

第9章 操作系统接口 (2学时)



主讲教师：张春元

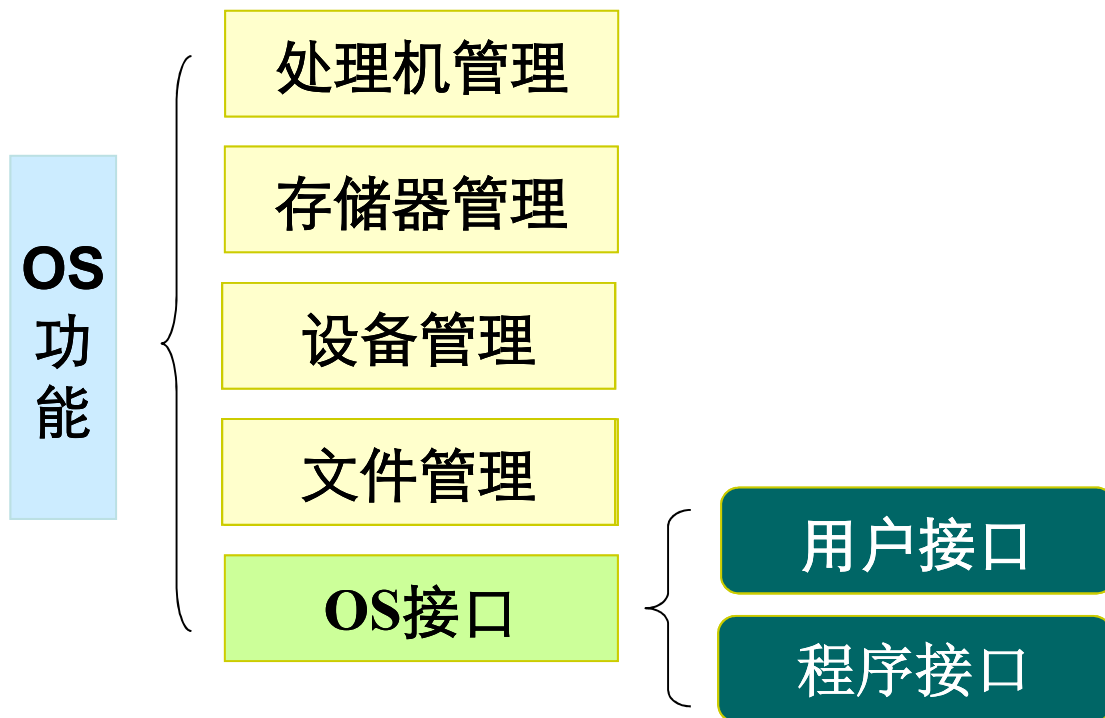
联系电话：13876004640

课程邮箱：haidaos@126.com

邮箱密码：[zhangchunyuan](#)



本章内容所处位置





操作系统接口

❖ 1> 用户接口

* 脱机用户接口（批处理用户接口）

- 作业说明书控制作业的运行
- 不能实现用户与计算机间的交互

* 联机用户接口（联机命令接口）

- 字符显示式用户界面
- 图形化用户界面

❖ 2> 程序接口

- * 程序员编程时使用，又称应用程序接口（API）
- * 程序通过系统调用取得OS的服务



本章主要内容

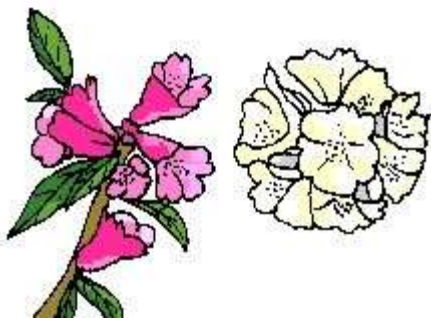
- ❖ 9.1 用户接口
- ❖ 9.2 Shell命令语言
- ❖ 9.3 联机命令接口的实现
- ❖ 9.4 系统调用的概念与类型
- ❖ 9.5 UNIX系统调用
- ❖ 9.6 系统调用的实现





9.1 联机命令接口

- ❖ 9.1.1 字符显示式联机用户接口
- ❖ 9.1.2 图形化联机用户接口
- ❖ 9.1.3 联机命令的类型





9.1.1 字符显示式联机用户接口

❖ 1、字符显示式联机用户接口

- * 又称**联机命令接口**，是指用户通过命令语言实现地作业的控制，以及取得操作系统的服务。
- * **命令语言的两种方式**
 - **1> 命令行方式**
 - 以行为单位输入和显示命令
 - **2> 批命令方式**
 - 用户把一系列命令组织在一种称为批命令文件的文件中，一次建立，多次执行
 - **UNIX/LINUX可用Shell命令语言编写批命令文件**
 - **DOS中的.BAT文件**



9.1.2 图形化联机用户接口

❖ 1、图形化用户接口（GUI）

- * **GUI:** 使用**WIMP**技术，将窗口**Window**、图标**Icon**、菜单**Menu**和鼠标**Pointing Device**等集成在一起，引入形象的各种图符将系统中的各项功能、各种应用程序和文件，直观、逼真地表示出来，形成一个图文并茂的操作环境。
- * 1981年，**Xerox公司**在Star 8010工作站上的OS中首次推出了图形用户界面。
- * 1983年，Apple公司在Lisa机和Macintosh机上的OS中使用了图形用户界面。
- * **Windows、IBM OS/2、UNIX X-Window**等随后均采用了图形用户界面。



