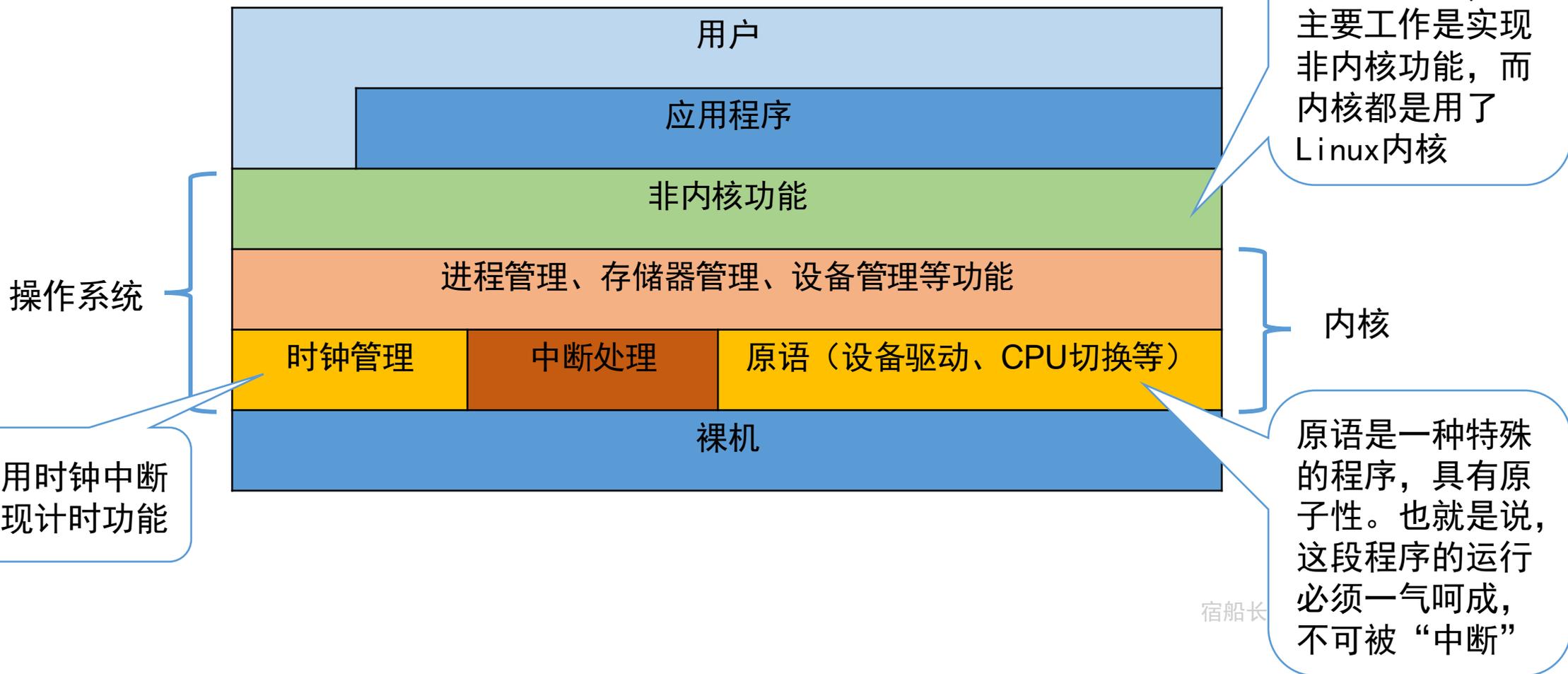
异步是指，在多道程序环境下，允许多个程序并发执行，但由于资源有限，进程的执行不是一贯到底的，而是走走停停，以不可预知的速度向前推进，这就是进程的异步性。

