

## 第一章

### 1. OS 作为用户与计算机硬件系统之间的接口：

含义是：OS 处于用户和计算机硬件系统之间，用户通过 OS 来使用计算机系统。

用户可以通过以下三种方式使用计算机：命令方式；系统调用方式；图形、窗口方式

### 2. 操作系统的发展过程：

无操作系统的计算机系统、单道批处理系统、多道批处理系统、分时系统、实时系统  
多道批处理系统是操作系统成熟的标志。

### 3. 操作系统的定义：

操作系统是一组控制和管理计算机硬件和软件资源，合理地各类作业进行调度，以及方便用户使用的程序的集合。

### 4. 分时系统--（1）人机交互的特征是边运行边调试。（2）共享主机（3）便于用户上机

### 5. 实时系统的及时性：及时响应外部事件请求，在规定的时间内完成对该事件的处理，控制所有实时任务协调一致运行。

### 6. 分时系统的特征：

(1)多路性 即同时性，宏观上同时，微观上轮流 (2)独占性 每个用户感觉独占主机  
(3)及时性 较短时间响应(1-3 秒) (4)交互性

### 7. 实时系统与分时系统特征的比较：

分时系统是指在一台主机上连接多个带有显示器和键盘的终端，同时允许多个用户通过自己的终端，以交互方式使用计算机，共享主机中的资源。

实时系统（Real Time System）是指系统能及时（或即时）响应外部事件的请求，在规定的时间内完成对该事件的处理，并控制所有实时任务协调一致地运行。

实时系统的特点：多路性、独占性、及时性、交互性、可靠性，主要是及时性。

分时系统的特征：多路性、独占性、及时性、交互性，其中最主要的就是交互性。

### 8. 操作系统的基本特征：并发性（最重要特征）、共享性、虚拟性、异步性

### 9. 并行与并发：并行性是指多个事件在同一时刻同时发生；并发性是指两个或多个事件在同一时间间隔内发生，宏观上在同一时间段内同时运行，微观上交替执行。

### 10. 共享：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，相应地把这种资源共同使用成为资源共享或称为资源复用。

### 11. 临界资源：在一段时间内只允许一个进程访问的资源称为临界资源或独占资源。

### 12. 并发和共享是操作系统的二个最基本特征，他们又是互为存在的条件

### 13. 虚拟技术：操作系统中的所谓“虚拟”（Virtual），是指通过某种技术把一个物理实体变为若干个逻辑上的对应物。物理实体是实的，即实际存在的，后者是虚的，是用户感觉上的东西。用于实现虚拟的技术称为虚拟技术。

### 14. 操作系统的主要功能：

处理机管理功能：对处理机进行分配——进程管理和调度

存储器管理功能：对内存进行分配、保护和扩充

设备管理功能：缓冲管理、设备分配、设备处理

