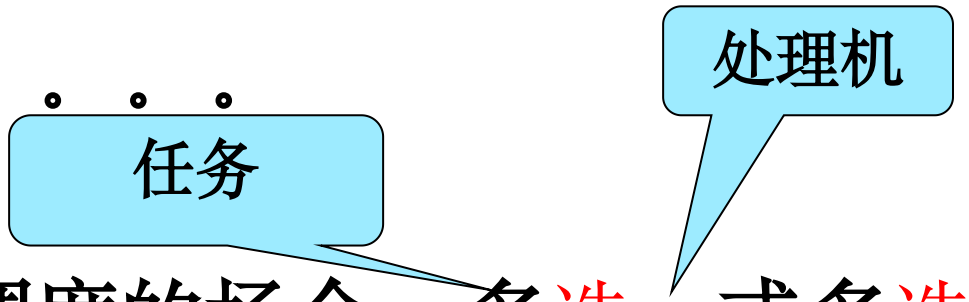
第3章 处理机调度

公交站调度的例子：



调度的场合：多选一或多选少

调度的内容：（1）挑选（最值或top N）

（2）分派

3.1 处理机调度的类型

- 处理机调度需要解决:
  - 多任务以**何种方式**共享处理机。互斥
  - 处理机**时间的分配及长短**确定。合理
  - 处理机分配策略。如何分配
  - 进程切换频度与系统效率的权衡。频繁开销大，反之则并发性低。如何权衡