9.1.3 联机命令的类型

❖ 1、系统登录类命令

- * 多用户系统登录命令 **login:** **password:**

❖ 2、磁盘操作类命令

- * 1> 磁盘格式化命令 **Format**
- * 2> 复制整个软盘命令 **Diskcopy**
- * 3> 软盘比较命令 **Diskcomp**
- * 4> 备份命令 **Backup**
- * 5> 恢复命令 **Restore**

❖ 3、文件操作命令

- * 1> 文件内容显示命令 **type**



9.1.3 联机命令的类型

- * 2> 文件拷贝命令 **copy**
- * 3> 文件比较命令 **comp**
- * 4> 重新命名命令 **Rename**
- * 5> 删除文件命令 **erase**

❖ 4、目录操作命令

- * 1> 建立子目录命令 **mkdir**
- * 2> 显示显录命令 **dir**
- * 3> 删除子目录命令 **rmdir**
- * 4> 显示目录结构命令 **tree**
- * 5> 改变当前目录命令 **chdir**



9.1.3 联机命令的类型

❖ 5、其它命令

- * 1> 输入输出重定向命令 >、<
- * 2> 管道连接命令 `Command1|Command2|...|Commandn`
- * 3> 过滤命令 `find/N` “字符串” 路径名
- * 4> 批命令
 - MS_DOS: BAT文件
 - UNIX: Shell文件



本章主要内容

- ❖ 9.1 用户接口
- ❖ 9.2 Shell命令语言
- ❖ 9.3 联机命令接口的实现
- ❖ 9.4 系统调用的概念与类型
- ❖ 9.5 UNIX系统调用
- ❖ 9.6 系统调用的实现





9.2 Shell命令语言

❖ 1、Shell是什么？

- * Shell位于UNIX最外层，是用户和UNIX的接口。
- * Shell是命令级程序设计语言。
 - 可作为命令语言，用户用其与UNIX交互
 - 可作为程序设计语言，用来编写Shell程序
- * Shell包括有 Shell命令解释程序。



9.2 Shell命令语言

❖ 2、Shell 简单命令

- * 是指一条命令行中仅有一个命令，实际上是一个能完成某种功能的目标程序的名字。
- * 格式：命令 -选项 参数列表
- * 简单命令的分类
 - 系统命令 和 用户自定义命令
 - 内部命令 和 外部命令



9.2 Shell命令语言

❖ 3、Shell 的种类

* 1> Bourne Shell (简称B shell)

- 具有多个版本，最著名的是Bourne Again Shell (**Bash**), 是Linux中缺省的Shell

* 2> C Shell

- 比B Shell更适于编程，其语法和C语言相似

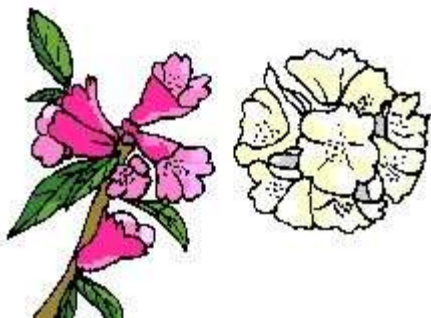
* 3> Korn Shell

- 集合了C Shell和B Shell的优点



9.2 Shell命令语言

- ❖ 9.2.1 简单命令
- ❖ 9.2.2 重定向和管道命令
- ❖ 9.2.3 通信命令
- ❖ 9.2.4 后台命令





9.2.1 简单命令

❖ 1、系统登录命令

* 进入系统 **login :**

* 退出系统 **ctrl+D**

❖ 2、文件操作命令

* 1> 文件内容显示命令 **cat 文件名**

* 2> 文件或目录复制命令 **cp source target**

* 3> 文件或目录更名命令 **mv oldname newname**

* 4> 文件或目录删除命令 **rm 文件名或目录名**

* 5> 文件类型显示命令 **file 文件名**



9.2.1 简单命令

❖ 3、目录操作命令

- * 1> 目录建立命令 **mkdir** 目录名
- * 2> 目录删除命令 **rmdir** 目录名
- * 3> 工作目录改变命令 **cd** 目录名
- * 4> 文件存取权限改变命令 **chmod** 权限 文件名