文件管理功能：文件存储空间的管理、目录管理、文件的读写管理和保护

操作系统与用户之间的接口 用户接口和程序接口

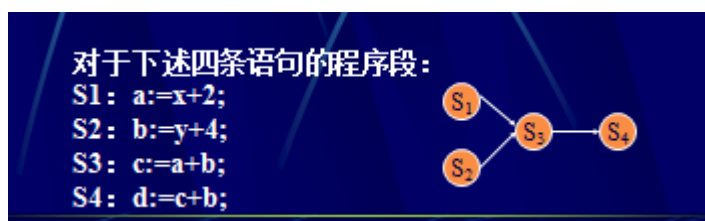
## 第二章 进程管理

### 0.程序顺序执行的特征：顺序性、封闭性、可再现性

#### 1. 前趋图(Precedence Graph)：

一个有向无循环图、描述程序或程序段之间执行的前后关系

前趋图是一个有向无循环图。（必须不存在循环） 根据程序画前趋图：



2. 并发执行时的特征：间断性——“停停走走”；  
失去封闭性——原因：多个程序共享资源；不可再现性

### 3. 进程的定义和特征：

**定义：**进程是进程实体的运行过程，是系统进行资源分配和调度的一个独立单位。进程是程序的一次执行，进程是一个程序及数据在处理机上顺序执行时所发生的活动。

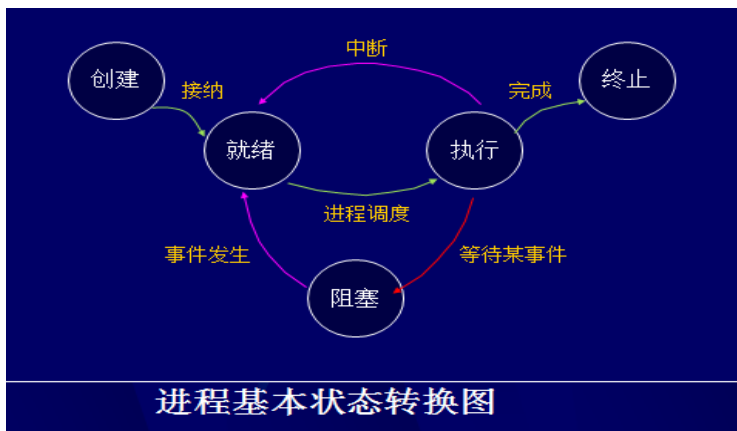
**进程的特征：**1) **结构性：**进程由程序段、数据段及 PCB 三部分组成，在 Linux 中称为“进程映像”2) **动态性：**“它由创建而产生，由调度而执行，由撤销而消亡”。是进程的最基本特征3) **独立性：**进程是一个能独立运行、独立分配资源和独立调度的基本单位。各进程的地址空间相互独立。4) **并发性：**引入进程的目的正是为了使其程序能和其他进程的程序并发执行；5) **异步性：**进程按各自独立的、不可预知的速度向前推进

#### 进程的三种基本状态

**就绪状态 (Ready)：**得到了除 CPU 以外的所有必要资源。

**执行状态 (Running)：**已获得处理机，程序正在被执行。

**阻塞状态 (Blocked)：**因等待某事件发生而暂时无法继续执行，从而放弃处理机，使程序执行处于暂停状态。



4. **进程控制块 PCB (Process Control Block)：**是进程实体的一部分，是操作系统中最重要记录性数据结构。PCB 中记录了操作系统所需的、用于描述进程的当前情况以及控制进程运行的全部信息。进程控制块的作用是使一个在多道程序环境下不能独立运行的程序成为一个能独立运行的基本单位，一个能与其他进程并发执行的进程。PCB 是进程存在的惟一标志。进程控制块中的信息：1) 进程标示符 2) 处理机状态—通用寄存器、指令计数器、程序状态字 PSW、用户栈指针 3) 进程调度信息 4) 进程控制信息
5. **临界区：**人们在每一个进程中访问临界资源的那段代码称为临界区
6. **同步机制应遵循的规则：**
  - (1) 空闲让进。当无进程处于临界区时，表明临界资源处于空闲状态，应允许一个请求进入临界区的进程立即进入自己的临界区，以有效地利用临界资源。
  - (2) 忙则等待。当已有进程进入临界区时，表明临界资源正在被访问，因而其他试

