西安电子科技大学
XIDIAN UNIVERSITY

计算机科学与技术学院
SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY
国家示范性软件学院
NATIONAL PILOT SCHOOL OF SOFTWARE ENGINEERING

操作系统原理

第二章 作业管理和用户接口

主讲：黄伯虎

引言

本章内容

- 主要讲解操作系统服务接口和批处理系统中作业调度的基本策略。通过对本章内容的学习，理解操作系统提供服务的机制，同时了解一些经典的调度算法。
- 本章我们首先讲解用户接口，然后讲解作业管理。

一、用户接口

一、用户接口类型

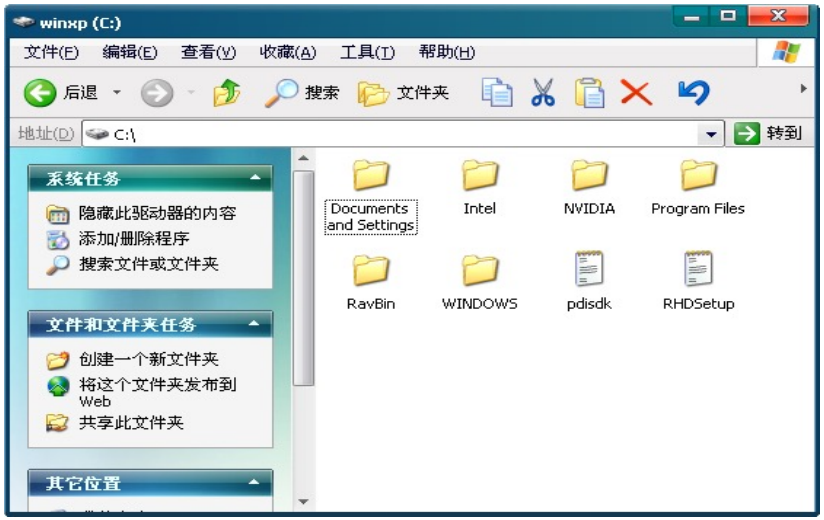
作业控制级接口——面向人

- 命令驱动
- 图形化驱动（视窗）

UNIX Shell伪代码：

```
while (true) {
type_prompt (); //显示命令提示符
read_command (command, parameters) //从用户获得命令
if (fork () ==0) {//fork一个子进程
execve (command, parameters, 0); //子进程代码，执行用户命令
}
else {//父进程代码段
waitpid (-1, &status, 0); //等待子进程结束
}
}
```

命令方式

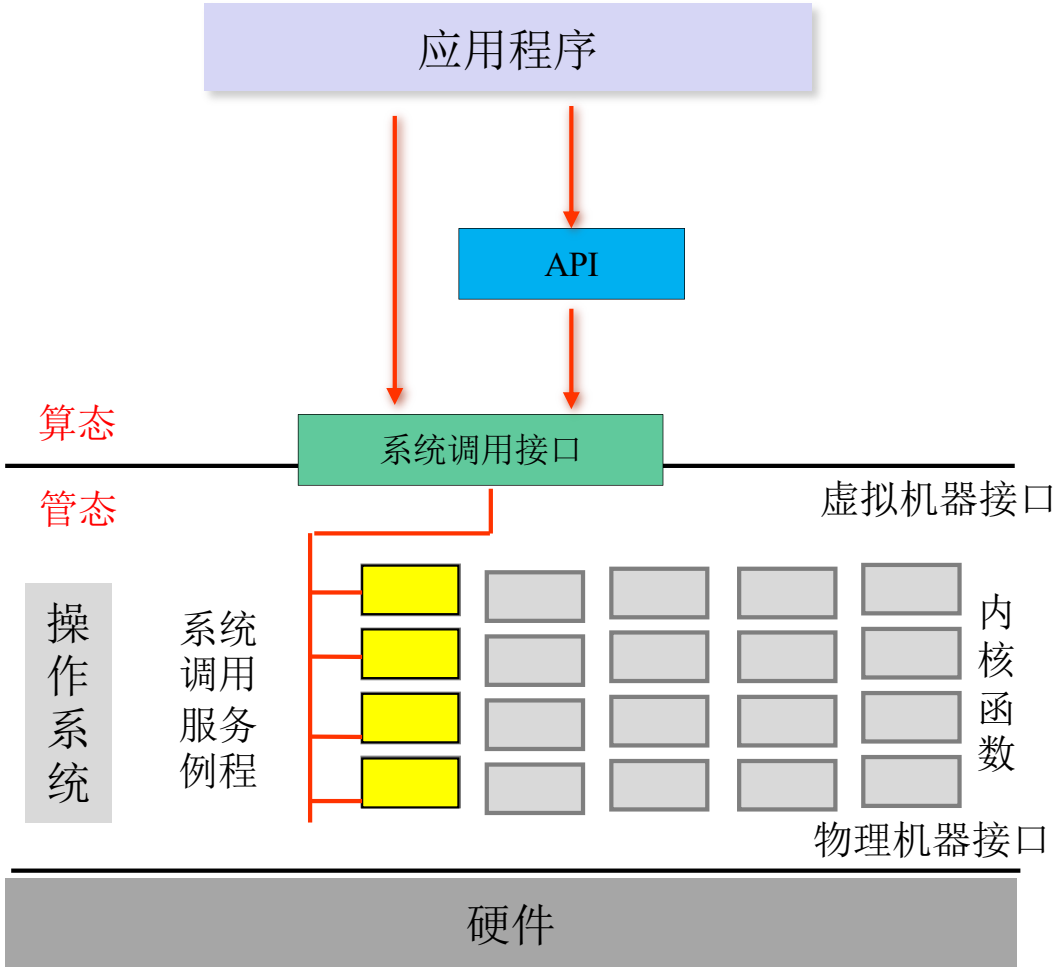


图形方式

一、用户接口类型

程序级接口——面向应用程序

❖ 系统功能调用(System Call)



1.程序的管态和算态

- 操作系统运行的状态称为管态(系统态，核心态)
- 用户程序运行的状态称为算态(用户态，目态)

