

第六章 设备管理

目的与要求： 理解设备管理子系统的层次、功能及技术

重点与难点： 层次结构，设备驱动程序，缓冲技术，虚拟设备技术

设备I/O子系统的主要目录

1、I/O层次结构

2、缓冲技术

3、设备分配

4、虚拟设备

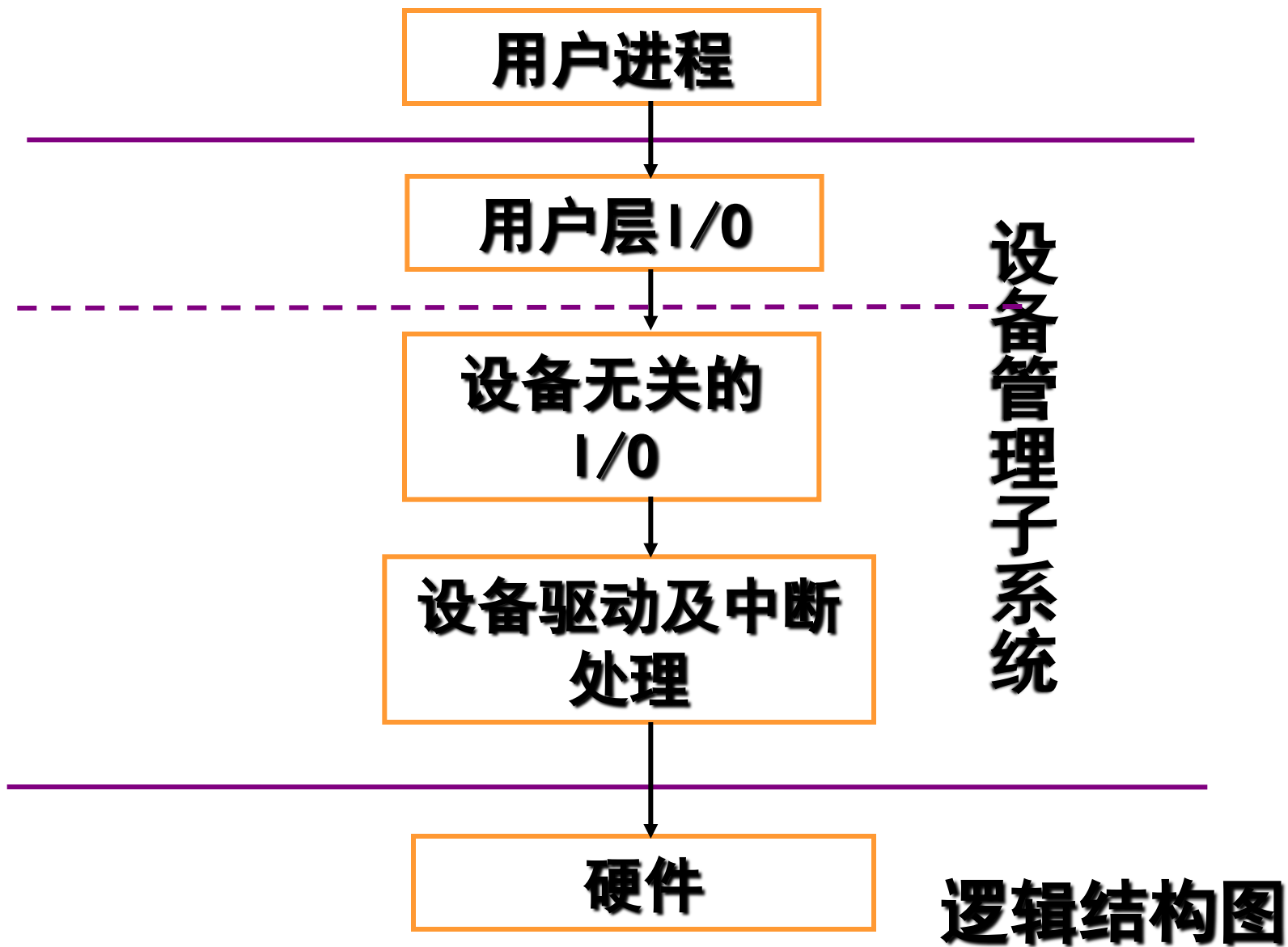
1、I/O软件的设计目标和原则

I/O软件总体设计目标:

- 高效率。
- 通用性。

I/O软件总体设计要考虑的问题:

- 设备无关性
- 出错处理
- 同步（阻塞）—异步（中断驱动）传输
- 独占性外围设备和共享性外围设备



系统调用各模块关系图



(1) 用户层I/O

与**设备**的控制细节**无关**，不直接与设备打交道。将所有的设备看做**逻辑资源**

为用户进程提供各类**I/O函数**。用户以**设备标识符**和一些简单的**函数**来使用设备，如打开、关闭、读、写等

如C库中的函数fopen(), fread(), fwrite(), fclose()等

(2) 设备无关的I/O

执行适用于所有设备的**公共I/O功能**，并向其上层**提供统一的系统调用接口**。

主要任务包括：

1. **设备命名及与设备驱动程序的映射。**
在UNIX中，如/dev/tty00惟一地确定了一个inode数据结构，其中包含了主设备号，通过主设备号可以找到相应的设备驱动程序。
2. **设备访问保护。**
3. **分配及释放独占型设备。**
4. **设备I/O数据缓冲机制。**
5. **文件系统功能、TCP/IP协议等。**

(3) 设备驱动与中断处理

1. 设备驱动程序

包括了所有与设备相关的代码，其功能是从与设备无关的软件中接收I/O的请求，排入请求队列或执行之。

2. 中断处理

当进程进行I/O操作时，将其阻塞至I/O操作结束并发生中断。中断发生时，由中断处理程序启动请求排队的下一请求并解除等I/O进程的阻塞状态，使其能够继续执行。

Linux 设备管理实例

- 屏蔽各类硬件设备，提供统一操作
- Linux内核：提供硬件抽象层、磁盘及文件系统的控制，处理机调度，内存分配与管理的系统软件。采用可加载模块化设计（LKMs），支持可插入式模块。有三种类型
 - 编译在内核中
 - 可动态加载的模块
 - 不加载，从Linux中取消
- 驱动程序：动态加载驱动模块，如声卡、网卡驱动。Linux基本驱动，如CPU、PCI总线、ACPI、VFS等编译在内核中

Linux中加载硬件设备

- **#>Dmesg | less命令**
- **查看Linux的启动过程**
- **#>ls pci**
- **查看硬件设备**

Linux中的模块

- 列出系统中已加载的模块
- `#>ls mod`
- 删除模块
- `#>rmmod cdrom`
- 加载模块
- `#>modprobe cdrom`
- 可通过配置内核加载模块

2、缓冲技术

在进程数据区与外设之间设立缓冲区可以改善进程运行速度与I/O传输速度之间的速差矛盾。

(1) 单缓冲

当用户进程发出I/O请求时，操作系统在主存的系统空间为该操作分配一个缓冲区，可以实现预读和滞后写。

问题：说明为什么在单、双缓冲的情况下，对一块（假定一块缓冲正好存放一块）信息的处理时间分别为 $\text{Max}(C, T) + M$ 和 $\text{Max}(C, T)$ ？

(2) 双缓冲

可以实现用户数据区—缓冲区之间交换数据和缓冲区—外设之间交换数据的并行

(3) 循环缓冲

引入系统缓冲池，采用有限缓冲区的生产者/消费者模型对缓冲池中的缓冲区进行循环使用

缓冲区结合预读和滞后写技术对具有重复性及阵发性I/O进程、提高I/O速度很有帮助。

3、设备分配

- **（1）设备的独立性**
- **（2）与设备被分配有关的数据结构**

(1)设备独立性

- 用户不指定特定的设备，而指定逻辑设备，使得用户作业和物理设备独立开来，再通过其它途径建立逻辑设备和物理设备之间的对应关系，称这种特性为“设备独立性”。
- 设备独立性带来的好处
用户与物理的外围设备无关，系统增减或变更外围设备时程序不必修改；易于对付输入输出设备的故障。