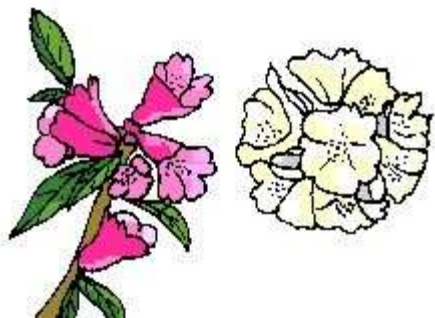
❖ 4、系统询问命令

- * 1> 当前日期及时间查看命令 **date**
- * 2> 系统当前用户查看命令 **who**
- * 3> 当前目录路径名查看命令 **pwd**



9.2 Shell命令语言

- ❖ 9.2.1 简单命令
- ❖ 9.2.2 重定向和管道命令
- ❖ 9.2.3 通信命令
- ❖ 9.2.4 后台命令





9.2.2 重定向与管道命令

❖ 1、重定向命令

* 1> 输出定向 > >>

■ 例: `cat file1>file3` `cat file2>>file3`

* 2> 输入定向 <

■ 例: `wc<file1`

* 3> 输入、输出重定向

■ 例: `a.out<file1>file0`

❖ 2、管道命令

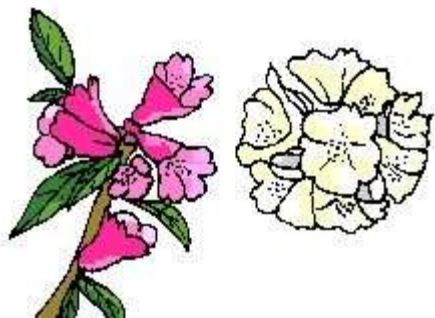
* `Command1|Command2|...|Commandn`

* 管道命令具有单向性、同步性



9.2 Shell命令语言

- ❖ 9.2.1 简单命令
- ❖ 9.2.2 重定向和管道命令
- ❖ 9.2.3 通信命令
- ❖ 9.2.4 后台命令





9.2.3 通信命令

❖ 1、间接通信——信箱通信命令mail

- * 信箱地址 /usr/spool/mail
- * 发送者和接收者均通过mail命令完成通信

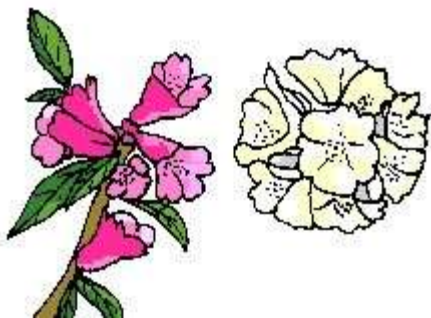
❖ 2、直接通信

- * 1> 对话通信命令write
 - write 用户帐号[终端名称]
- * 2> 允许或拒绝接收消息命令mesg
 - mesg [y/n]
 - 用于设定是否允许其它用户用write命令给自己发送消息



9.2 Shell命令语言

- ❖ 9.2.1 简单命令
- ❖ 9.2.2 重定向和管道命令
- ❖ 9.2.3 通信命令
- ❖ 9.2.4 后台命令





9.2.4 后台命令

❖ 后台命令

- * 对于执行时间较长的命令可在其命令后面添加“&”号，将其转入后台，等系统有空时再执行。
- * 后台命令的输入应加以重定向，以避免其与前台别的命令同时从终端进行读入造成混乱。



本章主要内容

- ❖ 9.1 用户接口
- ❖ 9.2 Shell命令语言
- ❖ 9.3 联机命令接口的实现
- ❖ 9.4 系统调用的概念与类型
- ❖ 9.5 UNIX系统调用
- ❖ 9.6 系统调用的实现





9.3.1 键盘终端处理程序

❖ 1、字符接收功能

* 1> 面向字符方式接收

- 输入的字符不加修改地送给用户程序

* 2> 面向行方式接收

- 输入的字符暂存在行缓冲中，仅当一行输入结束才将整行信息送命令解释程序

❖ 2、字符缓冲功能

* 1> 专用缓冲区方式

- 为每一终端设置一个缓冲区，利用率较低

* 2> 公用缓冲池方式

- 所有终端共享一个缓冲池，利用率高



9.3.1 键盘终端处理程序

❖ 3、回送显示

- * 回送显示是指每当用户从键盘输入一个字符后，终端处理程序便将该字符送往屏幕显示。
- * 硬件回显（速度快，但缺乏灵活性）
- * 软件回显（速度较慢，但灵活）

❖ 4、屏幕编辑

- * 允许用户对从键盘输入的字符进行修改，常用编辑键有：删除字符键、删除一行键、插入键、移动光标键、屏幕上卷或下移键。



9.3.1 键盘终端处理程序

❖ 5、特殊字符处理

- * 1> 中断字符：用来中止当前程序的运行
 - Break、Delete、Ctrl+C
- * 2> 停止上卷字符：使正在上卷的屏幕停止上卷
 - Ctrl+S
- * 3> 恢复上卷字符：使停止上卷的屏幕恢复上卷
 - Ctrl+Q



9.3.2 命令解释程序

❖ 1、常见命令解释程序（用户与OS之间的接口）

- * MS-DOS: COMMAND.COM
- * UNIX: Shell

❖ 2、命令解释程序的作用

- * 在屏幕上给出提示符，请用户键入命令，然后读入该命令，识别命令，再转到相应命令处理程序的入口地址，把控制权交给该处理程序去执行，并将处理结果送屏幕上显示。
- * 若用户键入的命令有错，而命令解释程序未能予以识别，或在执行中间出现问题时，则应显示出某一出错信息。