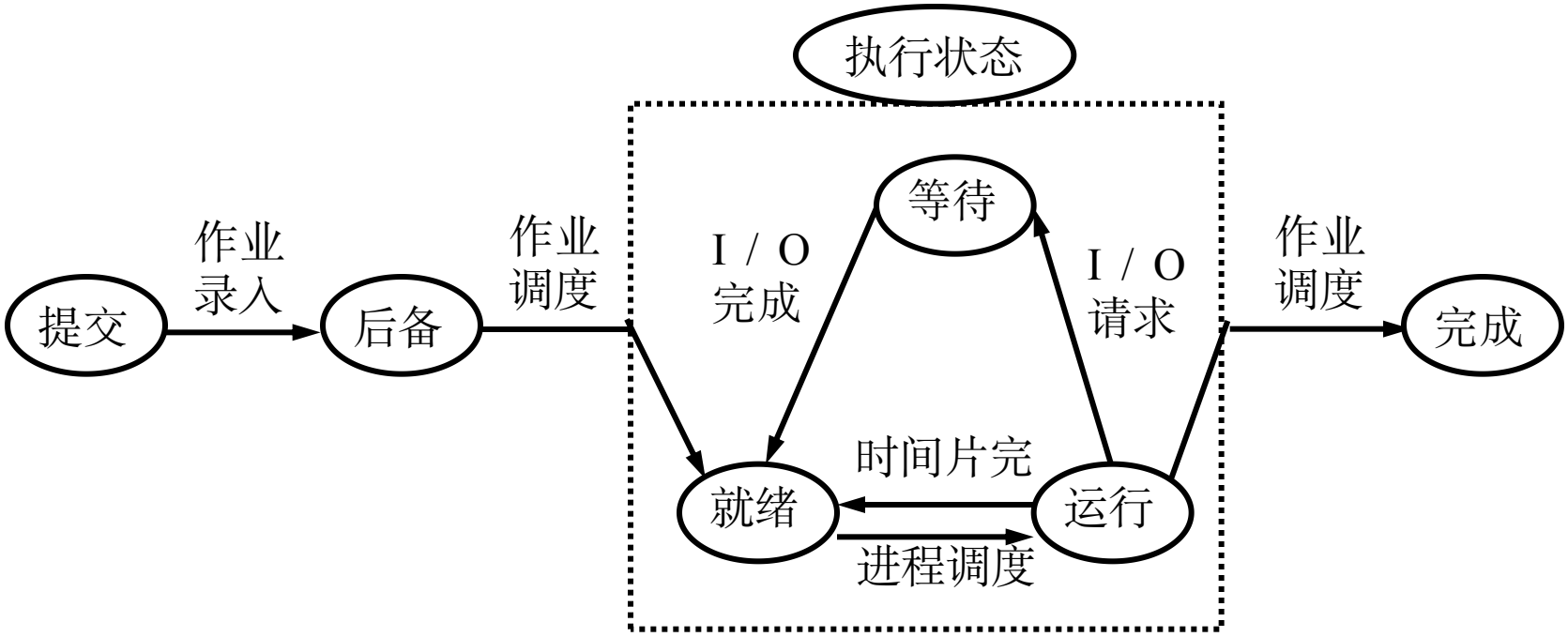
处理机的三级调度

- 处理机的三级调度:
  - 作业调度（高级调度、宏观调度、长程调度）
  - 进程调度（低级调度、微观调度）
  - 中程调度（交换调度、中程调度）

1.作业的状态

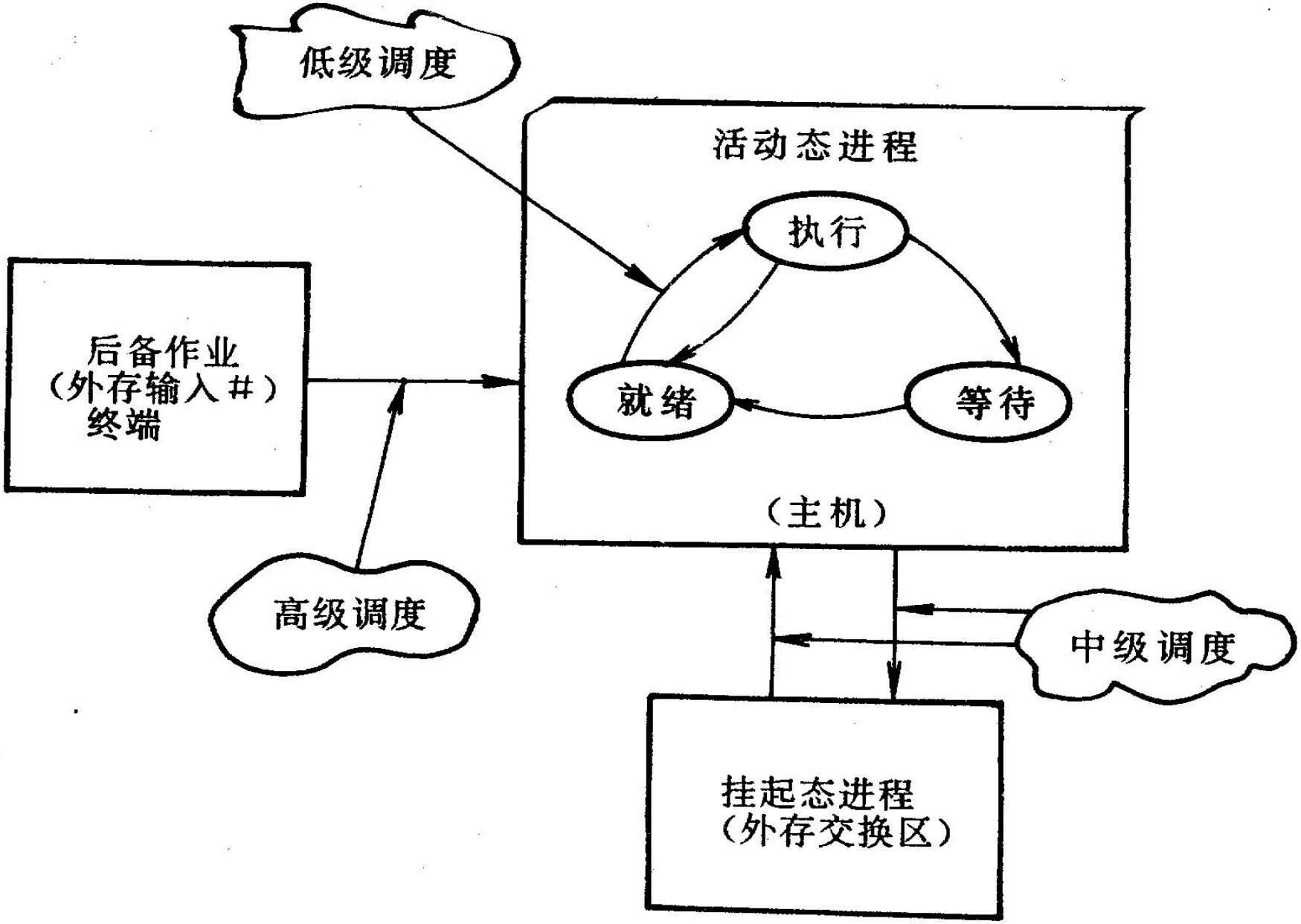
- 作业从提交到完成要经历四种状态：
  - 提交状态：用户作业由输入设备向系统外存输入时作业所处的状态。
  - 后备状态：作业输入到外存后，系统为其建立了作业控制块，并把它插入到**后备作业队列**中等待调度运行。
  - 执行状态：作业在内存中执行。
  - 完成状态：作业正常或异常结束，但作业占有的资源还未被系统全部回收。