图进入临界区的进程必须等待，以保证对临界资源的互斥访问。

- (3) 有限等待。对要求访问临界资源的进程，应保证在有限时间内能进入自己的临界区，以免陷入“死等”状态。
- (4) 让权等待。当进程不能进入自己的临界区时，应立即释放处理机，以免陷入“忙等”状态。
7. **Wait (S) 操作描述：**执行一次 wait 操作意味着请求分配一个单位的资源，因此描述为： $s.value = s.value - 1$ 。减 1 后：若  $s.value \geq 0$ ，则进程继续进行；若  $s.value < 0$ ，表示已无资源可用，因此请求该资源的进程将被阻塞，要把它排在信号量  $s$  的等待队列中，此时， $s.value$  的绝对值等于该信号量等待队列上的进程数目。
8. **Signal(S)操作描述：**执行一次 signal 操作意味着释放一个单位的资源，故  $s.value = s.value + 1$ 。加 1 后：若  $s.value > 0$ ，则进程继续；若  $s.value \leq 0$ ，表示信号量请求队列中仍有因请求该资源而被阻塞的进程，因此应把队列中的一个或几个进程唤醒，使之转至就绪队列中。
9. **进程通信：**是指进程之间的信息交换。  
**进程通信的类型即高级通信机制：**共享存储器系统、消息传递系统、管道通信系统三种。

### 第三章 处理机调度与死锁

1. **处理机调度的层次：**高级调度、低级调度、中级调度
2. **高级调度 (High Level Scheduling)：**  
又叫作业调度或长程调度 (LongTerm Scheduling)，其主要功能是根据某种算法，把外存上处于后备队列中的哪些作业调入内存。也就是说它的调度对象是作业。
3. **低级调度 (Low Level Scheduling)：**通常也称为进程调度或短程调度 (ShortTerm Scheduling)，它所调度的对象是进程（或内核级线程）。决定就绪队列中的哪个进程应获得处理机，然后再由分派程序执行把处理机分配给该进程的具体操作。进程调度是最基本的调度，在三种类型的 OS 中都必须配置
4. **低级调度的功能：**  
保存处理机的现场信息；按照某种算法选取进程；把处理机分配给进程。
5. **进程调度方式**
  - a) 非抢占方式
  - b) 抢占方式
6. **先来先服务调度算法：**是一种最基本的调度算法，既可用于作业调度也可用于进程调度。比较有利于长作业进程，而不利于短作业进程。
6. **短作业优先调度算法：**指对短作业或段进程优先调度算法可以分别用于作业调度和进程调度。该算法对长作业不利，完全未考虑作业的紧迫程度，不能保证紧迫性作业及时处理。该算法不一定真正做到短作业优先调度。
7. **死锁 (Deadlock) 定义：**  
死锁是指两个或两个以上的进程在运行过程中，因争夺资源而造成的一种互相等待（谁也无法再继续推进）的现象，若无外力作用，它们都将无法推进下去。
8. **产生死锁的原因：**
  - 1、竞争资源
  - 2、进程间推进顺序非法
9. **产生死锁的必要条件：**
  - 1、**互斥条件。**一个资源一次只能被一个进程使用。
  - 2、**请求和保持条件（部分分配）。**保留已经得到的资源，还要求其它的资源。
  - 3、**不可剥夺条件（不可抢占）。**资源只能被占有者释放，不能被其它进程强行抢占。
  - 4、**环路等待条件（循环等待）。**系统中的进程形成了环形的资源请求链。

**预防死锁的方法** 1. 摒弃请求和保持条件 2. 摒弃不剥夺条件 3. 摒弃环路等待条件

#### 10. 安全状态：

允许进程动态的申请资源，但在分配前，应先计算分配的安全性。

所谓“安全状态”：指系统能按某种进程顺序( $P_1, P_2, \dots, P_n$ )，来为每个进程  $P_i$  分配其所需资源，直至最大需求，使每个进程都可以顺利完成。反之，则系统处于不安全状态。

不安全状态不一定发生死锁，但死锁一定属于不安全状态。

#### 11. 安全状态之例：

安全状态				转化	不安全状态			
进程	最大需求	已分配	系统可用		进程	最大需求	已分配	系统可用
P1	10	5	3		P1	10	5	2
P2	4	2			P2	4	2	
P3	9	2			P3	9	3	
系统资源总数：12				系统资源总数：12				