由于并发运行的程序会争抢着使用系统资源，而系统中的资源有限，因此进程的执行不是一贯到底的，而是走走停停的，以不可预知的速度向前推进

如果失去了并发性，即系统只能串行地运行各个程序，那么每个程序的执行会一贯到底。**只有系统拥有并发性，才有可能导致异步性。**

操作系统的体系结构

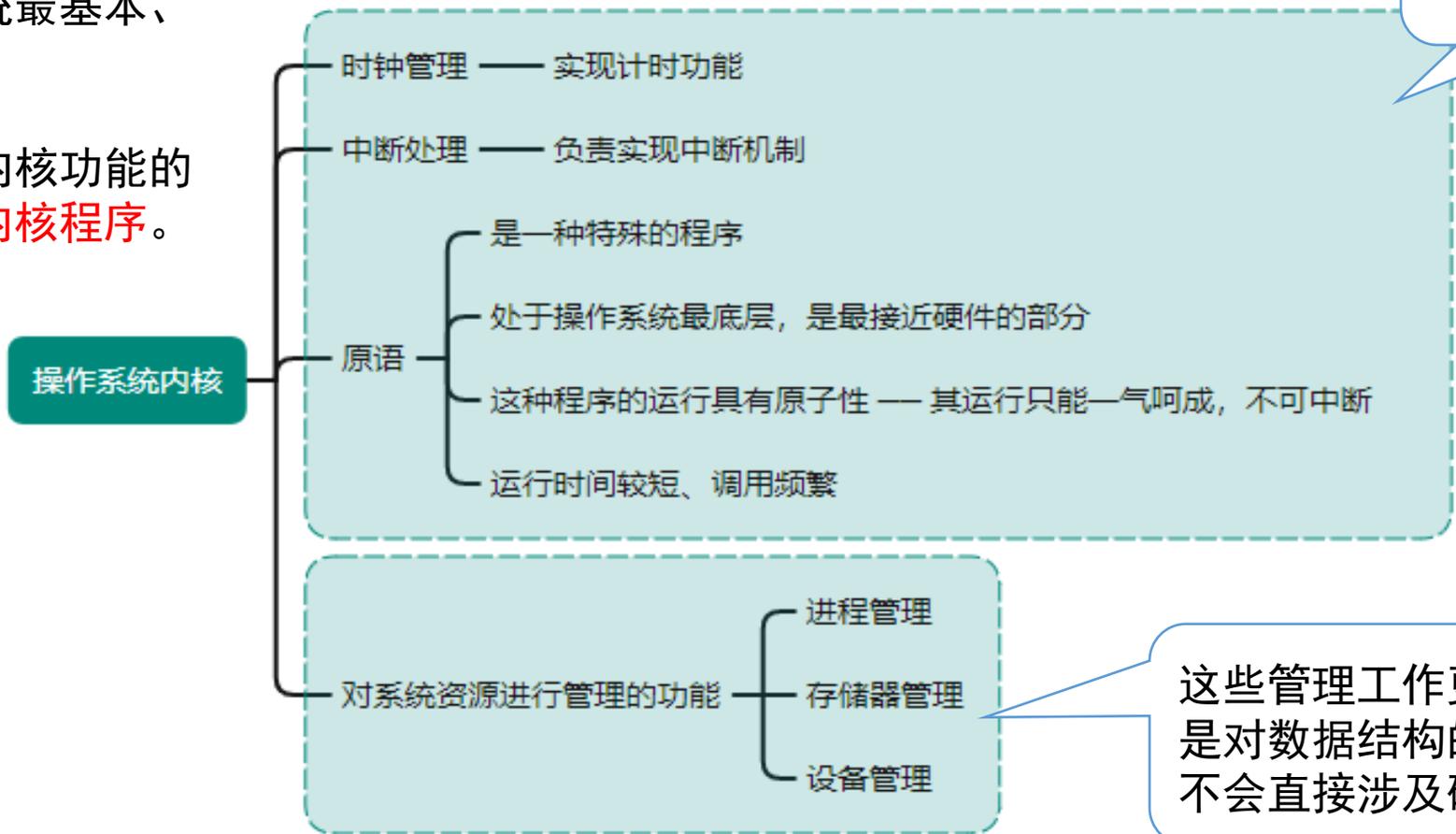
操作系统的内核



操作系统的体系结构