作业状态转换图



作业调度

- 作业调度的主要任务是按**一定的原则（作业调度算法）**从外存上处于**后备状态的作业（后备队列）**中选择一个或多个作业，给它们分配内存、输入/输出设备等必要的资源，并建立相应的进程，以使该作业具有获得竞争处理机的权利。
- 作业调度的运行频率较低，通常为**几分钟**一次。


低级调度
活动态进程
执行
后备作业
（外存输入#）
终端
就绪
等待
（主机）
高级调度
中级调度
挂起态进程
（外存交换区）

2.中程调度

- 中程调度又称中级调度或交换调度，其功能是将内存中暂时不用的信息移到外存，以腾出空间给内存中的进程使用，或将需要的信息从外存读入内存。
- 引入中程调度的目的是提高内存利用率和系统吞吐量。
- 中程调度的运行频率介于两者之间。

中程调度程序

- 中程调度程序由**换入**和**换出**两个过程组成：
  - 换出过程把内存中的程序或数据换到交换区。
  - 换入过程把外存中的程序或数据换到内存。
- 为了加快交换速度，外存交换区采用连续分配方式。

3.进程调度

- 进程调度又称低级调度、微观调度或短程调度，其主要任务是按照某种策略和方法从**就绪队列**中**选取**一个进程，将处理机分配给它。
- 进程调度的运行频率很高，一般**几十毫秒要运行一次**。

进程调度的功能

- 记录系统中所有进程的状态、优先数和资源情况。
- 按**调度算法**选择进程运行。
- 实施处理机的切换（选中进程和当前进程之间）。

引起进程调度的原因

- 正在运行进程结束
- 运行进程因某种原因阻塞，如P操作、I/O等
- 有进程进入就绪队列且就绪队列为空，或进程优先级高于当前运行进程且为剥夺调度方式
- 从系统调用或中断返回
- 时间片用完

4. 选择调度算法的准则

- 由于操作系统的类型及目标不同，因此选择的调度算法也不同。
- 选择调度算法有以下准则：
  - 面向**系统**的准则
  - 面向**用户**的准则

面向系统的准则

- **公平性**：系统中的每个进程应获得合理的CPU时间。
- **CPU利用率高**：对微机和实时系统不太重要。
- **系统吞吐量大**：吞吐量指单位时间内所完成的进程数。
- **合理利用各类资源**：让各类资源都忙碌（负载均衡），对微机不太重要。

面向用户的准则

- 周转时间短：指从作业**提交**到作业完成的时间间隔。（批处理系统）
- 响应时间快：指从用户**提交请求**到系统产生响应的时间间隔。（分时系统）
- 截止时间的保证：截止时间是指某任务必须开始执行或必须完成的最迟时间。（实时系统）

周转时间

- 作业的周转时间是指从作业提交到作业完成之间的时间间隔。
- **平均周转时间**是指多个作业的周转时间的平均值。 n 个作业的平均周转时间：
  - $T = (T_1 + T_2 + \ldots + T_n) / n$ （T_i为作业 i 的周转时间）

带权周转时间

- 带权周转时间是指作业周转时间与作业**实际运行时间**的比。
- 平均带权周转时间是指多个作业的带权周转时间的平均值。 n 个作业的平均带权周转时间:
  - $W = (W_1 + W_2 + \cdots + W_n)/n$ （W_i为作业 i 的带权周转时间）

3.2 调度算法

- 调度算法（挑选原则）。
- 本章的算法有些适合作业调度，有些适合进程调度，有些适用于两者。

1. 先来先服务调度算法FCFS(first come first serve)

- 先来先服务算法既可用于作业调度，也可用于进程调度。
  - 在作业调度中：从后备作业队列中选择一个或多个最先进入该队列的作业，将它们调入内存，为它们分配资源，创建进程，然后放入就绪队列。
  - 进程调度中：从**就绪队列**中选择一个**最先进入该队列**的进程，为之分配处理机，使之投入运行。该进程一直运行到完成或因等待某一事件而阻塞时才释放处理机。

算法如何实现？（以进程调度为例）

方案一：

1、进程**乱序**进入就绪队列（每次进入就绪队列时不考虑调度算法参照的因素的取值，例如从尾部挂入队列）

2、挑选时，从队列中挑选**最值**元素进行分派

方案二：

1、进程严格按照调度参照的因素的取值入就绪队列（队列始终保持**有序**）

2、调度时挑选**队首或队尾**元素进行分派

哪种方案更优？

"入队列"为低频事件，"挑选"为高频事件。所以方案二更优!

调度**算法**由调度队列的**排队原则**体现!