9.3.2 命令解释程序

❖ 3、MS-DOS命令解释程序

❖ A、COMMAND.COM的组成

* 1> 常驻内存部分

- 包括一些中断服务子程序

* 2> 初始化部分

- 在常驻内存部分之后，在OS启动时获得控制权，包括对Autoexec.bat文件的处理程序，并决定应用程序装入的基地址。

* 3> 暂存内存部分

- 这部分主要是命令解释程序、内部命令处理程序、批文件处理程序、外部命令程序等，在用户程序不运行时暂存内存中，用户程序运行时将覆盖掉这部分内存



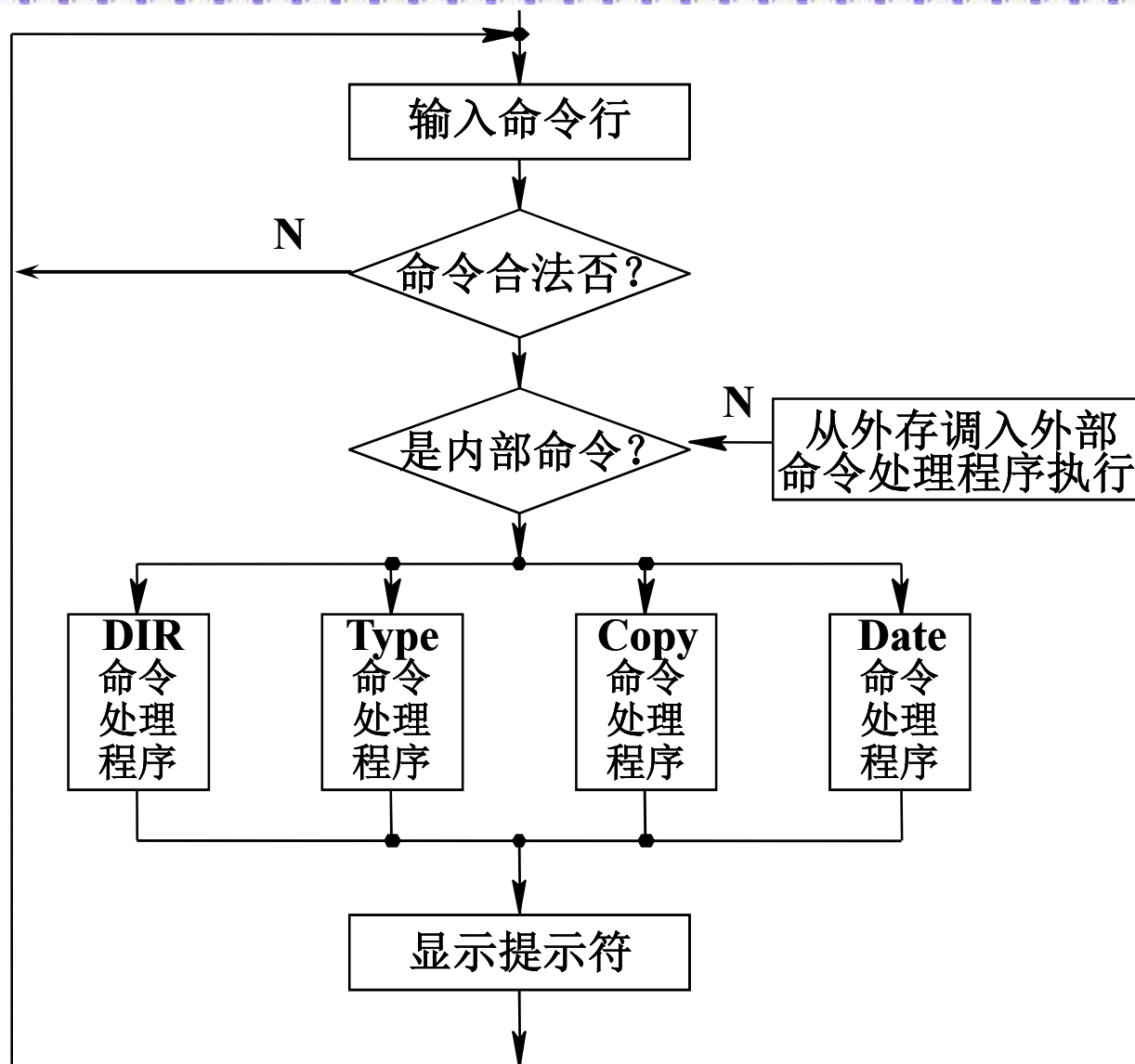
9.3.2 命令解释程序

❖ B、COMMAND.COM的工作流程

- * 1> 系统通电或复位后，**初始化部分**获得控制权：对整个系统完成初始化工作，并自动执行AUTOEXEC.BAT文件。
- * 2> **暂存部分**获得控制权：读入键盘缓冲区中的命令，判断该命令**是否合法**：若有错，给出出错信息后返回；若无错，再识别该命令是内部命令还是外部命令：若是**内部命令**，则把控制权交给该命令的处理程序以执行该命令；若是**外部命令**，为之建立命令行，装入该命令的处理程序，将控制权交给该程序执行。



COMMAND.COM的工作流程





9.3.2 命令解释程序

❖ 4、Shell命令解释程序

* A、Shell命令的特点

- 1> 一条命令行中有多个命令
- 2> 具有不同的分隔符
- 注：上述特点使得Shell命令不能采用简单的译码方式，而是按一定规律转换成二叉树结构的命令行树。



9.3.2 命令解释程序

* B、Shell命令解释程序的工作流程

- 1> 读取用户由键盘输入的命令行。
- 2> 对命令进行分析，建立相应的二叉树结构命令行树。
- 3> 对二叉树结构命令行树的每一条命令建立相应的子进程。
- 4> 等待子进程完成：对“;”型结点，需在其左子树执行完成后，才可继续处理下一条命令；对“&”型结点，在启动其左子结点执行后，因它是后台命令，不需要待就可执行其右子树。