内核是操作系统最基本、最核心的部分。

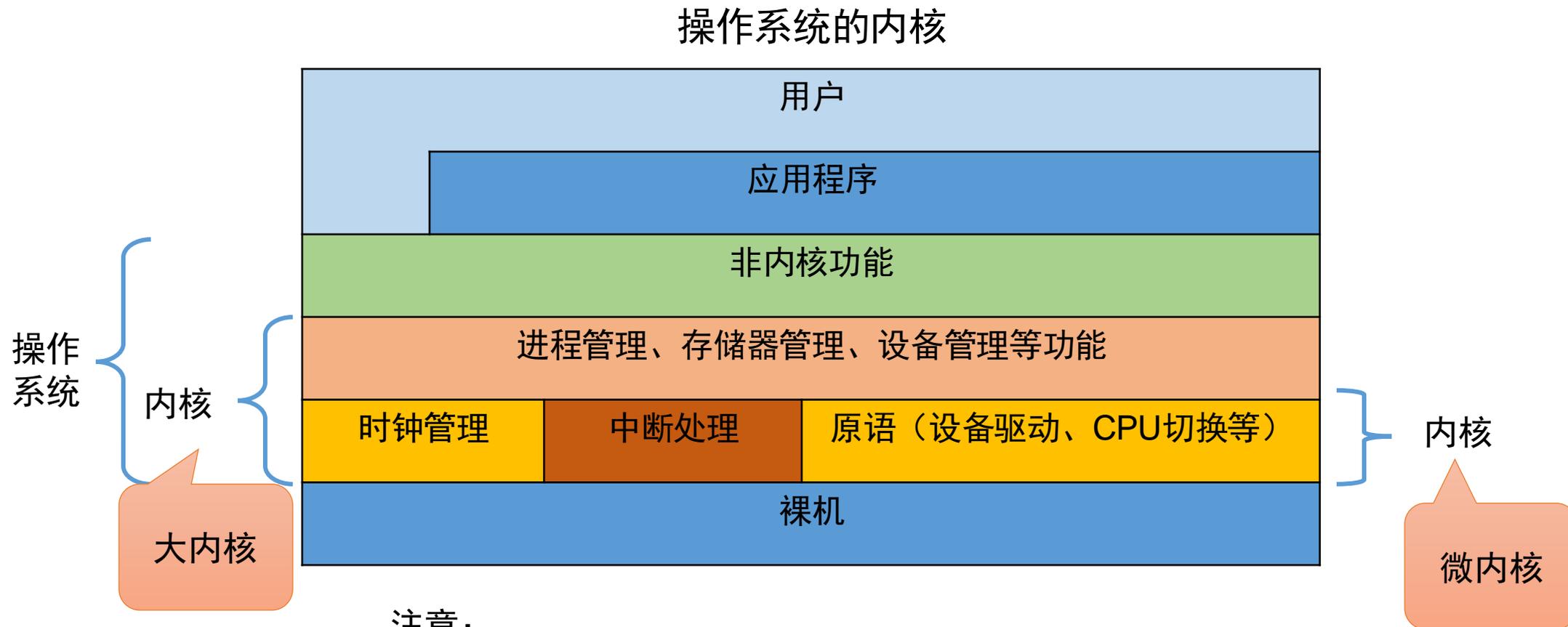
实现操作系统内核功能的那些程序就是内核程序。



与硬件关联较紧密的模块