设备分配的数据结构

- 设备类表和设备表。
- 系统中拥有一张设备类表，每类设备对应于表中一栏，包括内容有：设备类、总台数、空闲台数和设备表起始地址等。
- 每一类设备都有各自的设备表，用来登记这类设备中每一台设备的状态，包含的内容有：物理设备名、逻辑设备名、占有设备的进程号、已分配/未分配、好/坏等。

设备分配的数据结构

- 采用**通道结构**的系统中，设备分配的数据结构设置：**系统设备表、通道控制表、控制器控制表和设备控制表**。
- 系统建立一张系统设备表，记录配置在系统中的所有物理设备的情况。
- 每个通道、控制器、设备各设置一张表，记录各自的地址(标识符)、状态(忙/闲)、等待获得此部件的进程队列指针、及一次分配后相互勾链的指针，以备分配和执行I/O时使用。

4、虚拟设备

(1) 问题的提出

(2) SPOOLING的设计和实现

(3) SPOOLING应用例子

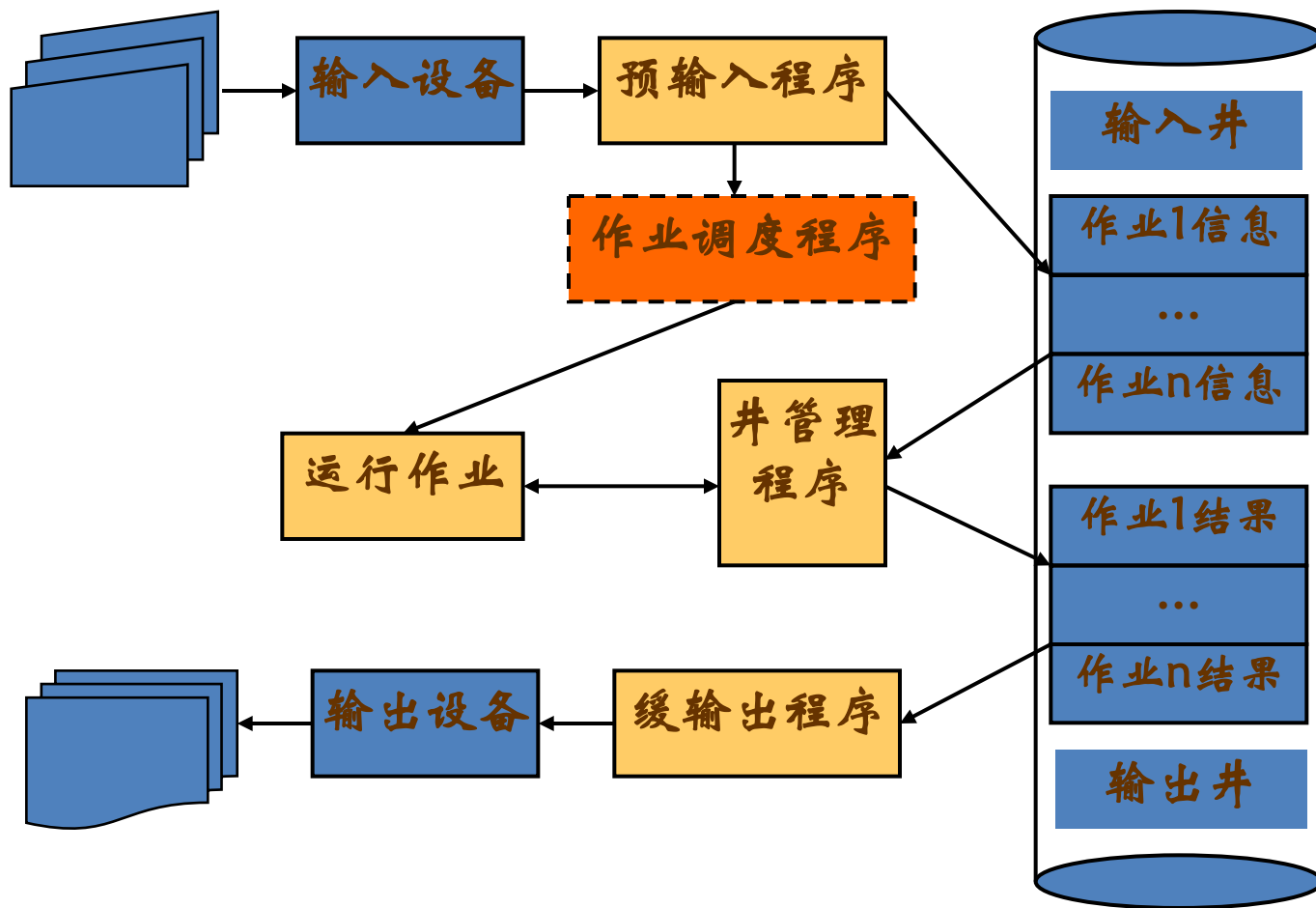
(1) 问题的提出

- 静态分配方式是不利于提高系统效率
- 采用脱机外围设备操作
- 联机同时外围设备操作(又称作假脱机操作)

5.7.2 斯普林系统的设计和实现(1)

- “井”是用作缓冲的存储区域，采用井的技术能调节供求之间的矛盾，消除人工干预带来的损失。
- “预输入程序”
- “缓输出程序”
- “井管理程序”

SPOOLING组成和结构



输入井中作业状态

- 输入状态:
- 收容状态:
- 执行状态:
- 完成状态:

SPOOLING数据结构

- 作业表 登记进入系统的所有作业的作业名、状态、预输入表位置等信息。
- 预输入表 每个用户作业有一张用来登记该作业的各个文件的情况，包括设备类、信息长度及存放位置等。
- 缓输出表 每个用户作业拥有一张包括作业名、作业状态、文件名、设备类、数据起始位置、数据当前位置等。

井文件空间的管理

- 第一种是**连接方式**，输入的信息被组织成连接文件，这种方式优点是数据信息可以不连续存放，文件空间利用率高。

井文件空间的管理

- 第二种是**计算方式**，假定磁盘井文件空间，每个磁道存放100个80字节记录，每张卡片为80个字节，若每个柱面有20个磁道，则一个柱面可存放2000张卡片信息。第n张卡片信息被存放在：