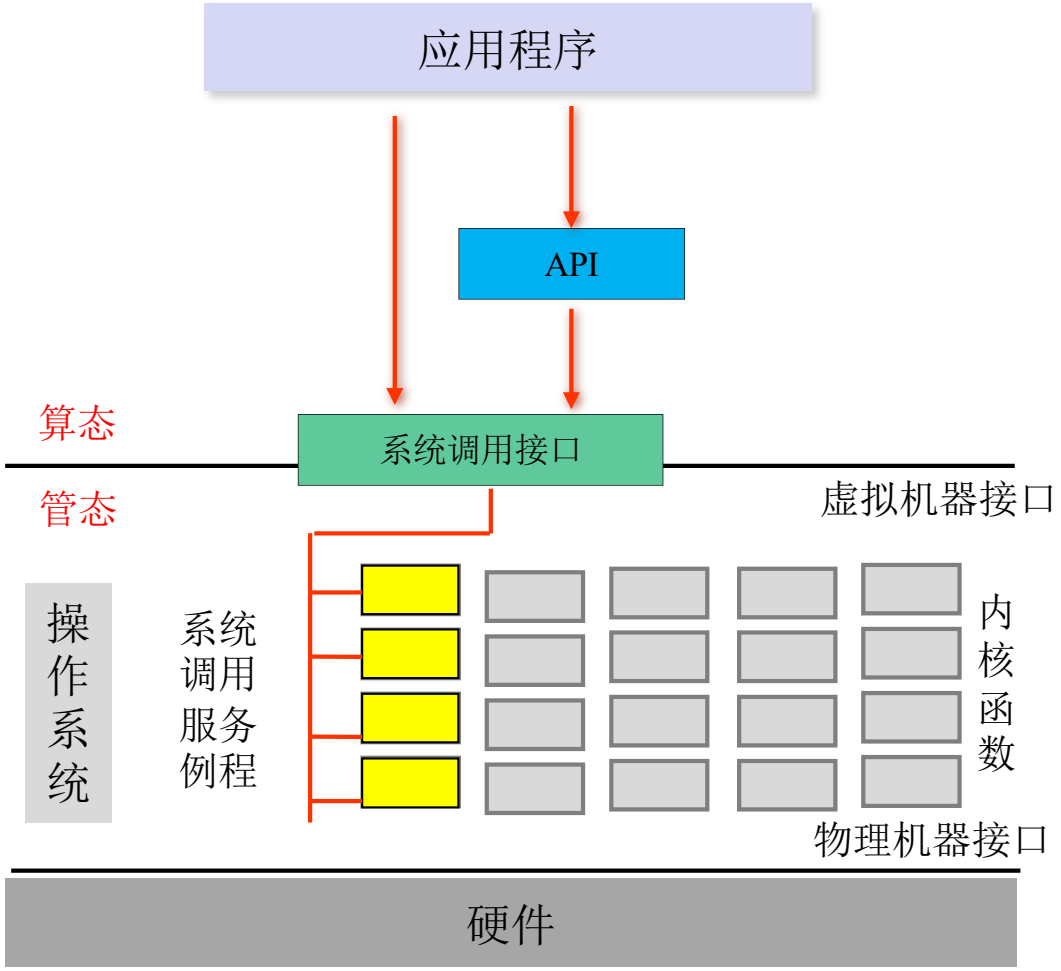
2.特权指令

- 只能在管态下运行而不能在算态下执行的一类特殊指令。这类指令通常和硬件操作相关，如I/O指令、存取中断寄存器、时钟寄存器、地址寄存器指令，CPU清内存指令等等。

二、系统功能调用

程序级接口——面向应用程序

❖ 系统功能调用(System Call)

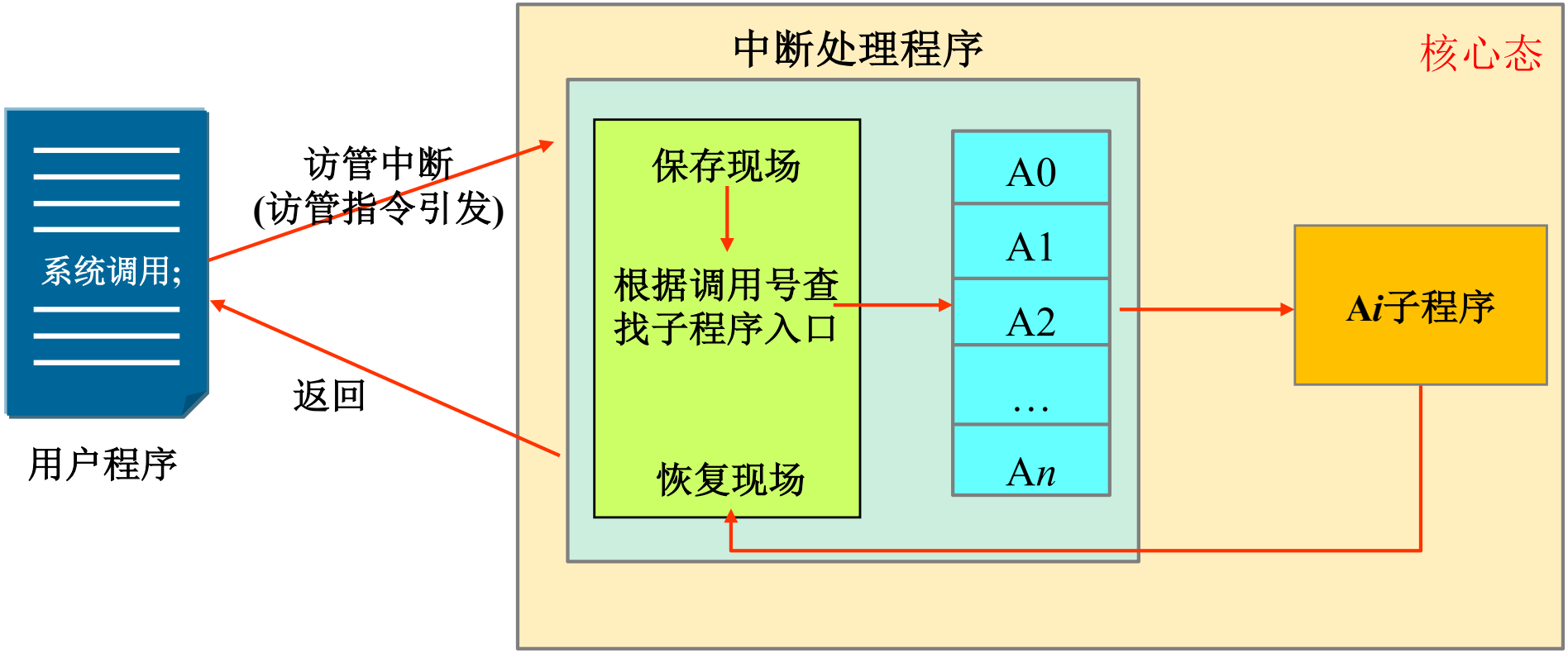


系统功能调用类型（6大类）：

- 进程控制类
- 文件管理类
- 设备管理类
- 内存管理类
- 信息维护类
- 通信类

二、系统功能调用

系统功能调用过程



访管指令：也叫陷阱指令、自陷指令，是在算态下执行的一条特殊的指令，可引发中断，使得程序由算态切换到管态（自愿进管）。

二、系统功能调用

系统功能调用实例：getpid()

用户程序

```
#include <stdio.h>
#include <sys/types.h>
int main(void) {
long ID;
ID = getpid(); //获取进程ID号
printf ("getpid()=%ld\n", ID);
return(0);
}
```

算态

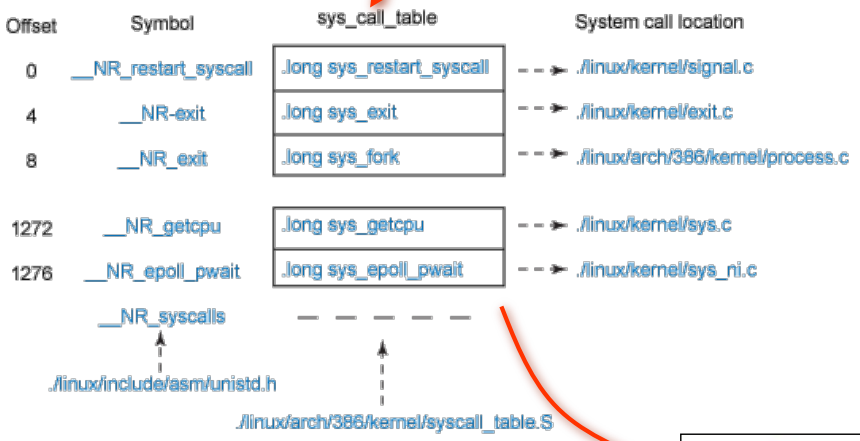
管态

system_call

保存现场;
读取eax寄存器的值__NR_getpid;
读取数组 sys_call_table[__NR_getpid]的值;
执行内核函数sys_getpid();
复制eax寄存器值到用户态eax寄存器栈单元;
恢复现场;

库函数

```
getpid()
{
往eax寄存器存入__NR_getpid;
int 0x80 产生软中断;
-------系统功能调用执行------
从堆栈里面读取ID;
}
```



读取用户进程数据结构task_struct的成员变量pid，把用户进程的pid放在eax寄存器中。

```cpp
long sys_getpid(void)
{
        return current->tgid;
}
```