### 第四章 存储器管理

#### 1. 程序的装入和链接：

如何将一个用户源程序变为一个可在内存中执行的程序，通常都要经过以下几个步骤：**首先是编译**，由编译程序将用户源代码编译成若干个目标模块；**其次是链接**，由链接程序将编译后形成的一组目标模块，以及他们所需要的库函数链接在一起，形成一个完整的装入模块；**最后是装入**，由装入程序将模块装入内存。

#### 2. 重定位：通常是把在装入时对目标程序中指令和数据的修改过程称为重定位。

**静态重定位：**因为地址变换通常是在装入时一次完成的，以后不再改变。

#### 3. 动态分区分配：根据进程的实际需要，动态的分配内存空间。

#### 4. 分区分配算法：

**首次适应算法：**空闲分区按起址递增次序排列，从头开始直至找到第一个满足要求的空闲分区。 **特点：**内存低端会留下小的空闲区，高端有大的空闲区；

**循环首次应算法：**从上次分配的位置之后开始查找。

**特点：**使内存的空闲分区均匀，但缺乏大的空闲分区；

**最佳适应算法：**空闲分区按大小递增的次序排列，从头开始找到第一个满足要求的空闲分区。 **缺点：**会留下大量小碎片。

**最坏适应算法：**空闲分区按大小递减的次序排列，最前面的最大的空闲分区就是找到的分区。 **优点：**分配后剩下的可用空间比较大。 **缺点：**一段时间后就不能满足对于较大空闲区的分配要求。

#### 5. 页面和物理块：分页存储管理是将一个进程的逻辑地址控件分成若干个大小相等的片，称为页面或页并为各页加以编号。相应的把内存空间分成与页面相同大小的若干个存储块，称为物理块或页框，也对它们加以编号。

#### 6. 页表的作用：页表的作用是实现在页号到物理块号的地址映射。

#### 7. 地址变换机构的基本任务：实现从逻辑地址到物理地址的转换，借助于页表完成的。

8. 分页地址结构由页号  $P$  和位移量  $W$  组成  
例子：系统的页面大小为 1KB，设  $A=2017B$ ，可以求出页号  $P=2$ ，页内地址/位移量  $d=122$ 。
9. **段表的作用**：段表是用于实现从逻辑段到物理内存区的映射。定义：在系统中为每个进程建立一张段映射表。
10. **地址变换机构**是为了实现从进程的逻辑地址到物理地址的变换功能。
11. **分页和分段的主要区别**：  
**相似之处**：两者都采用离散分配方式且通过地址映射机构来实现地址变换。  
**不同之处**：（1）页是信息的物理单位，段是信息的逻辑单位；  
（2）页的大小固定，段的大小动态变化；  
（3）分页系统中的逻辑地址空间是一维的，分段系统中的是二维的。
12. **虚拟存储器的定义**：虚拟存储器是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。
13. **虚拟存储器的特征**：  
**多次性**。一个作业被分成多次调入内存运行；  
**对换性**。允许在作业的运行过程中进行换进、换出；  
**虚拟性**。能从逻辑上扩充内存容量，使用户“看到”的内存容量远大于实际大小。  
该特征是以上两个特征为基础的。
14. **页面置换算法**：最佳置换算法（未来最长时间不再被访问的页面淘汰。OPT）  
先进先出（FIFO 最早进入页面的淘汰）  
最近最久未使用的置换算法（LRU 最近最久未使用的页面淘汰）  
一个好的页面置换算法，应具有较低的页面更换频率

