# 操作系统复习

第二章： 操作系统结构

第二章总纲：

* Operating System Services
* User Operating System Interface
* System Calls
* Types of System Calls
* System Programs
* Operating System Design and Implementation
* Operating System Structure
* Operating System Debugging
* Operating System Generation
* System Boot

**#操作系统服务**

操作系统提供一个环境来执行程序和对用户的服务。

**Helpful to user:**

User interface(UI) 用户界面

Program execution 程序执行

IO operations IO操作：用户不能直接控制IO设备，系统需要提供进行IO操作的方法

File -system manipulation 文件系统操作：读写，创建，删除。

Communications 通信

Error detection错误检测

**Efficient operation**

Resource allocation 资源分配

Accounting 统计：记录哪些用户需要哪些资源和使用了多少和什么类型的资源。

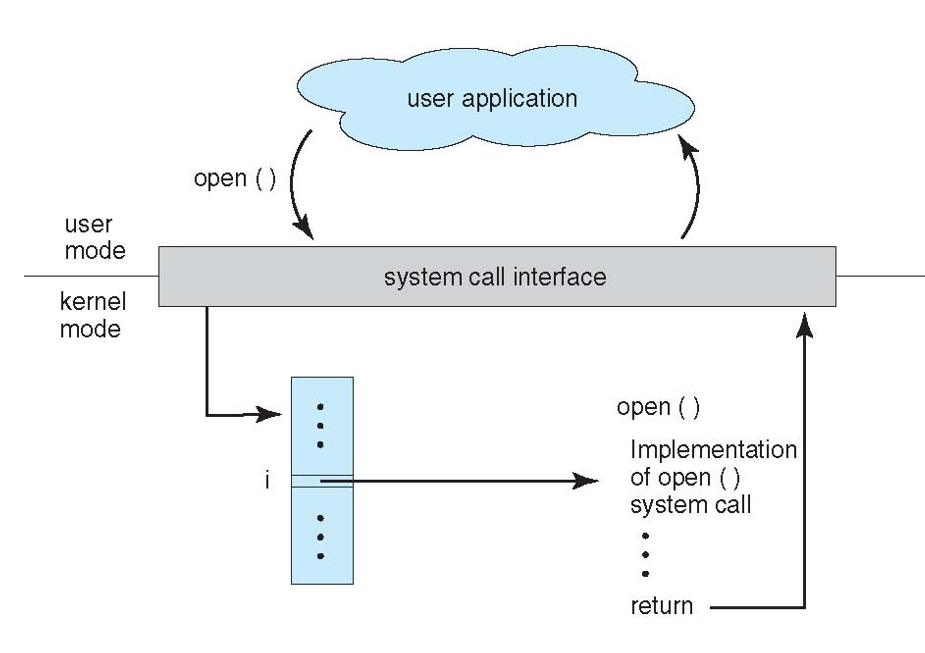
Protection and security

Tips：什么是shell: 在计算机科学中，Shell俗称壳（用来区别于核），是指“提供使用者使用界面”的软件（命令解析器）。它类似于DOS下的command和后来的cmd.exe。它接收用户命令，然后调用相应的应用程序。

**#系统调用**

\*系统调用提供了操作系统提供的有效服务界面。Programming interface to the services provided by the OS.

\*Mostly accessed by programs via a high-level Application Programming Interface (API) rather than direct system call use. 通常程序使用它是通过API而不是直接的系统调用。