二、系统功能调用

系统功能调用与普通过程调用区别

❖ 运行状态不同

- 系统功能调用的调用过程和被调用过程运行在不同的状态，而普通的过程调用均运行在相同的状态（算态）。系统功能调用开销较大，普通过程调用开销较小。

❖ 调用方法不同

- 系统调用必须通过软中断机制进入系统核心，然后才能执行相应的处理程序；普通过程调用则可以直接调用。

❖ 返回问题

- 普通的过程调用完成后直接返回调用过程继续执行。系统功能调用在返回时需要重新进行调度。

二、作业管理（批处理系统）

一、作业的基本概念

什么是作业？

- 作业(Job)是用户在一次算题过程中或者一个事务处理过程中要求计算机系统所做工作的总和。简单来讲，我们可以将作业看作是用户要求计算机处理的一项任务(task)。
- 作业具体体现形式为程序，一个作业可能对应一个或多个程序。
- 作业的概念多出现于批处理系统和大型、巨型机系统；PC机和工作站中一般不使用这个词。

一、作业的基本概念

作业的分类

脱机作业：

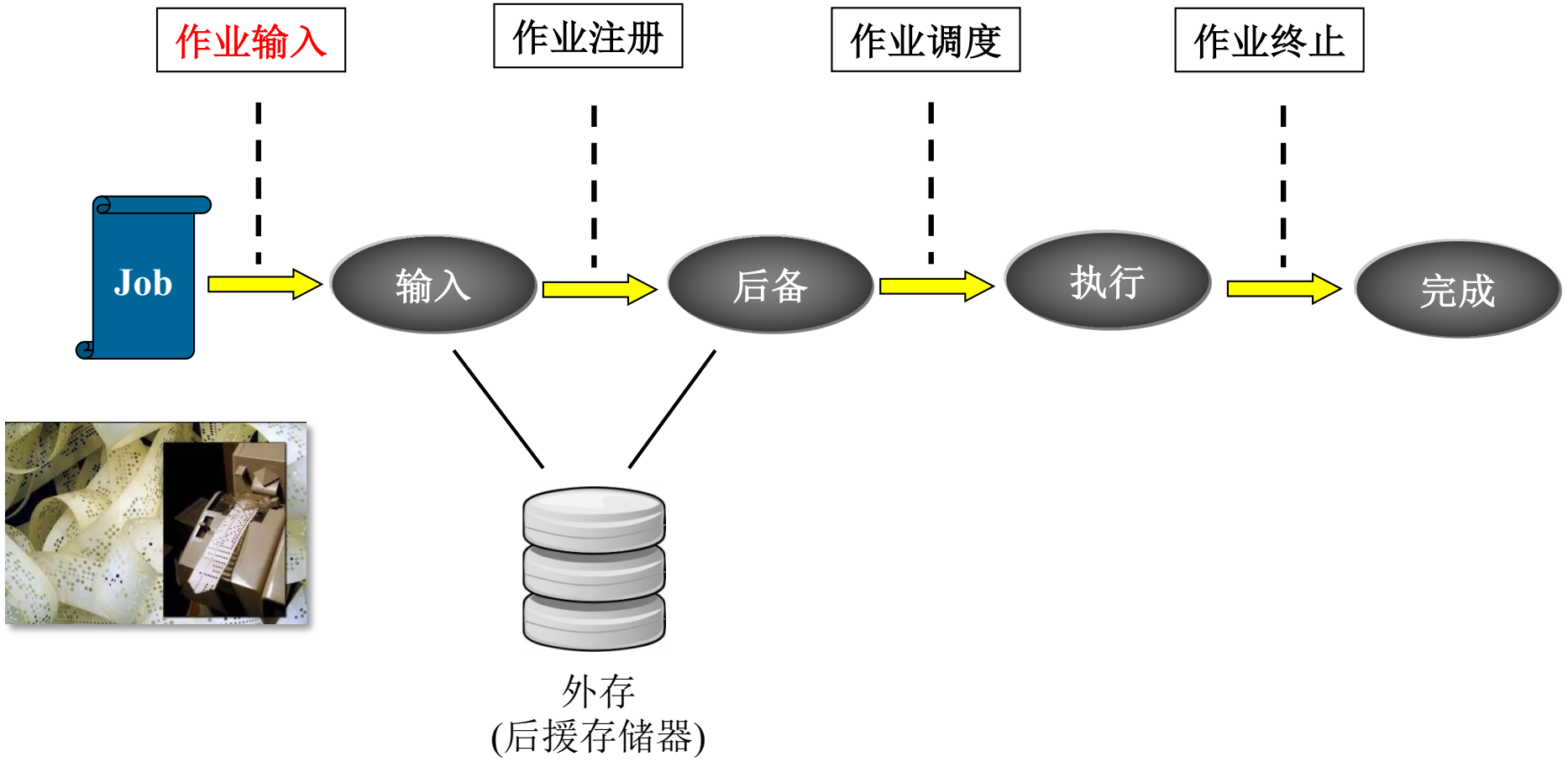
- 特点：用户不直接和计算机系统交互，其执行过程由操作员辅助完成。
- 常在批处理系统使用，也称为批量型作业/后台作业。

联机作业：

- 特点：用户在作业执行过程中可直接和计算机系统交互（人机对话），控制执行的过程。
- 现在的系统中多为联机作业，也称为交互型作业/终端作业/前台作业。

一、作业的基本概念

作业的处理过程（批处理系统中）



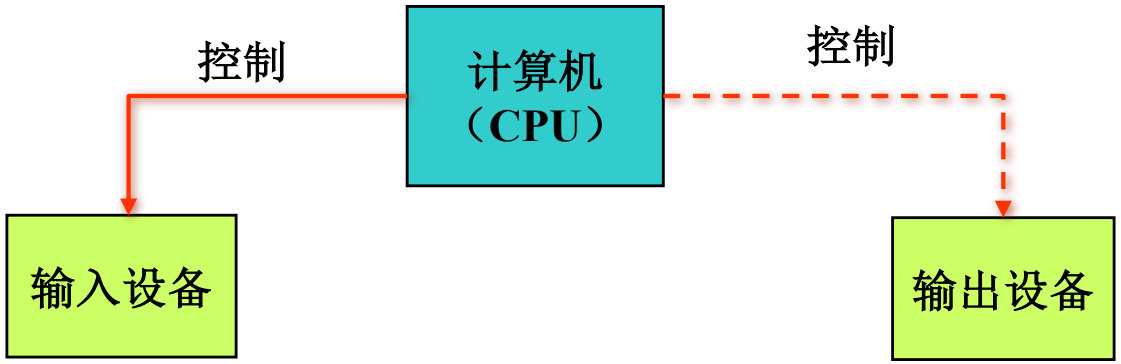
二、作业的输入/输出方式

脱机输入/输出(人工干预)