## 第五章 设备管理

1. **I/O 设备分类**：  
**按设备使用特性分类**：第一类存储设备，第二类输入/输出设备  
**按传输速率分类**：按传输书读的高低，可分为三类：  
**低速设备**，典型设备有键盘、鼠标器、语音的输入输出等设备。  
**中速设备**，典型设备有行式打印机、激光打印机等。  
**高速设备**，典型的设备有磁带机、磁盘机、光盘机等。  
**按信息交换单位分类**：第一类是块设备（Block Device），用于存储信息。例如磁盘  
第二类是字符设备（Character Device），用于数据的输入和输出。交互式终端、打印机
2. **I/O 通道**：I/O 通道是一种特殊的处理机，它具有执行 I/O 指令的能力，并通过执行通道（I/O）来控制 I/O 操作。
3. **I/O 控制方式**：**程序 I/O 方式**，数据传输的基本单位是字节  
**中断驱动 I/O 方式**，数据传输的基本单位仍是字节  
**DMA 控制方式**，以多个块为单位进行数据传送；数据传输的基本单位是数据库  
**I/O 通道控制方式**，以多个块为单位进行数据传送；一次传送多组数据到多个不同的内存区域。
4. **缓冲技术分为**：单缓冲，双缓冲，循环缓冲、缓冲池。  
**单缓冲和双缓冲**：  
**单缓冲**：在设备和处理机之间设置一个缓冲区。T 和 C 是可以并行的。系统对每个数据的处理时间为  $\max(C, T) + M$ 。  
**双缓冲—缓冲对换**：  
系统处理每个数据的时间可粗略认为  $\max(C, T)$ 。当  $T > C$ ，可使块设备连续输入；反

之可使 CPU 不必等待设备输入。**目的：**加快输入输出的速度。

**循环缓冲：**循环缓冲是把多个缓冲区连接起来组成两部分，一部分专门用于输入，另一部分专门用于输出的缓冲结构。

5. **设备驱动程序：**设备驱动程序通常又称为设备处理程序，它是 I/O 进程与设备控制器之间的通信程序，又由于它常以进程的形式存在，故简称为设备驱动程序。

6. **设备分配中的数据结构：**

**设备控制表 DCT，**系统为每台设备配置一张

**控制器控制表 COCT，**系统为每一个控制器都设置了一张控制器控制表。

**通道控制表 CHCT，**每个通道都配有一张通道控制表。

**系统设备表 SDT，**记录了系统中全部设备的情况，每个设备占一个表目。

7. **基本的设备分配程序：**

按下述步骤进行设备分配：分配设备 → 分配控制器 → 分配通道

8. **磁盘访问时间：**

**寻道时间  $T_s$ ：**(可优化处理)

把磁臂（磁头）移动到指定磁道上所经历的时间，包含启动磁臂和磁头移动  $n$  条磁道所花费的时间。是优化的基础。

**旋转延迟时间  $T_r$ ：**

指定扇区移动到磁头下面所经历的时间。与盘面的旋转速度有关。

5400 转—平均旋转延迟时间 5.55ms；7200 转—平均旋转延迟时间 4.16ms

**传输时间  $T_t$ ：**

把数据从磁盘读出或向磁盘写入数据所经历的时间。与旋转速度和一次读写的数据量有关

10. **磁盘调度：**

**先来先服务 FCFS：**根据进程请求访问磁盘的先后次序进行调度。

优点：公平、简单，每个进程的请求依次得到处理

缺点：平均寻道时间可能较长，仅适用于磁盘请求较少的场合。

有九个进程先后提出  
磁盘I/O请求：  
55, 58, 39, 18, 90,  
160, 150, 38, 184

(从100号磁道开始)	
被访问的下一个磁道号	移动距离 (磁道数)
55	45
58	3
39	19
18	21
90	72
160	70
150	10
38	112
184	146
平均寻道长度：55.3	

**最短寻道时间优先 SSTF：**选择要求访问的磁道与当前磁头所在的磁道距离最近的进程（磁盘请求），使每次的寻道时间最短。该算法不能保证平均寻道时间最短。可能导致“饥饿”现象。

有九个进程先后提出  
磁盘I/O请求：  
55, 58, 39, 18, 90,  
160, 150, 38, 184

(从100号磁道开始)	
被访问的下一个磁道号	移动距离 (磁道数)
90	10
58	32
55	3
39	16
38	1
18	20
150	132
160	10
184	24
平均寻道长度: 27.5	