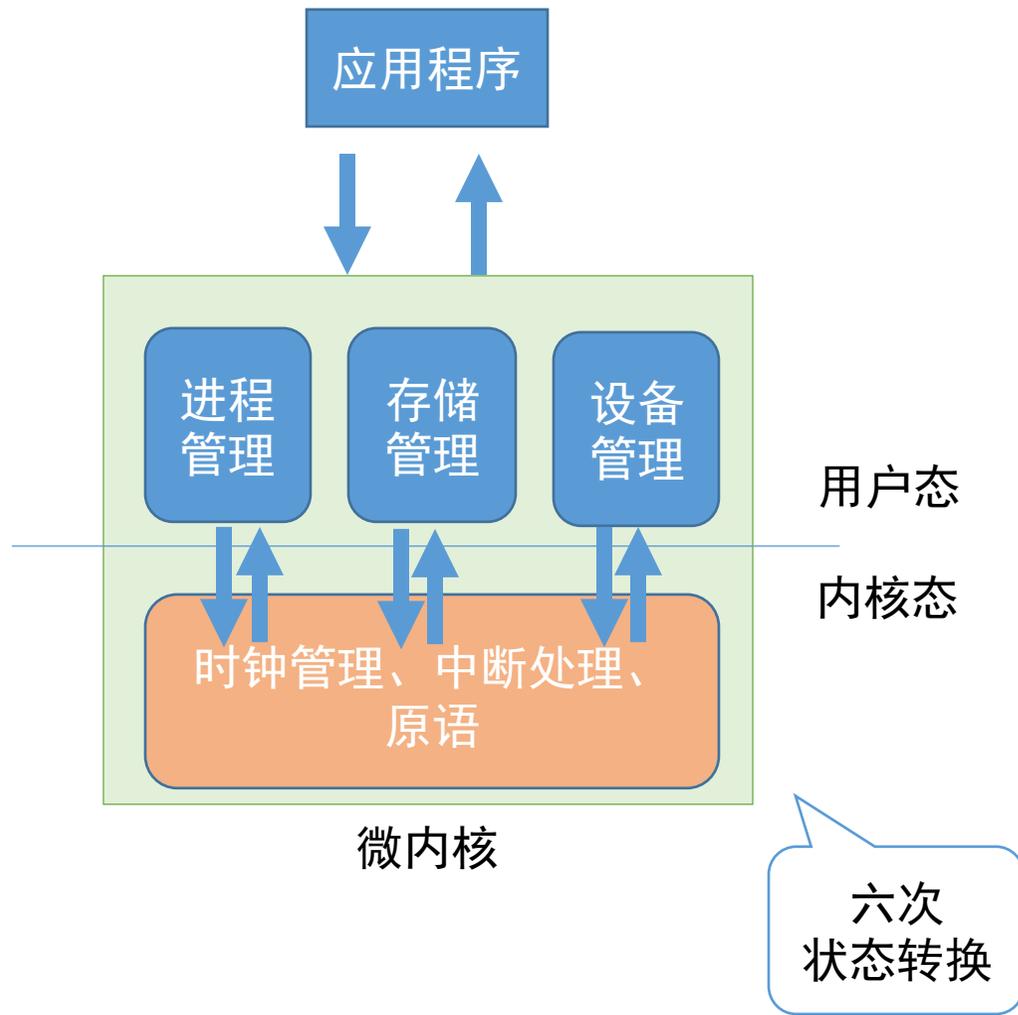
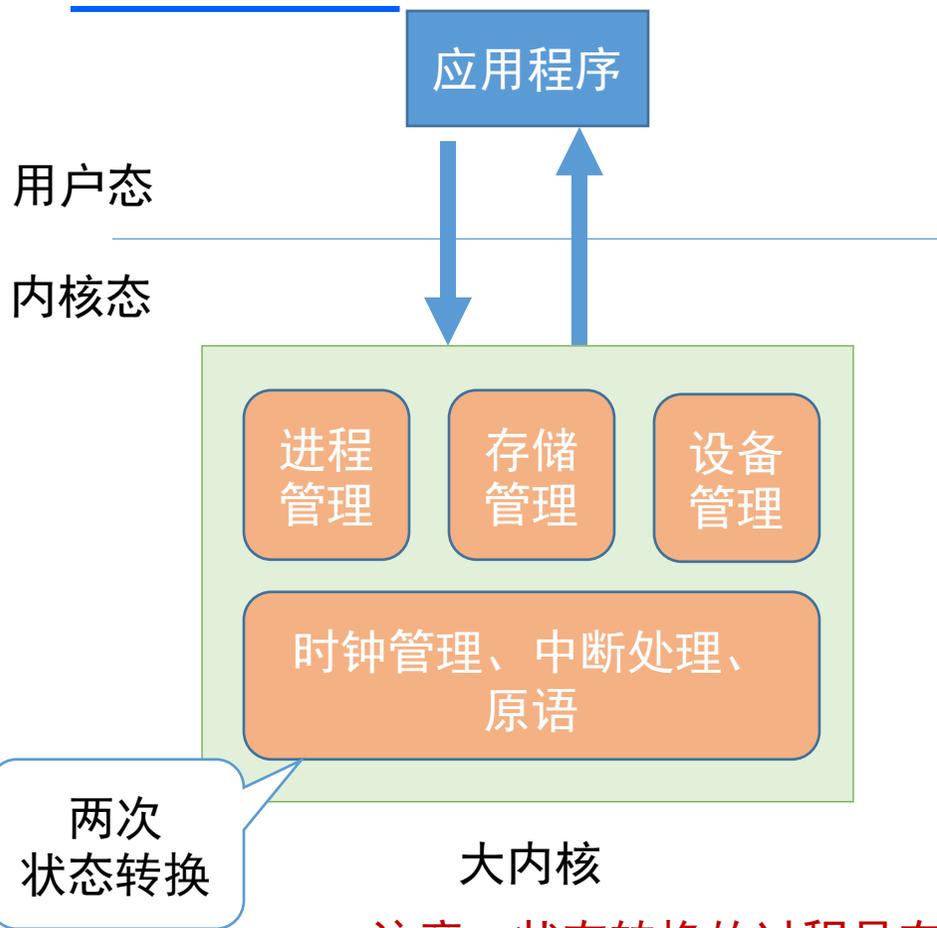
这些管理工作更多的是对数据结构的操作, 不会直接涉及硬件