问题：效率低

优点：主机和输入/输出机可以并行工作。

联机输入/输出方式



问题：如何控制才能最大限度提高效率？

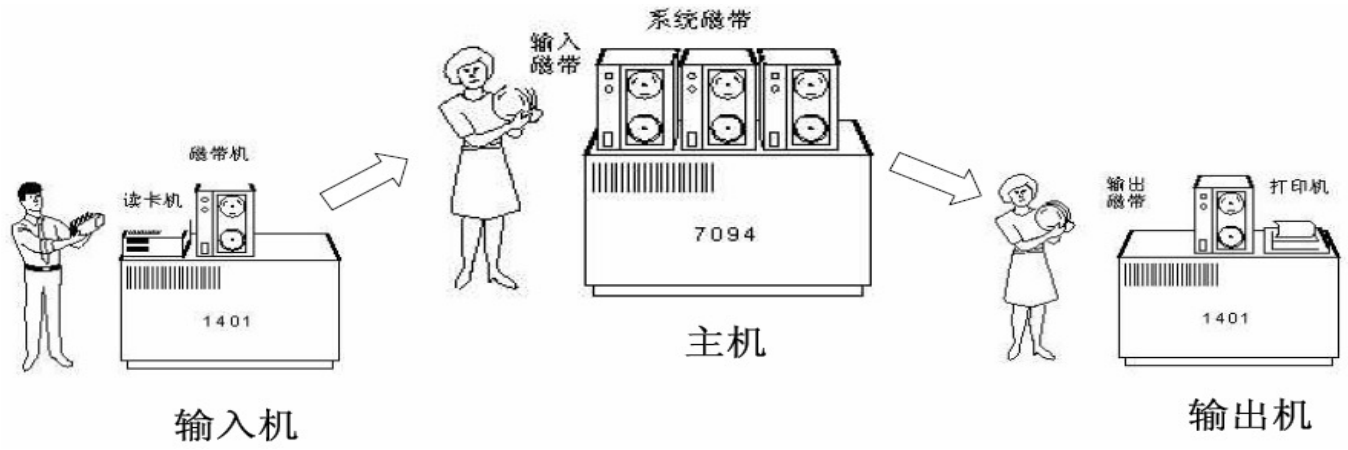
二、作业的输入/输出方式

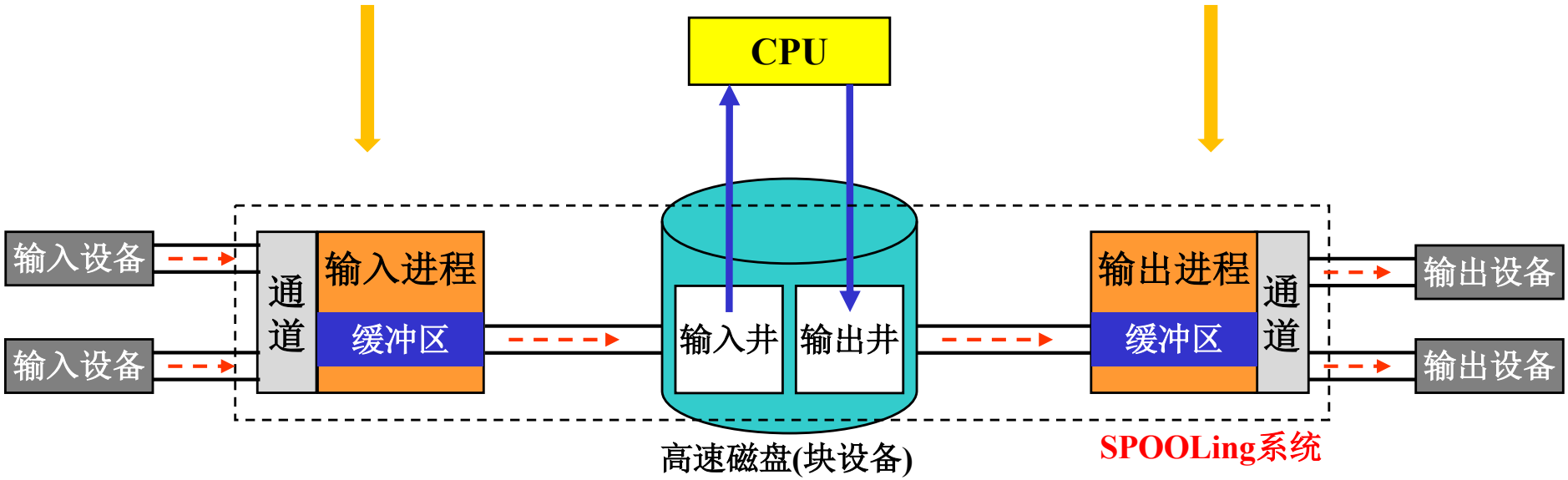
SPOOLing系统

- 外围设备同时联机操作(Simultaneous Peripheral Operation On-Line)。
- 特点：兼具脱机和联机方式的优点，可以实现联机方式下的主机和外围设备的同时工作，又称为假脱机，也即以联机的方式得到脱机的效果。
- 主要技术基础：
  - 多道程序设计技术（并发）
  - 通道技术