**扫描（Scan）算法：**又称为“电梯调度算法”。磁头每次只作单方向移动，直到到达边缘磁道为止，然后再作反向移动。下一次待访问的磁道只能在此磁头移动的前方，且选择磁头移动距离最近的一个磁盘请求响应。消除了饥饿现象。

**循环扫描（CScan）算法：**磁头只作由内向外的单方向扫描，到达外边缘后，则返回最内侧的磁道重新进行下一轮扫描。改进了对于边缘区磁道访问的不公平。

## 第六章 文件管理

1. **记录**是一组相关数据项的集合，用于描述一个对象的某些属性。

**关键字：**能够唯一标识一个记录的数据项

2. **按文件的性质和用途分：**

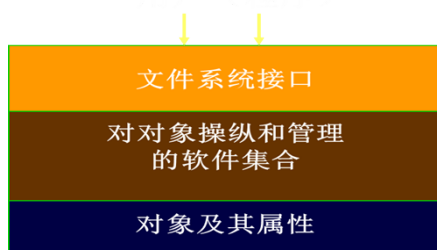
系统文件：由系统软件构成的文件，只允许调用执行，不允许用户读和修改。

用户文件：只允许文件的授权者使用。

库文件：允许用户调用不允许修改。

3. **文件系统模型：**

用户（程序）



文件系统模型

**对象及其属性：**文件管理系统管理的对象有：

文件：文件管理的直接对象；目录：方便用户对文件的存取和检索；

磁盘（磁带）存储空间

4. **文件的逻辑结构：**这是从用户观点出发所观察到的文件组织形式，是用户可以直接处理的数据及其结构，它独立于文件的物理特性，又称为文件组织。
5. **文件的物理结构，**又称为文件的存储结构，是指文件在外存上的存储组织形式。
6. **文件逻辑结构的类型：**文件的逻辑结构可分为两大类：

一类是有结构文件，这是指一个以上的记录构成的文件，故又把它称为记录式文件。



二是无结构文件，这是指由字符流构成的文件，故又称为流式文件。

7. **外存分配方式：**常用的外存分配方法有连续分配、链接分配和索引分配三种。

8. **链接分配：**将文件装到多个离散的盘块中，是离散的分配方式。

**链接方式又可分为：**隐式链接、显式链接两种

9. **隐式链接：**

在文件的每个目录项中，都含有指向链接文件第一盘块和最后一个盘块的指针。每个盘块中都有指向下一个盘块的指针。 特点：只适合于顺序访问，随机访问效率极低。

10. **显式链接：**

把用于链接文件各物理块的指针，显式地存放在内存的一张“链接表”中。该表在整个磁盘只设置一张。即文件分配表（FAT）。序号为盘块号  $0..n-1$



11. **目录管理的要求：**实现“按名存取”——是目录管理的最基本的功能，也是文件系统向用户提供的最基本的服务；**提高对目录的检索速度；文件共享；允许文件重名。**

12. 为了能对一个文件进行正确的存取，必须为文件设置用于描述和控制文件的数据结构，称为“**文件控制块 FCB**”。文件管理程序可借助于文件控制块中的信息，对文件施以各种操作。文件与文件控制块一一对应，而人们把文件控制块的有序集合称为**文件目录**，即一个文件控制块就是一个文件目录项。一个文件目录页被看做是一个文件，称为**目录文件**。

13. **文件存储空间的管理：**分配方式：连续分配、离散分配

**存储空间的基本分配单位**是以磁盘块（扇区）为单位，而非字节。

14. **文件存储空间的管理方法**

1) **空闲表法：**连续分配方式，为外存上的所有空闲区建立一张空闲表。

2) **空闲链表法：**离散分配方式，根据构成链所用基本元素不同分为以下两种形式：

**空闲盘块链：**将磁盘上的所有空闲空间，以盘块为单位拉成一条链。

**空闲盘区链：**将磁盘上所有空闲盘区（每个盘区可包含若干个盘块）拉成一条链。

3) **位示图法：****位示图：**利用二进制的一位来表示磁盘中一个盘块的使用情况。由所有盘块所对应的位构成一个集合，称为位示图。用  $m \times n$  个位数构成位示图。