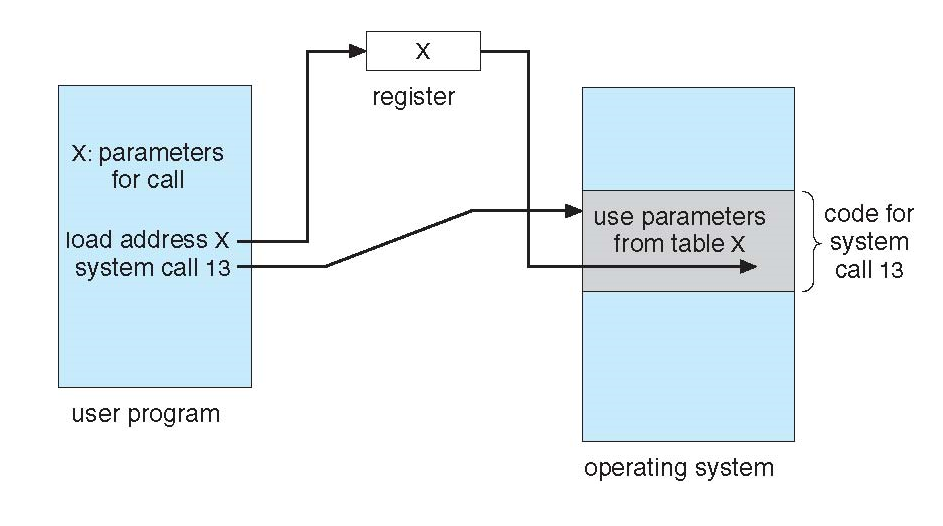


操作系统传递参数有三种方法：

1. 通过寄存器来传递参数

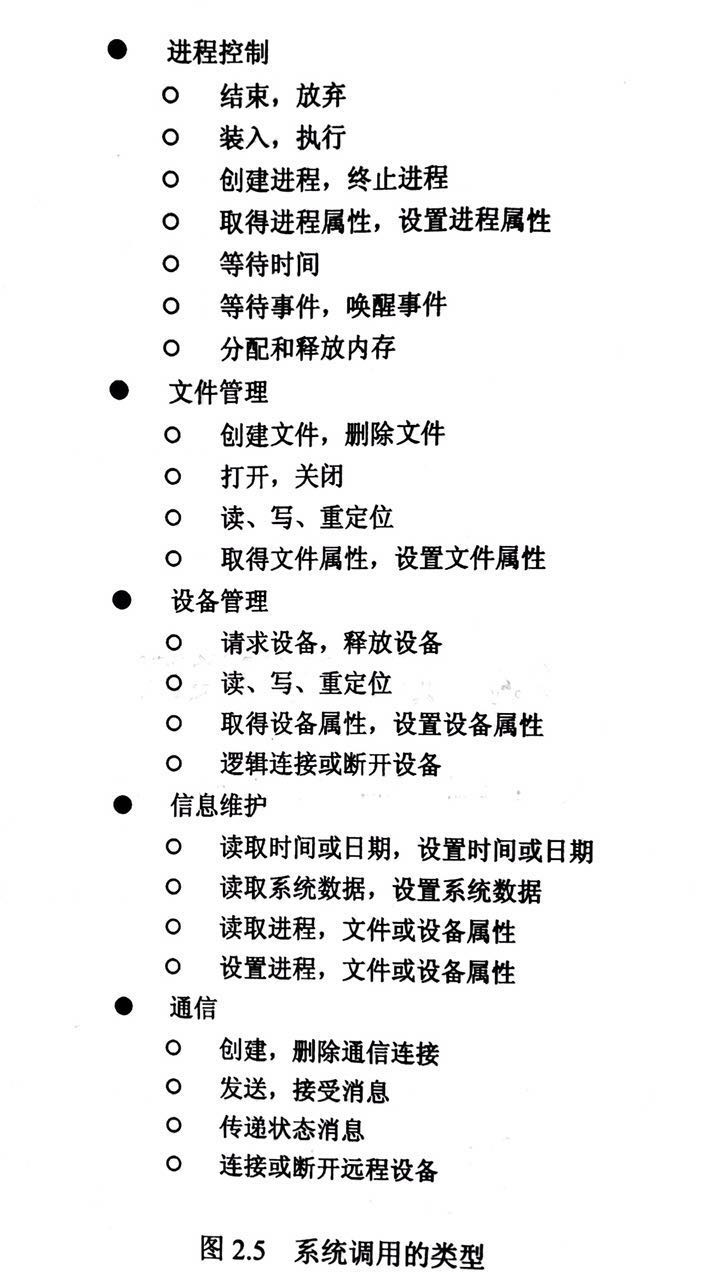
2. 参数存储在内存的块或表中，将地址通过寄存器来传递

3. 程序将参数压入（pushed）堆栈中并通过操作系统弹出（popped）。

表传递

**#系统调用的类型**

系统调用可以大致分为五大类：进程控制、文件管理、设备管理、信息维护和通信。



**#系统程序**

System programs provide a convenient environment for program development and execution. 系统程序提供了一个方便的环境，用来开发程序和执行程序。