二、作业的输入/输出方式

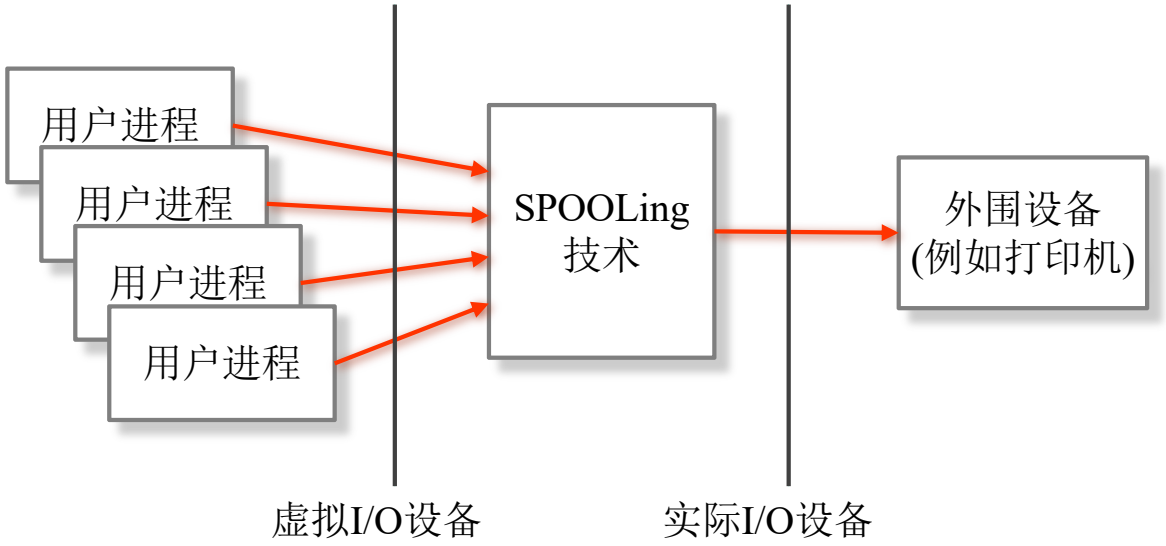
❖ SPOOLing系统原理





二、作业的输入/输出方式

❖ SPOOLing实例——打印机共享



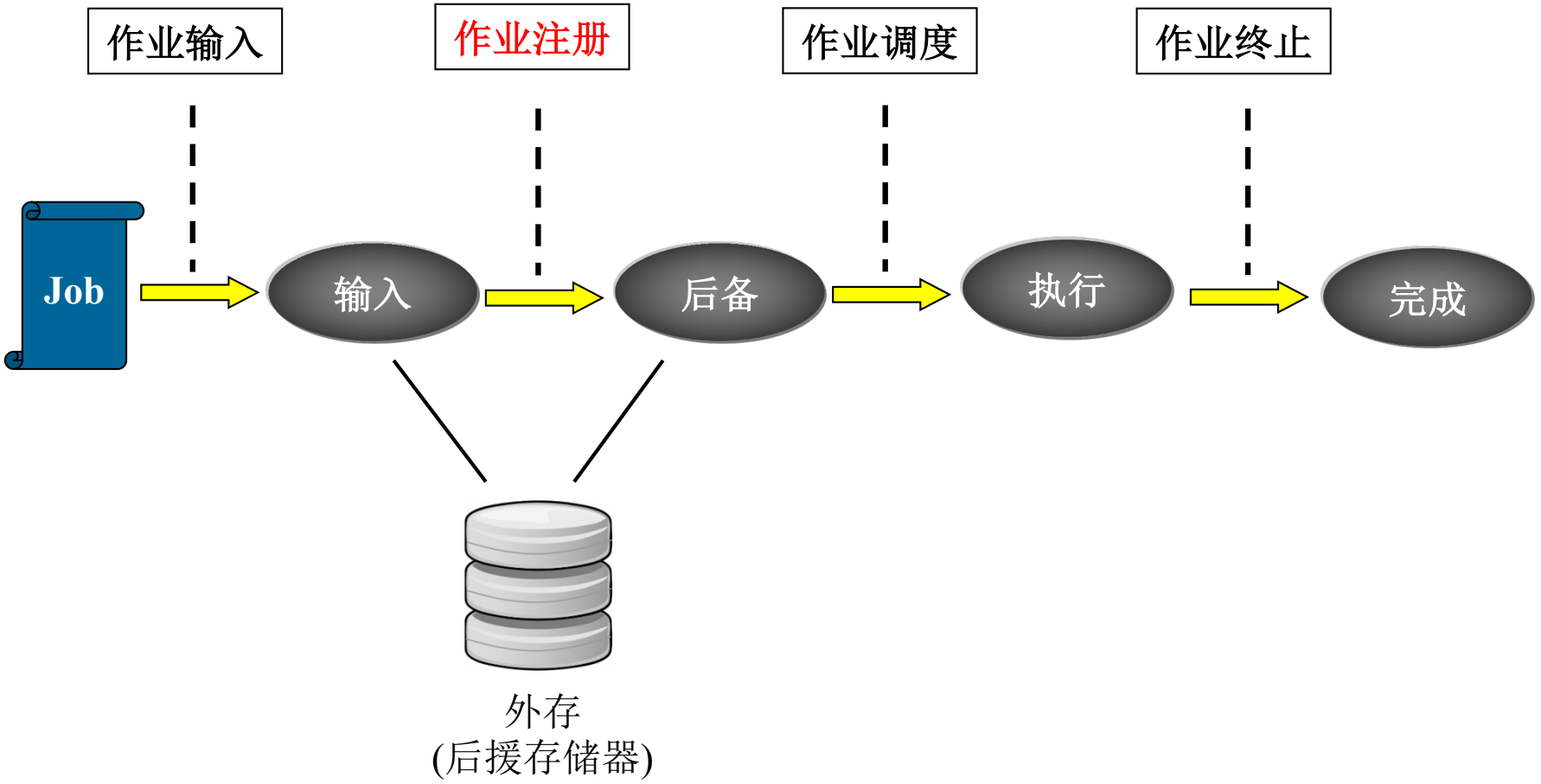
SPOOLing系统特点：

- 实现了I/O设备的虚拟化。将一台物理I/O设备虚拟为多台逻辑I/O设备。
- 提高了系统效率。减少了CPU与外设之间速度差异的矛盾，提高了I/O设备和CPU并行工作程度。

具体过程：当用户进程请求打印输出时，SPOOLing系统立即同意为该进程执行打印输出，但并不真正把打印机分配给该用户进程，而是由输出进程先在输出井中申请一个空闲盘块区，并将要打印的数据送入其中；然后为用户申请并填写请求打印表，将该表挂到请求打印队列上。若打印机空闲，输出进程从请求打印队列队首取出一张打印表，将要打印的数据从输出井传送到内存缓冲区，再进行打印，直到打印队列为空。

三、作业注册

作业的处理过程(批处理系统)



三、作业注册

作业控制块(JCB, Job Control Block)