4) **成组链接法**

15. **常用的两种文件共享方法：**1) 基于索引结点的共享方式；2) 利用符号链实现文件共享  
**基于索引结点的共享方式：**文件目录中只设置文件名及指向相应索引结点的指针；文件的物理地址及其它的文件属性等信息只存放在索引结点中；



程序:

1、利用 AND 信号量机制解决哲学家进餐问题:

```

Var chopstick array of semaphore:=(1,1,1,1,1);
Processi
Repeat
    think;
    Swait(chopstick[(i+1)mod 5],chopstick[i]);
    eat;
    Ssignat(chopstick[(i+1)mod 5],chopstick[i]);
until false;

```

2、利用记录型信号量解决生产者---消费者问题

```

Var mutex, empty, full:semaphore:=1,n,0;
buffer:array[0, ..., n-1] of item;
in,out:integer:=0,0;
begin
    parbegin
        proceducer: begin
            repeat
                .
                .
                producer an item nextp;
                .
                .
                wait(empty);
                wait(mutex);
                buffer(in):=nextp;
                in:=(in+1) mod n;
                signal(mutex);
                signal(full);
            until false;
        end
        consumer: begin
            repeat
                wait(full);
                wait(mutex);
                nextc:=buffer(out);
                out:=(out+1) mod n;
                signal(mutex);
                signal(empty);
                consumer the item in nextc;
            until false;
        end
    parend
end

```

3、利用信号量实现进程互斥:

```

Var mutex:semaphore:=1;
begin
    parbegin
        process 1:begin
            repeat
                wait(mutex);
                critical section
                signal(mutex);

```

```

        remainder section
    until false;
end
process 2:begin
    repeat
        wait(mutex);
        critical section
        signal(mutex);
        remainder section
    until false;
end

```

算法：

调度算法：周转时间=完成时间---到达时间 ； 带权周转时间=周转时间/服务运行时间

先来先服务算法（FCFS）是最简单的调度算法，有利于长作业，不利于短作业

短作业优先调度算法（SJF）针对短作业

**举例：**已知系统中有两台打印机，又有A、B、C、D四个进程都要使用打印机，分析其wait、signal操作和信号量变化。

● 设打印机资源信号量为s，初值为2。

A:	B:	C:	D:
{ wait(s);	{ wait(s);	{ wait(s);	{ wait(s);
使用打印机;	使用打印机;	使用打印机;	使用打印机;
signal(s);	signal(s);	signal(s);	signal(s);
}	}	}	}

s=1	s=0	s=-1	s=-2	s=-1	s=0	s=1	s=2
-----	-----	------	------	------	-----	-----	-----

### 第1种解决方法方法:

```
Semaphore chopstick [5]={1, 1, 1, 1, 1};
Semaphore sm=4;
while(true){
    wait (sm) ;
    wait ( chopstick[i] ) ;
    wait ( chopstick[(i+1)%5] ) ;
    .....
    eating;
    .....
    signal ( chopstick[i] ) ;
    signal ( chopstick[(i+1)%5] ) ;
    signal ( sm ) ;
    .....
    thinking;
}
```

### 第3种解决方法方法: 则第i位哲学家的活动描述:

```
Semaphore chopstick [5]={1, 1, 1, 1, 1};
```

```
while(true)
{ if i%2==0 then
{ wait(chopstick[i]);
  wait(chopstick[(i+1)%5]);
  .....
  eating;
  .....
  signal(chopstick[i]);
  signal(chopstick[(i+1)%5]);
  .....
  thinking;
}
```

```
else
{wait(chopstick[(i+1)%5]);
 wait(chopstick[i]);
  .....
  eating;
  .....
  signal(chopstick[(i+1)%5]);
  signal(chopstick[i]);
  .....
  thinking;
}
}
```