操作系统的体系结构



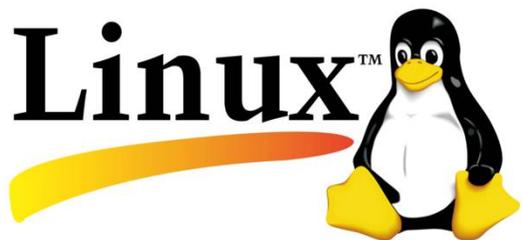
注意：
操作系统内核需要运行在内核态
操作系统的非内核功能运行在用户态

操作系统的体系结构

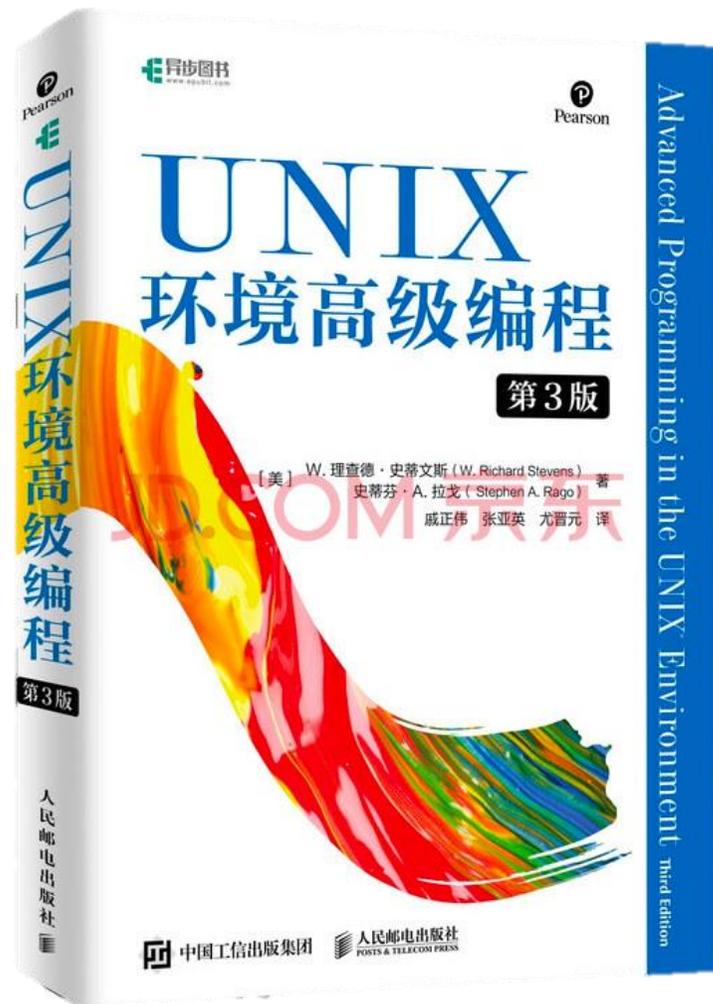
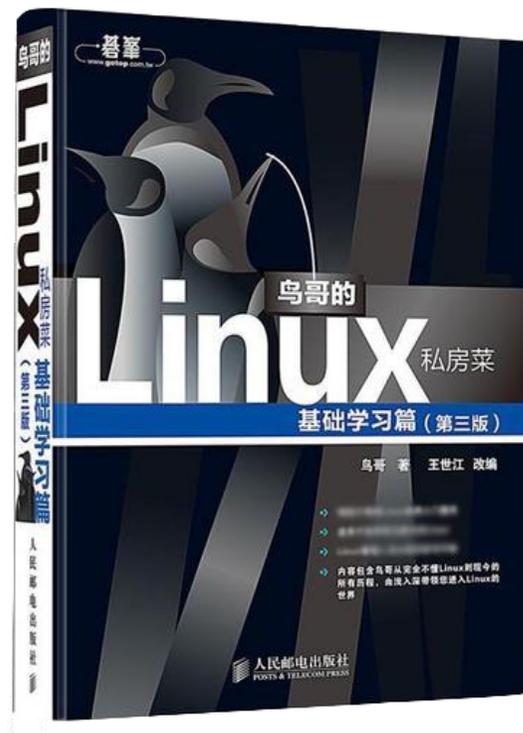
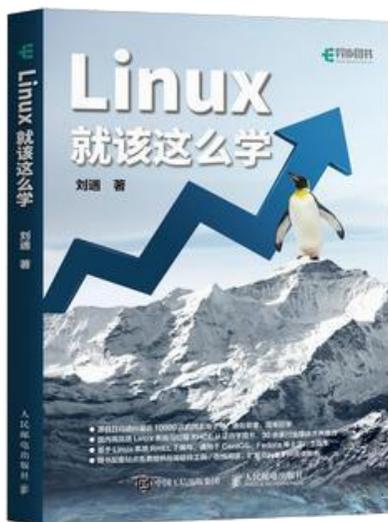


注意：状态转换的过程是有成本的，要消耗不少时间，频繁地切换CPU状态会降低系统性能

常见的操作系统



书籍推荐



Linux学习之路

从入门使用到理解操作系统内核机制
成为具备系统思想的开发者

阶段一

简单使用Linux

- 抛弃旧的思维习惯，理解Linux
- 熟练掌握Linux基本命令

阶段二

使用Linux进行编程

- 掌握在Linux下进行程序设计的基本方法
- 熟练掌握Linux系统提供的各种系统调用

阶段三

了解Linux内核机制

- 了解操作系统内核的运行机制
- 能够对操作系统底层的实现有所了解
- 对系统功能的底层原理，特别是文件、进程、线程、内存、设备相关的底层知识有较深的理解
- 在程序设计时，具备根据操作系统特点来优化程序的能力

阶段四

阅读内核，定制开发

- 具备内核代码阅读能力
- 能够针对具体实现，对内核代码进行重点突破
- 在内核代码阅读能力之上，具备定制化开发的能力

学习本课程的关键



操作系统的学习主要在于明确操作系统提供给我们的服务与功能，并将它们**用**起来

拜拜