JCB是系统为管理作业设置的一个数据结构，是系统中作业存在的唯一标志。

JCB里面记录有与作业相关的各种信息，只有当作业退出系统时，JCB才被撤销。

JCB内容：

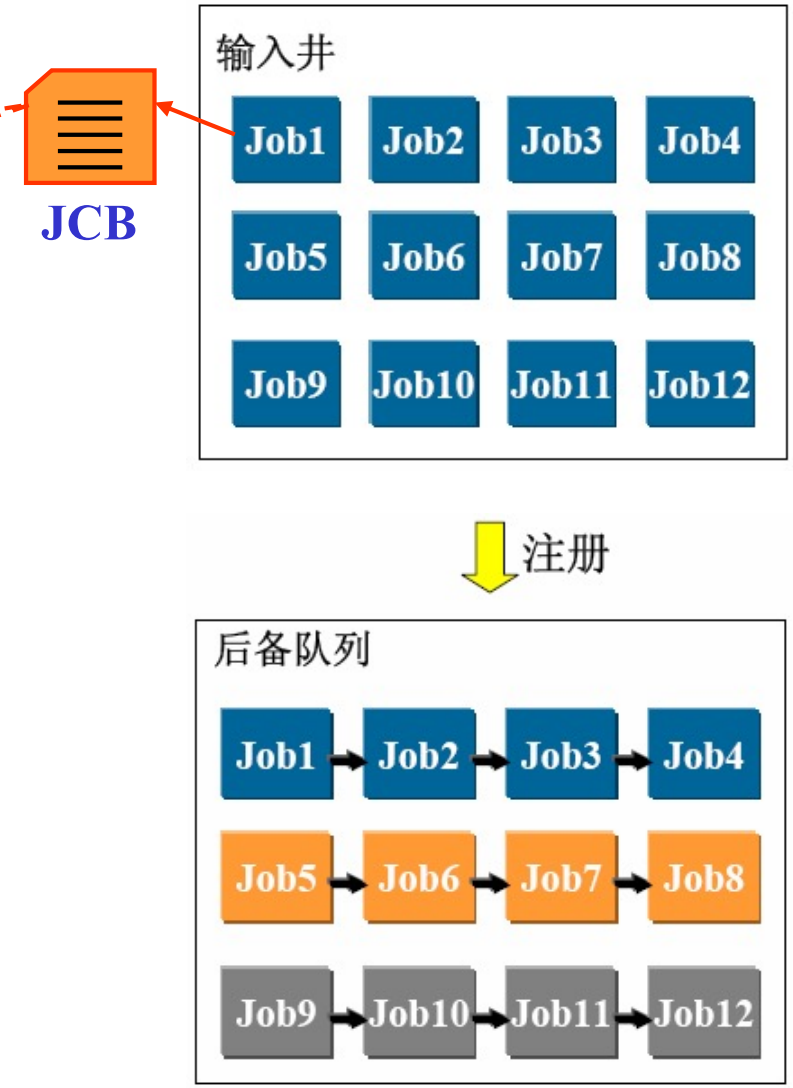
标识信息：作业名，用户名，用户帐号；

状态信息：提交、后备、执行、就绪、等待、完成；

调度参数：优先级；

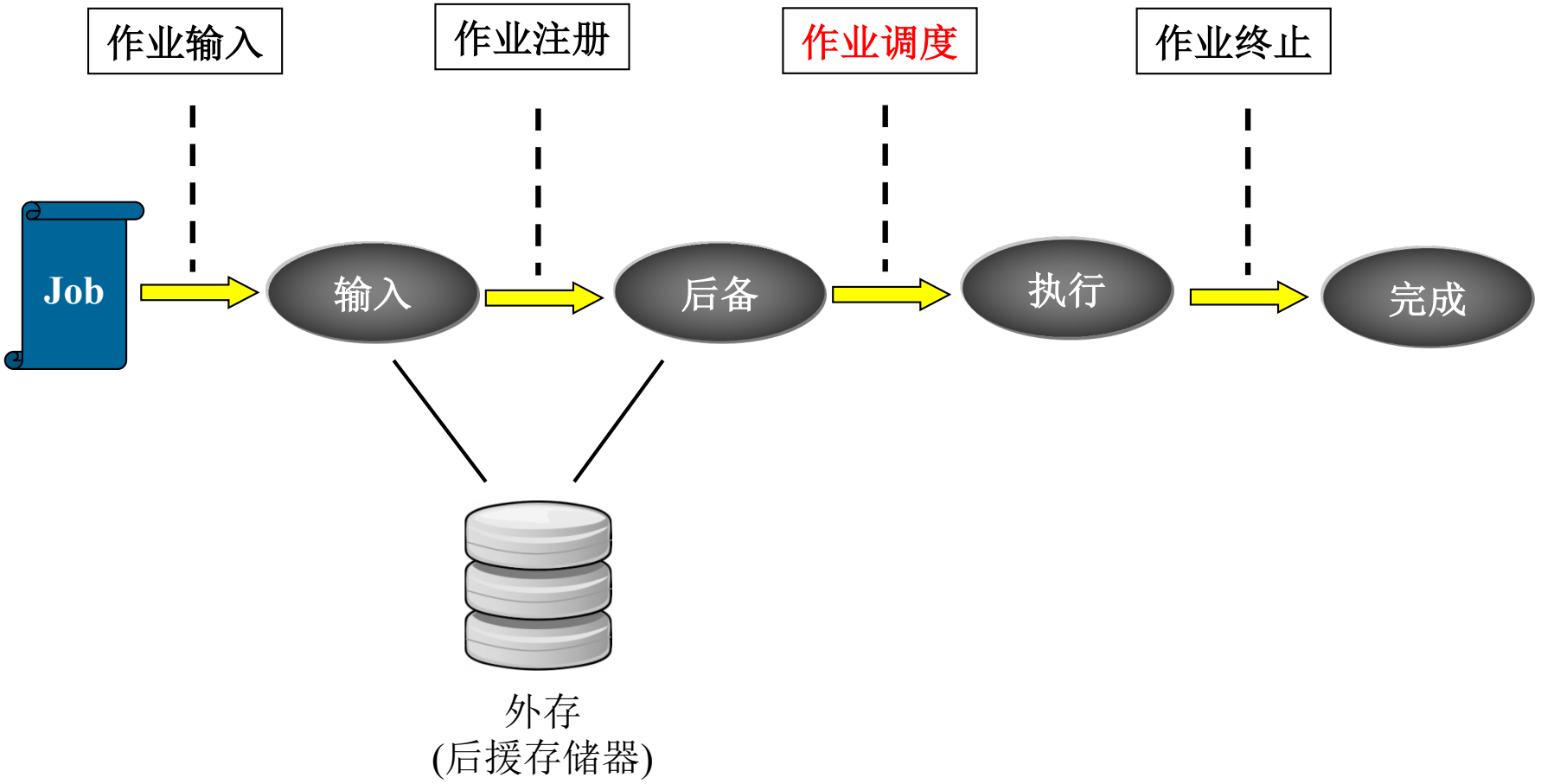
资源需求量；

……



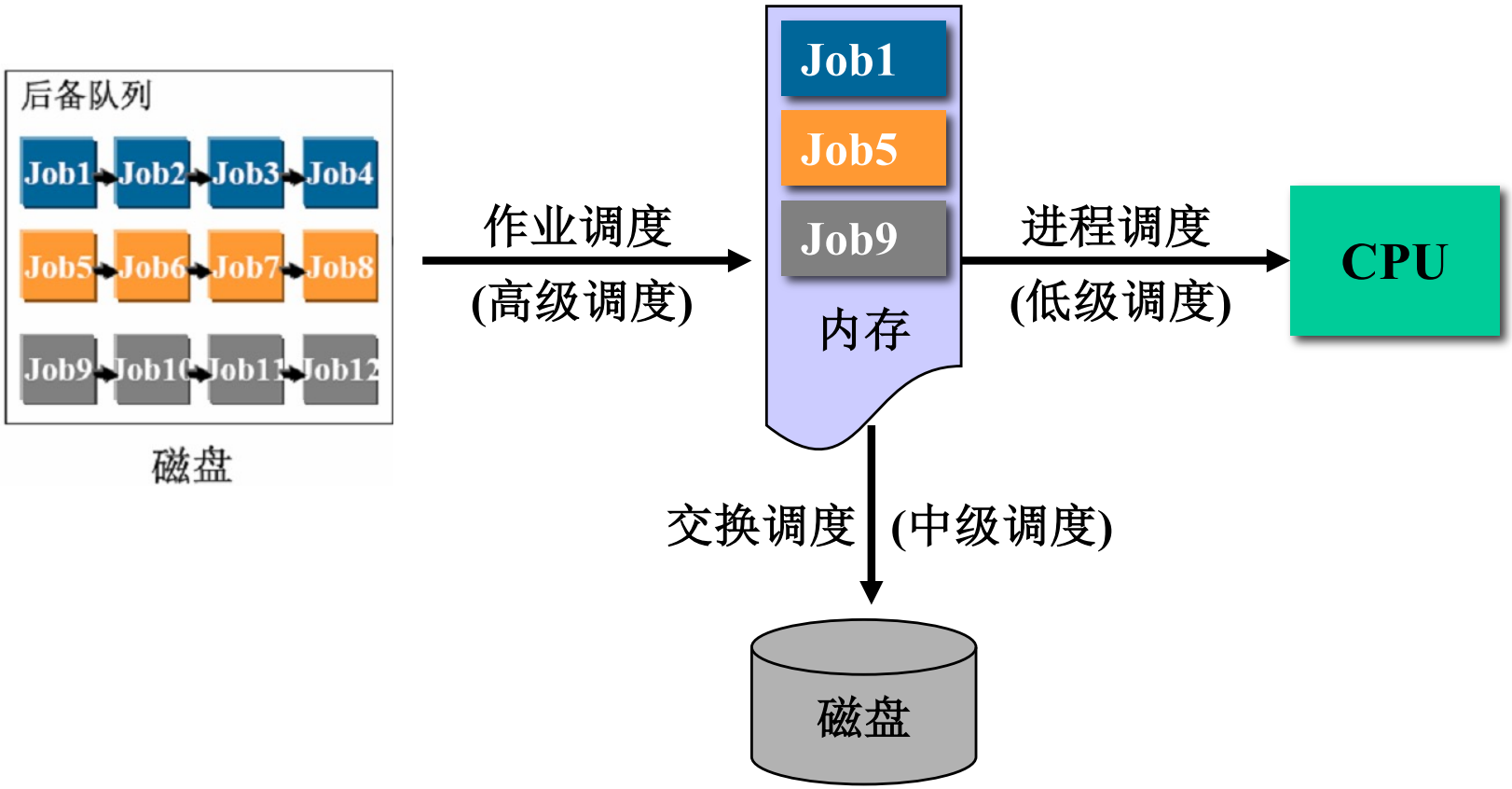
四、作业调度

作业的处理过程(批处理系统)