$$\text{磁道号} = \text{卡片号}n / 100$$

$$\text{记录号} = (\text{卡片号}n) \bmod 100$$

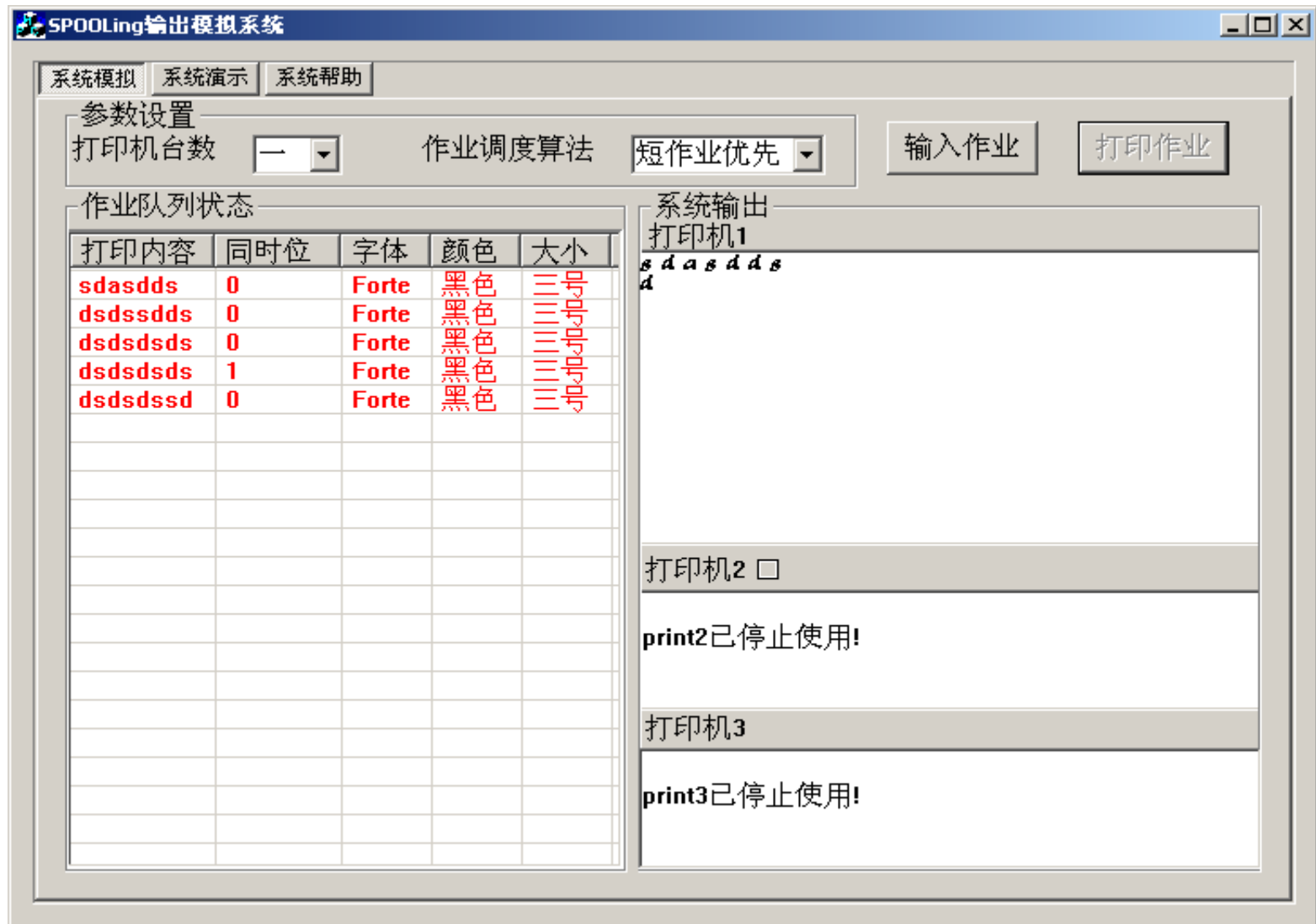
- 用卡片号n除以100的整数和余数部分分别为其存放的磁道号和记录号。

Spooling应用例子

(1) 打印机spooling守护进程

(2) 网络通信spooling守护进程

Spooling的模拟



打印任务	属主	字体大小	打印内容
------	----	------	------

属主

字体大小

打印内容

输入进程

删除进程

暂停进程

[使用帮助](#)

打印进程信息

当前打印任务:

sdsdd

字体大小:

10

打印输出:dsadsdfdsfdfsdf

dsadsd fds fds

当前打印任务的进度:

输入井信息

打印任务

块数

输出井信息

打印任务

块数

sdsdd

16

2

8

2

9

1

6

操作系统的设备管理应具备的主要功能是__、__、__和

——^o监视设备状态、进行设备分配、完成I/O操作、缓冲管理与地址转换

在UNIX系统中， 所有的I/O设备按其物理特性分为字符设备和块设备。

SPOOLing技术可以实现设

备的（ C ）分配。

A. 独占

B. 共享

C. 虚拟

D. 物理

使用用户所编制的程序与实际使用的物理设备无关，这是由设备管理的（ A ）功能实现的。

- A. 设备独立性**
- B. 设备分配**
- C. 缓冲管理**
- D. 虚拟设备**

判断对错并改正

虚拟

利用共享分配技术可以提高设备的利用率，使得打印机之类的独占设备成为可共享的、快速I/O设备。（ × ）