作业 调度 情况 算法	进程名	A	B	C	D	E	平均
	到达时间	0	1	2	3	4	
	服务时间	4	3	5	2	4	
FCFS	完成时间	4	7	12	14	18	
	周转时间	4	6	10	11	14	9
	带权周转时间	1	2	2	5.5	3.5	2.8
FJS	完成时间	4	9	18	6	13	
	周转时间	4	8	16	3	9	8
	带权周转时间	1	2.67	3.1	1.5	2.25	2.1

● 周转时间 = 完成时间 - 到达时间

● 带权周转时间 = 周转时间 / 服务时间

**例1:** 已知某分页系统，主存容量为64k，页面大小为1k，对一个4页大的作业，第0、1、2、3页被分配到内存的2、4、6、7块中。

求：将十进制的逻辑地址1023、2500、4500转换成物理地址。

解：(1)  $1023/1K$ ，得到页号为0，页内地址1023。

又对应的物理块号为2，故物理地址为 $2*1k+1023=3071$

(2)  $2500/1K$ ，得到页号为2，页内地址452。

又对应的物理块号为6，故物理地址为 $6*1k+452=6596$

(3)  $4500/1K$ ，得到页号为4，页内地址404。

因为页号不小于页表长度，故产生越界中断。

例1：在一个请求分页系统中，假定系统分给一个作业的物理块数为3，并且此作业的页面走向为2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2。用FIFO、LRU、OPT计算缺页次数和缺页率。

**分析：**如果所访问的页还没有装入内存，将发生一次缺页中断。访问过程中发生缺页中断的次数就是缺页次数。缺页次数除以总的访问次数，就是缺页率。

76

**使用OPT算法：**将来再也不用或最长时间不用的页面——黄色标志

页面走向	2	3	2	1	5	2	4	5	3	2	5	3
1	2	2	2	2	2	2	4	4	4	2	2	2
2		3	3	3	3	3	3	3	3	3	3	3
3				1	5	5	5	5	5	5	5	5
缺页中断	+	+		+	+		+			+		

缺页次数：6，缺页率：6/12，页面置换3次

77

## 使用FIFO算法:

驻留内存最久的页面——黄色标志

页面走向	2	3	2	1	5	2	4	5	3	2	5	2
1	2	2	2	2	5	5	5	5	3	3	3	3
2		3	3	3	3	2	2	2	2	2	5	5
3				1	1	1	4	4	4	4	4	2
缺页中断	+	+		+	+	+	+		+		+	+

缺页次数: 9, 缺页率: 9/12; 页面置换6次

79

## 使用LRU算法:

长时间没有访问的页面——黄色标志  
刚刚访问过的页面——绿色标志

页面走向	2	3	2	1	5	2	4	5	3	2	5	2
1	2	2	2	2	2	2	2	2	3	3	3	3
2		3	3	3	5	5	5	5	5	5	5	5
3				1	1	1	4	4	4	2	2	2
缺页中断	+	+		+	+		+		+	+		

LRU最接近OPT, 表明LRU优于FIFO。

81

例2：在一个请求分页系统中，假如一个作业的页面走向为1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5, 当分给该作业的物理块数M分别为3和4时，请用FIFO计算缺页次数和缺页率，并比较所得的结果。

82

### 使用FIFO算法——物理块数为3:

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1	4	4	4	5	5	5	5	5	5
2		2	2	2	1	1	1	1	1	3	3	3
3			3	3	3	2	2	2	2	2	4	4
缺页中断	+	+	+	+	+	+	+			+	+	

缺页次数：9，缺页率：9/12

83