四、作业调度

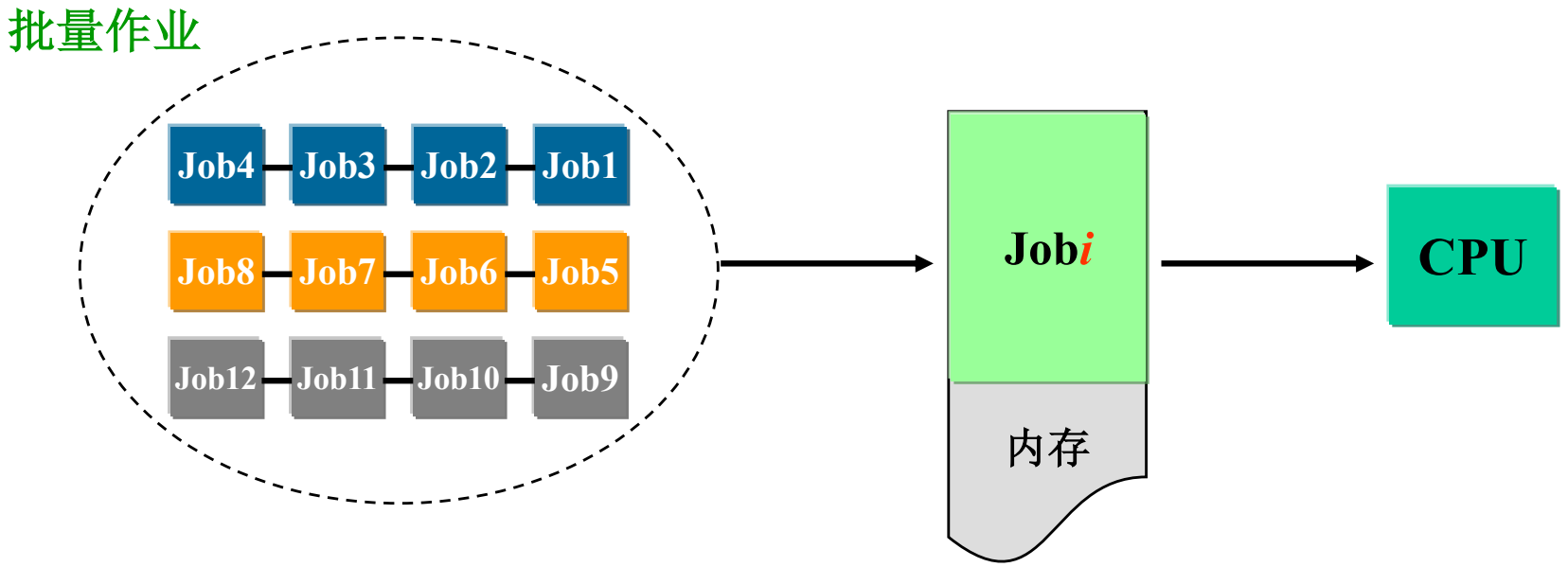
作业的执行过程



四、作业调度

单道批处理系统作业调度

- 单道批量处理系统特点：一次只能将一个作业调入内存运行(没有并发)。



单道：每次内存只能容纳下一道作业。

单道批处理系统作业调度算法

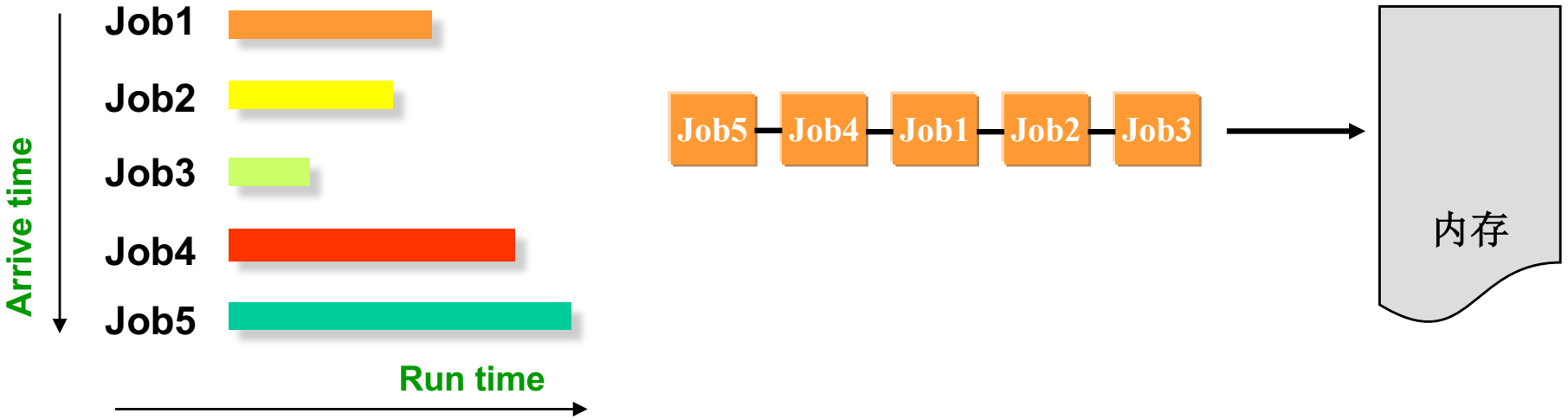
❖ 先来先服务调度算法(FCFS, First Come First Served)

优点：简单，公平

缺点：周转时间长，吞吐量小，不利于短作业的执行

四、作业调度

❖ 最短作业优先调度算法(SJF, Shortest Job First)



优点：降低作业平均周转时间，提高系统吞吐量

缺点：对长作业不利，出现“饥饿”现象

四、作业调度

❖ 最高响应比优先调度算法(HRP, Highest Ratio Priority)

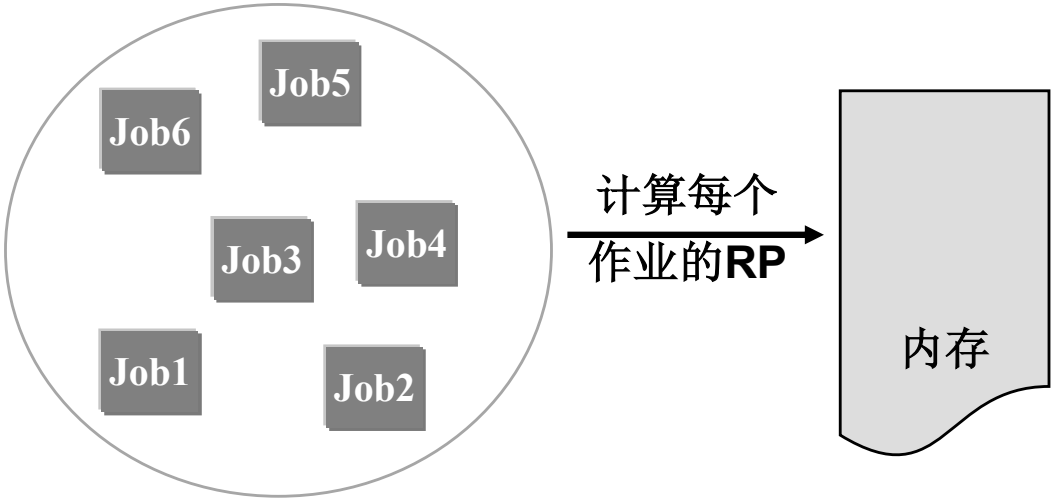
➢ 思想：为每个作业设置一个响应比(优先级)，使之能够尽量保证短作业优先执行，同时又不让长作业等待过长时间。

响应比计算方法：

RP ＝ 作业响应时间 / 作业运行时间

＝ (作业等待时间+ 作业运行时间) / 作业运行时间

＝ 1 + (作业等待时间 / 作业运行时间)