理解：现代系统的另一方面是一组系统程序。计算机的逻辑层次为最底层是硬件，上面是操作系统，接着是系统程序，最后是应用程序。

系统程序可分为：

文件管理File management，文件的创建删除键复制等。

状态信息Status information，程序从系统哪里得到日期时间，可用内存或磁盘空间的数量等

文件修改File modification，修改文件内容

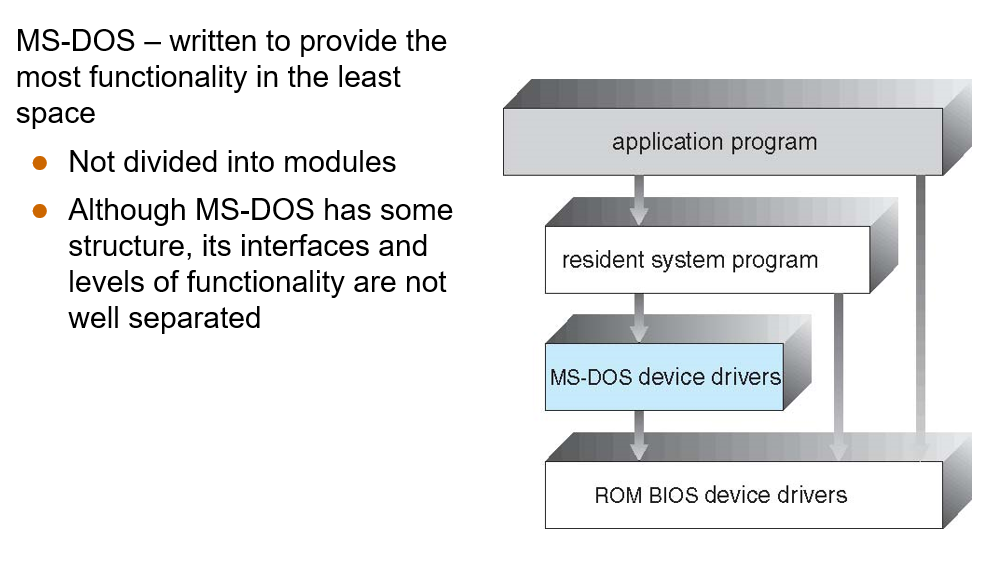
程序语言支持Programming-language support，常用的程序设计语言