本章主要内容

- ❖ 9.1 用户接口
- ❖ 9.2 Shell命令语言
- ❖ 9.3 联机命令接口的实现
- ❖ 9.4 系统调用的概念与类型
- ❖ 9.5 UNIX系统调用
- ❖ 9.6 系统调用的实现





9.4.1 系统调用的基本概念

❖ 1、系统态和用户态

在计算机系统中，为了保证OS程序不被应用程序（用户程序）有意或无意破坏，将处理机的执行状态分为系统态和用户态两种：

- * **系统态(管态、核心态)**：有较高特权，能执行一切指令，访问所有寄存器和存储区。
 - OS内核一般运行在系统态。
- * **用户态(目态)**：只能执行规定指令，访问指定寄存器和存储区。
 - 用户程序只能运行在用户态。
 - 系统调用会引起CPU从用户态转入核心态。



9.4.1 系统调用的基本概念

❖ 2、特权指令和非特权指令

根据CPU的执行状态，大多数OS将CPU的指令集分为**特权指令**和**非特权指令**两种：

- * **特权指令**：可在系统态下运行的指令，是关系到系统全局的指令，对内存空间的访问基本上不受限制。
 - 特权指令只允许操作系统使用，不允许用户程序使用。
- * **非特权指令**：只能在用户态下运行的指令，只能完成一般性的操作和任务，不能对系统中的硬件和软件**直接**访问，对内存的访问范围仅局限于用户空间。



9.4.1 系统调用的基本概念

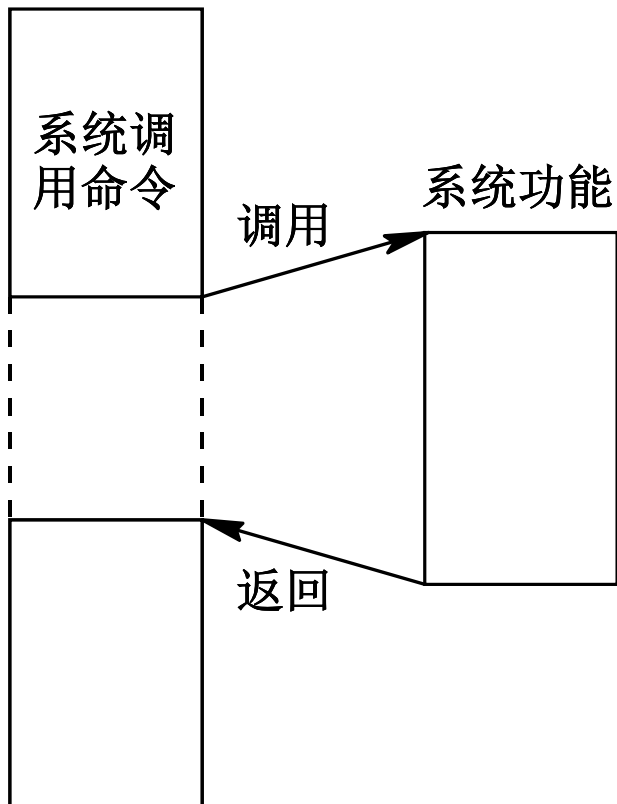
❖ 3、系统调用

系统调用是用户程序间接调用OS内核函数完成某功能的一种特殊调用。系统调用具有如下特性：

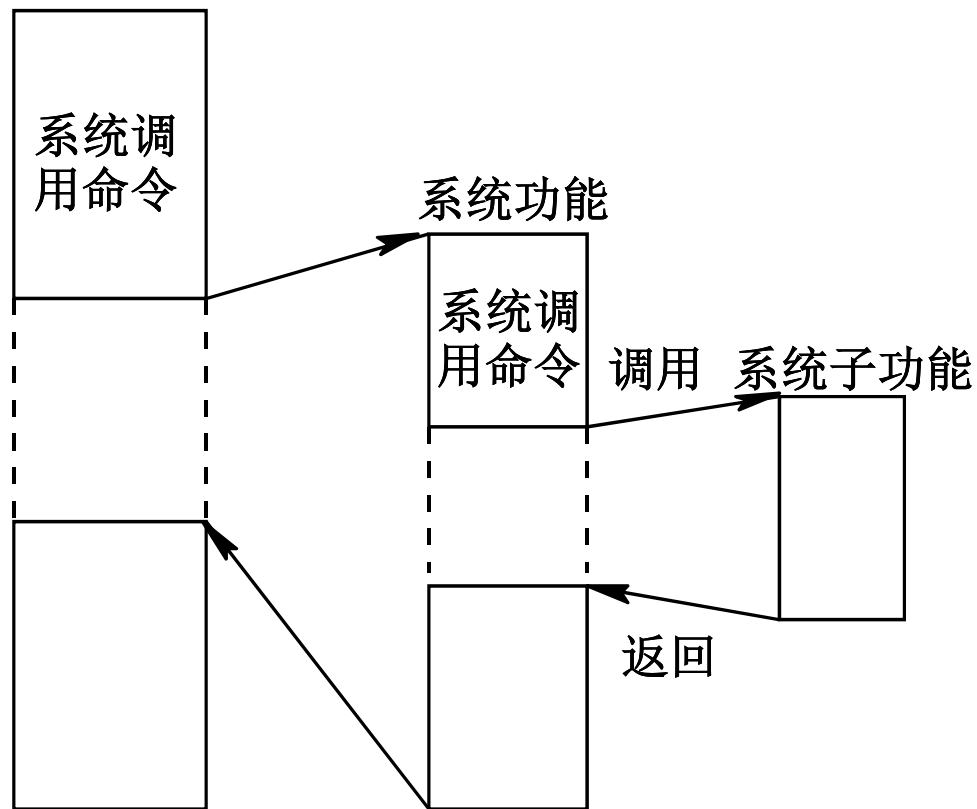
- * 1> 运行在不同的系统状态：调用程序运行在用户态，被调用程序运行在系统态。
- * 2> 状态的转换通过软中断进入：用户程序进行系统调用时，先由用户态转至系统态，经过核心分析后，才转向相应的系统调用处理子程序。
- * 3> 可嵌套调用：在一个系统调用函数执行期间，还可利用系统调用命令调用另一个系统的调用函数。
- * 4> 返回问题：系统调用完毕返回时可能引起进程的重新调度。