特点：既照顾到了短作业，也考虑到了长作业，能够克服长作业的“饥饿”现象，是一种折中的算法。

调度算法例：

系统在10:00同时发现有两个作业在后备队列中等待调度，其相关信息如下：

①作业1：9:00到达，需要运行60分钟；

②作业2：9:50到达，需要运行20分钟；

问：分别按照FCFS, SJF, HRP算法进行调度，那个作业会被优先选择？

答：（1） FCFS: 作业1优选被选择；

（2） SJF: 作业2优选被选择；

（3） HRP: 作业1：RP = 1 + 60min / 60min = 2

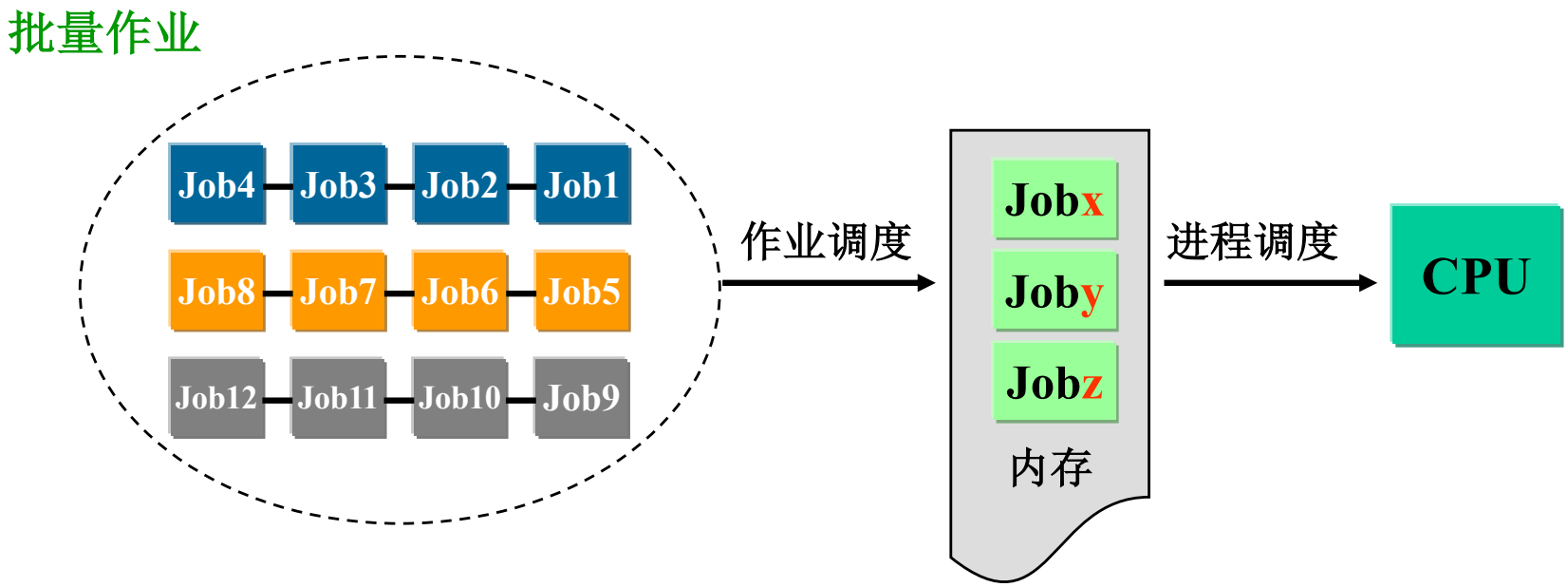
作业2：RP = 1 + 10min / 20min = 1.5

作业1优选被选择。

四、作业调度

多道批处理系统作业调度算法

- 多道批处理系统的特点：一次可以选择多个作业同时执行(并发)。



五、作业调度算法性能分析

评价准则

- CPU利用率。希望能够获得较高的CPU利用率。显然尽量让外设和CPU同时工作是提高CPU利用率的有效方法。
- 吞吐量：单位时间内CPU完成作业的数量。显然短作业优先有利于提高吞吐量。
- 周转时间：评价批处理系统的性能指标。

设作业J_i的提交时间为t_{si}，执行时间为t_{ri}，完成时间为t_{oi}，则：

① 作业J_i的周转时间 $T_i = t_{oi} - t_{si}$

② 周转系数 $W_i = T_i / t_{ri}$

③ 平均周转时间：

$$T = \frac{(T_1 + T_2 + \ldots + T_n)}{n} = \frac{1}{n}\sum_{i=1}^{n} T_i$$

④ 平均周转系数：

$$W = \frac{(W_1 + W_2 + \ldots + W_n)}{n} = \frac{1}{n}\sum_{i=1}^{n} W_i$$

五、作业调度算法性能分析

单道程序环境下作业调度算法的性能

- 例：假设有4个作业，提交时刻、执行时间分别如下：

作业 i	提交时刻 t_{si}	执行时间(h) t_{ri}
1	8.00(8:00)	2.00
2	8.50(8:30)	0.50
3	9.00(9:00)	0.10
4	9.50(9:30)	0.20

问：若分别使用FCFS、SJF、HRP算法进行调度，试分析其性能？

五、作业调度算法性能分析

❖ 先来先服务调度算法(FCFS)

作业 i	提交时刻 t_{si}	执行时间(h) t_{ri}	开始时刻	完成时刻 t_{oi}	周转时间 T_i	周转系数 W_i
1	8.00	2.00	8.00	10.00	2.00	1.00
2	8.50	0.50	10.00	10.50	2.00	4.00
3	9.00	0.10	10.50	10.60	1.60	16.00
4	9.50	0.20	10.60	10.80	1.30	6.50
平均周转时间T=1.725 h 平均周转系数W=6.875					6.90	27.50

五、作业调度算法性能分析

❖ 最短作业优先调度算法(SJF)

作业 i	提交时刻 t_{si}	执行时间(h) t_{ri}	开始时刻	完成时刻 t_{oi}	周转时间 T_i	周转系数 W_i
1	8.00	2.00	8.00	10.00	2.00	1.00
2	8.50	0.50	10.30	10.80	2.30	4.60
3	9.00	0.10	10.00	10.10	1.10	11.00
4	9.50	0.20	10.10	10.30	0.80	4.00
		平均周转时间T=1.55 h 平均周转系数W=5.15			6.20	20.60

五、作业调度算法性能分析

❖ 响应比高者优先算法

作业 i	提交时刻 t_{si}	执行时间(h) t_{ri}	开始时刻	完成时刻 t_{oi}	周转时间 T_i	周转系数 W_i
1	8.00	2.00	8.00	10.00	2.00	1.00
2	8.50	0.50	10.10	10.60	2.10	4.20
3	9.00	0.10	10.00	10.10	1.10	11.00
4	9.50	0.20	10.60	10.80	1.30	6.50
		平均周转时间T=1.625 h 平均周转系数W=5.675			6.50	22.72

作业 i	提交时刻 t_{si}	执行时间(h) t_{ri}	开始时刻	完成时刻 t_{oi}	周转时间 T_i	周转系数 W_i
1	8.00	2.00	8.00	10.00	2.00	1.00
2	8.50	0.50	10.00	10.50	2.00	4.00
3	9.00	0.10	10.50	10.60	1.60	16.00
4	9.50	0.20	10:60	10.80	1.30	6.50
		平均周转时间T=1.725 h 平均周转系数W=6.875			6.90	27.50

作业 i	提交时刻 t_{si}	执行时间(h) t_{ri}	开始时刻	完成时刻 t_{oi}	周转时间 T_i	周转系数 W_i
1	8.00	2.00	8.00	10.00	2.00	1.00
2	8.50	0.50	10.30	10.80	2.30	4.60
3	9.00	0.10	10.00	10.10	1.10	11.00
4	9.50	0.20	10.10	10.30	0.80	4.00
		平均周转时间T=1.55 h 平均周转系数W=5.15			6.20	20.60

作业 i	提交时刻 t_{si}	执行时间(h) t_{ri}	开始时刻	完成时刻 t_{oi}	周转时间 T_i	周转系数 W_i
1	8.00	2.00	8.00	10.00	2.00	1.00
2	8.50	0.50	10.10	10.60	2.10	4.20
3	9.00	0.10	10.00	10.10	1.10	11.00
4	9.50	0.20	10.60	10.80	1.30	6.50
		平均周转时间T=1.625 h 平均周转系数W=5.675			6.50	22.72

结论：就平均周转时间和平均周转系数来说，最短作业优先算法最小，先来先服务算法最大，响应比高者优先算法居中。

五、作业调度算法性能分析

多道程序环境下作业调度的性能

- 例：假设一个两道作业批处理系统，现有4个作业，提交时刻、执行时间分别如下。作业进入内存时采用短作业优先调度算法。作业占有处理器采用抢占式优先级调度算法，规定短作业享有较高优先级。

  请给出作业周转时间分析？

作业 i	提交时刻 t_{si}	执行时间(min) t_{ri}
1	10:00	30
2	10:05	20
3	10:10	20
4	10:20	10

五、作业调度算法性能分析

作业 i	提交时刻 t_{si}	执行时间 (min)t_{ri}	开始时刻	完成时刻 t_{oi}	周转时间 T_i	周转系数 W_i
1	10:00	30	10:00	11:20	80	2.667
2	10:05	20	10:05	10:25	20	1
3	10:10	20	10:35	10:55	45	2.25
4	10:20	10	10:25	10:35	15	1.5
平均周转时间T=140 min 平均周转系数W=1.854					160	7.417