程序的装入和执行Program loading and execution，一旦程序汇编和编译后，它必须装入内存才能执行

通信Communications，进程，用户，计算机系统之间创建虚拟连接的机制。

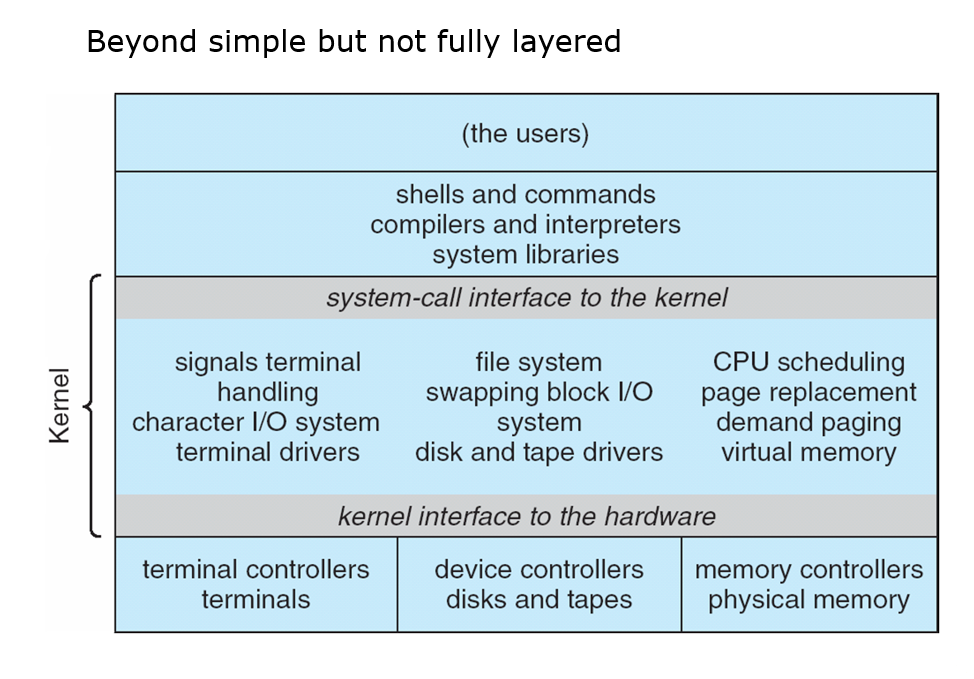
**#操作系统结构**

\*简单结构-MS-DOS



没有很好的区分接口和功能层次，应用程序可以直接访问基本的IO子程序，导致该系统容易受错误程序的伤害。

\*更复杂的结构-UNIX



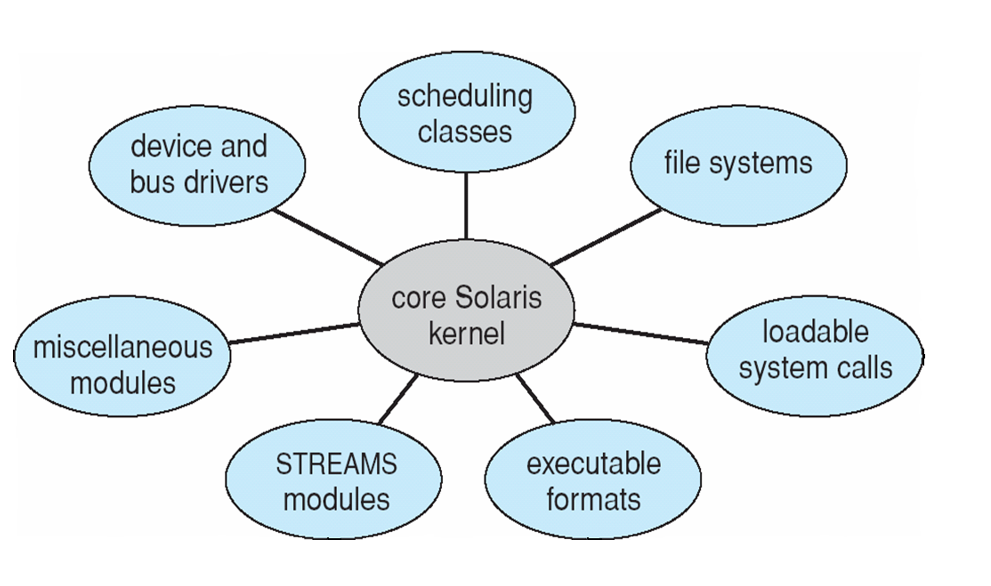
内核和系统程序分开。

Tips: 系统模块化的方法——分层法。每层都是利用较低层所提供的功能来实现的。该层不必知道如何实现这些操作，它只需要知道这些操作能做什么。因此，每层为较高层隐藏了一定的数据结构、操作和硬件的存在。

\*微内核

\*模块

最新的操作系统设计方法是面向对象编程技术来生成模块化的内核。



这样的设计允许内核提供核心服务，也能动态地实现特定的功能。

**#操作系统查错**

Debugging is finding and fixing errors, or bugs.

操作系统会生成日志文件来存储错误信息。App的失败会产生core damp来捕获进程的内存信息。

操作系统的失败会产生crash dump来存储内核内存。

**#操作系统生成**

Operating systems are designed to run on any of a class of machines; the system must be configured for each specific computer site. 可以为某处的某台机器专门设计、编写和实现操作系统。对此，必须要配置和生成系统，这一过程有时称为系统生成。

操作系统通常通过CD-ROM来发布。

**#系统启动**

1. 当系统通电后，首先修复（重新装载）内存位置，并且开始执行。

2. Small piece of code – bootstrap loader, stored in ROM or EEPROM locates the kernel, loads it into memory, and starts it.

Sometimes two-step process where boot block at fixed location loaded by ROM code, which loads bootstrap loader from disk.

3. 一般的引导程序被装载，开始选择内核。

4. 内核被加载，系统开始运行。