9.4.1 系统调用的基本概念

用户程序



(a) 系统调用与返回



(b) 系统函数嵌套调用

图9-6 系统功能的调用



9.4.2 系统调用的类型

❖ 1、进程控制类系统调用

- * 1> 创建和终止进程的系统调用
- * 2> 获得和设置进程属性的系统调用
- * 3> 等待某事件出现的系统调用

❖ 2、文件操纵类系统调用

- * 1> 创建和删除文件的系统调用
- * 2> 打开和关闭文件的系统调用
- * 3> 读和写文件的系统调用

❖ 3、进程通信类系统调用

❖ 4、设备管理类系统调用

❖ 5、系统维护类系统调用



9.4.3 POSIX标准

❖ 什么是POSIX?

- * POSIX (Portable Operating System Interface of UNIX), 即**基于UNIX的可移植操作系统接口**。
- * POSIX定义了**标准应用程序接口(API)**, 以保证遵循此标准的应用程序具有较好的可移植性, 即能够在不同操作系统上运行。
- * POSIX标准**定义了一系列过程**, 大多数系统调用对应其中某一个过程, 也有少量系统调用对应其中多个过程。
- * UNIX新版本、Linux、Windows、OS/2等操作系统中的系统调用过程采用C语言编写, 并以**库函数**的形式提供。**但一般来说, 库函数属于用户程序而非系统程序。**



9.4.3 POSIX标准

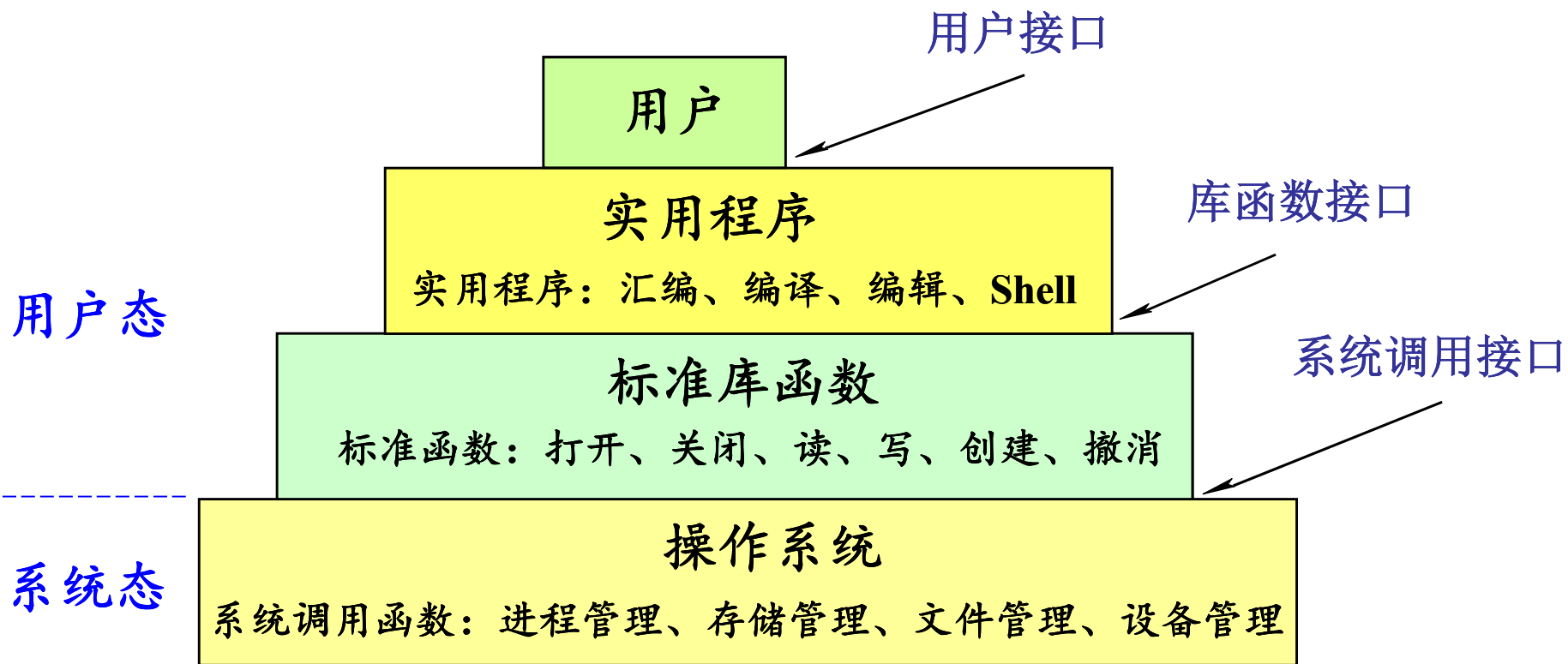


图9-7 Unix/Linux实用程序、库函数、系统调用函数的分层关系



本章主要内容

- ❖ 9.1 用户接口
- ❖ 9.2 Shell命令语言
- ❖ 9.3 联机命令接口的实现
- ❖ 9.4 系统调用的概念与类型
- ❖ 9.5 UNIX系统调用
- ❖ 9.6 系统调用的实现





9.5 UNIX系统调用

UNIX系统早期版本提供有56种系统调用函数，现已增至数百条，常用的大约有30来条。

❖ 1、进程控制类系统调用函数

- * 创建进程fork
- * 终止进程exit
- * 等待子进程结束wait
- * 执行一个文件exec
- * 获得进程ID getpid getppid getpgrp
- * 获得用户ID getuid geteuid getgid
- * 进程暂停pause



9.5 UNIX系统调用

❖ 2、文件操纵类系统调用函数

- * 创建文件**creat**
- * 打开文件**open**
- * 关闭文件**close**
- * 读文件**read**和写文件**write**
 - 三个参数：① 文件描述符**fd**； ② **buf**缓冲区首址；
③ 用户要求传送的字节数***n* byte**
- * 文件连接**link**
- * 去文件连接**unlink**



9.5 UNIX系统调用

❖ 3、进程间通信类系统调用函数（IPC软件包）

* 1> 消息机制

- `msgget`创建一个消息队列
- `msgsend`向指定消息队列发送消息
- `msgrcv`从指定消息队列接收指定类型的消息

* 2> 共享存储器机制

- `shmget`创建一个共享存储区
- `shmat`实现进程与共享存储区的连接
- `shmdt`拆除进程与共享存储区的连接

* 3> 信号量机制

- 与P53信号量集类似，详见实验三



9.5 UNIX系统调用

❖ 4、系统维护类系统调用函数

- * 设置系统时间**stime**
- * 获得系统时间**time**
- * 获得进程和子进程所使用的CPU时间**times**
- * 设置文件的访问和修改时间**utime**
- * 获得当前UNIX系统的名称**uname**



本章主要内容

- ❖ 9.1 用户接口
- ❖ 9.2 Shell命令语言
- ❖ 9.3 联机命令接口的实现
- ❖ 9.4 系统调用的概念与类型
- ❖ 9.5 UNIX系统调用
- ❖ 9.6 系统调用的实现





9.6 系统调用的实现

❖ 1、中断和陷入硬件结构

* 1> 中断和陷入的概念

中断是CPU对系统发生某事件的一种响应，按中断信号的来源不同，中断可分为：

- 外中断一般又称中断，是指来自处理器和主存储器之外的中断，包括电源故障中断、时钟中断、控制台中断、它机中断和I/O中断等。
- 内中断一般又称陷入或异常，是指来自处理器和主存内部的中断，包括通路校验错、主存奇偶错、非法操作码、地址越界、页面失效、调试指令、算术操作溢出等各种程序性中断等。由系统调用引起的中断属于内中断，由系统调用引起中断的指令称为陷入指令。



9.6 系统调用的实现

* 2> 中断和陷入的区别

- **中断（外中断）**是由与现行指令无关的中断信号触发的，通常在两条机器指令之间才可以响应中断，一般来说，中断处理程序提供的服务不是为当前进程所需的。
- **陷入**则是由处理器正在执行现行指令而引起的，因而一条指令执行期间允许响应陷入，通常陷入处理程序提供的服务是为当前进程所用的。陷入是不能被屏蔽的，一旦出现应立即响应。



9.6 系统调用的实现

* 3> 中断和陷入向量

- 中断向量由中断处理程序的入口地址和中断处理程序的程序状态字PSW（PSW中预先设置好了中断处理程序需要的处理机执行状态、处理机优先级等）两个字组成。
- 存放中断向量的单元称为**中断向量单元**。
- 所有中断向量统一组织存放在**中断向量表**中，中断处理时根据中断类型号可从中查找到相应的中断向量。
- **陷入向量**与中断向量类似。



9.6 系统调用的实现

❖ 2、系统调用号和参数的设置

- * 为了便于对所有系统调用函数进行管理和使用，系统将为每一系统调用函数赋予唯一的**系统调用号**。
- * 用户程序进行系统调用时根据系统调用号来查找**系统调用入口表（系统调用散转表）**，从中找出对应的系统调用函数的入口地址。
- * 系统调用函数运行时通常需要**传入相应的参数**，常见的传入方式有：
 - 1> 陷入指令自带方式
 - 2> 直接将参数传到相应的寄存器中：**MS-DOS**
 - 3> 参数表方式：**UNIX、Linux**



9.6 系统调用的实现

❖ 3、系统调用的处理步骤

* 用户程序（用户态）

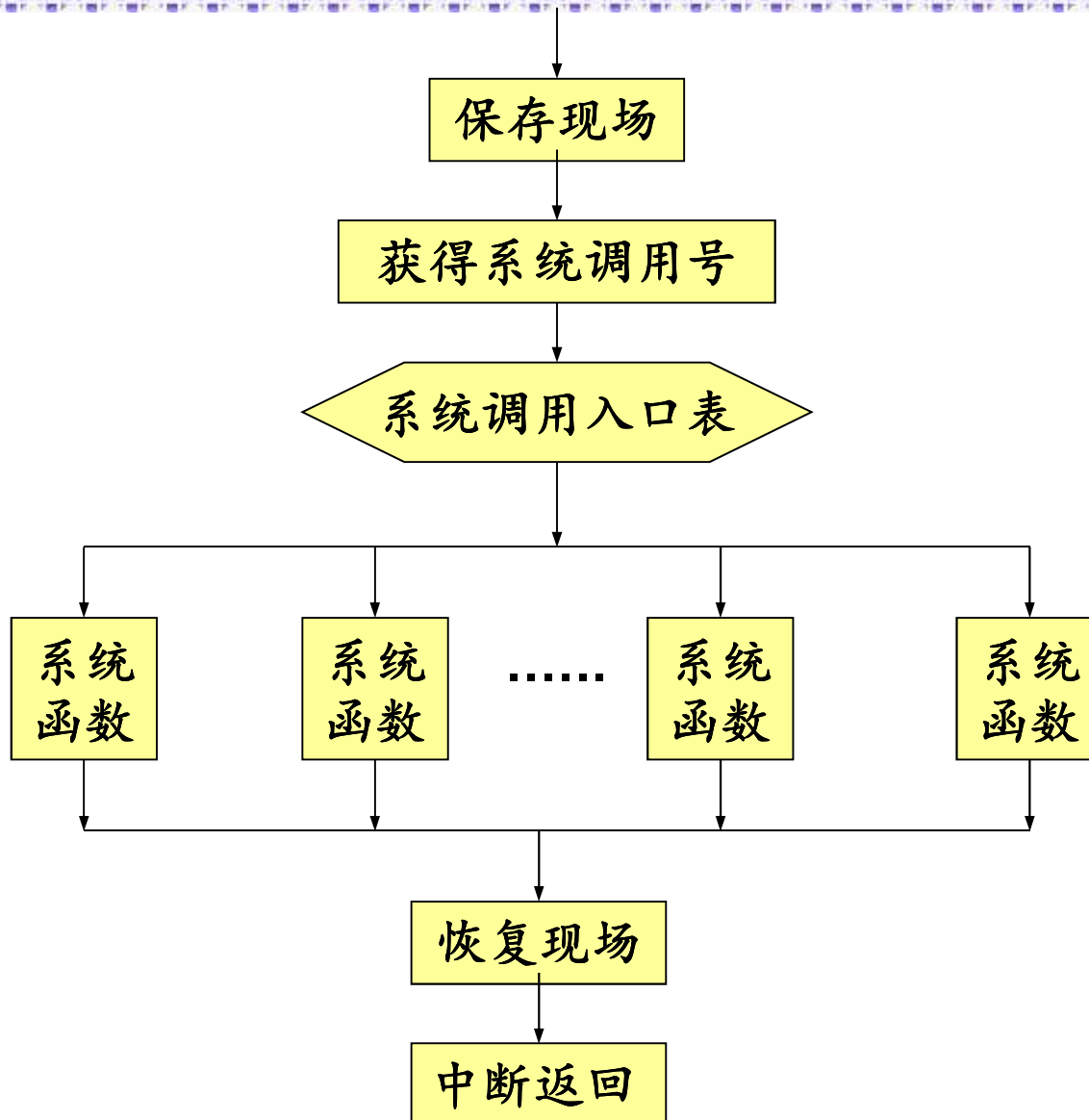
- ① 设置系统调用号和参数→约定的寄存器
- ② 执行陷入指令trap
- ③ 从约定的寄存器中取得返回值

* 操作系统（系统态）

- ① 通过中断/陷入机制执行系统调用总控程序
- ② 总控程序从约定的寄存器获得系统调用号和参数，根据系统入口表调用相应的系统调用函数
- ③ 系统调用函数执行完后，返回总控程序
- ④ 总控程序：返回值→约定的寄存器
- ⑤ 通过中断/陷入机制返回用户程序



系统调用处理流程





本章小结

- ❖ OS的接口分类*
- ❖ 键盘终端处理程序的功能*
- ❖ COMMAND.COM、Shell的工作流程*
- ❖ Shell命令语言*
- ❖ 系统调用的实现**
- ❖ UNIX系统调用
- ❖ 图形用户接口
- ❖ 重要概念**



**掌握
*理解

- * 用户态、系统态、特权指令、非特权指令、系统调用、内中断、外中断、中断向量



本章作业

❖ 作业内容:

- * 操作系统第9章网络在线测试 (6月14日前完成)



本章课程结束！谢谢大家！