先来先服务调度算法例

调度队列按照调度对象**到达队列的先后**排队

- 设有4道作业，它们的提交时间及执行时间如下表，若按先来先服务调度算法进行调度，试计算4个作业的平均周转时间和平均带权周转时间。（时间单位：小时，以**十进制**计算）。

作业	提交时间	估计运行时间
1	10	2
2	10.2	1
3	10.4	0.5
4	10.5	0.3

作业周转时间及带权周转时间的计算

调度顺序：1,2,3,4

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	10	2	10	12	2	1
2	10.2	1	12	13	2.8	2.8
3	10.4	0.5	13	13.5	3.1	6.2
4	10.5	0.3	13.5	13.8	3.3	11

- 平均周转时间T=(2.0＋2.8＋3.1＋3.3)/4=2.8
- 平均带权周转时间W=(1＋2.8＋6.2＋11)/4=5.25

先来先服务算法特点

- 算法简单，易于实现（**入队时入队尾，调度时取队首**）。
- 但不利于短作业。（**带权周转**时间长）
- 不利于IO频繁型作业。（**多次在就绪队列排队**）

2.短作业（进程）优先调度算法

调度队列按照调度对象的**估计运行时间**从短到长排队

- 在作业调度中，从后备队列中选择一个或多个估计运行时间最短的作业，将它们调入内存运行。
- 在进程调度中，从就绪队列中选择一个**估计运行时间最短**的进程，为之分配处理机，使之投入运行。该进程一直运行到完成或因等待某一事件而阻塞时才释放处理机。

短作业优先调度算法例

调度顺序是？

1,4,3,2!

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
1	10	2	10	12	2	1
2	10.2	1	12.8	13.8	3.6	3.6
3	10.4	0.5	12.3	12.8	2.4	4.8
4	10.5	0.3	12	12.3	1.8	6

- 平均周转时间　T=(2.0＋1.8＋2.4＋3.6)/4=2.45
- 平均带权周转时间　W=(1＋6＋4.8＋3.6)/4=3.85

短作业优先调度算法的特点

- 算法调度性能较好，（能获得**最短**的平均带权周转时间）

例如上例中，	先来先服务	短作业优先
平均周转时间	2.8	2.45
平均带权周转时间	5.25	3.85

- 但对长作业不利，会造成长进程**饥饿**
- 运行时间为估计。（**不准!** ）

3. 时间片轮转调度算法

就绪队列的排队原则是？

- 时间片轮转法：系统将所有就绪进程按到达时间的先后次序排成一个队列，每次调度时把CPU分配给队首进程，并令其执行一个时间片。当时间片用完时，停止该进程的执行，将它送至就绪队列末尾等待下一次执行，然后再把处理机分配给就绪队列中的新队首进程。如此不断循环，直至完成为止。

- 类比：跳大绳运动

时间片轮转算法例

- 设有A、B、C、D、E五个进程，其到达时间分别为0、1、2、3、4，要求运行时间依次为3、6、4、5、2，采用时间片轮转调度算法，当时间片大小为1和4时，试计算其平均周转时间和平均带权周转时间。

时间片大小为1

- A、B、C、D、E要求运行时间依次为3、6、4、5、2，到达时间依次为0、1、2、3、4。

0：A运行；
1：B运行，A等待；
2：A运行，CB等待；
3：C运行，BDA等待；
4：B运行，DAEC等待；
5：D运行，AECB等待；
6：A运行，ECBD等待；
7：E运行，CBD等待；
8：C运行，BDE等待；
9：B运行，DEC等待；
10：D运行，ECB等待；
11：E运行，CBD等待；
12：C运行，BD等待；
13：B运行，DC等待；
14：D运行，CB等待；
15：C运行，BD等待；
16：B运行，D等待；
17：D运行，B等待；
18：B运行，D等待；
19：D运行，

时间片为1的周转时间

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
A	0	3	0	7	7	2.33
B	1	6	1	19	18	3
C	2	4	3	16	14	3.5
D	3	5	5	20	17	3.4
E	4	2	7	12	8	4

- 平均周转时间　T=(7＋18＋14＋17+8)/5=12.8
- 平均带权周转时间 W=(2.33＋3＋3.5＋3.4+4)/5=3.246

时间片大小为4

- A、B、C、D、E要求运行时间 依次为3、6、4、5、2， 到达时间依次为0、1、2、3、4。

0：A运行，BCD依次到达；
3：B运行，CD等待，后E到达；
7：C运行，DEB等待；
11：D运行，EB等待；
15：E运行，BD等待；
17：B运行，D等待；
19：D运行

A、B、C、D、E开始时间依次为0、3、7、11、15；
结束时间依次为3、19、11、20、17。

时间片为4的周转时间

作业	提交时间	运行时间	开始时间	完成时间	周转时间	带权周转时间
A	0	3	0	3	3	1
B	1	6	3	19	18	3
C	2	4	7	11	9	2.25
D	3	5	11	20	17	3.4
E	4	2	15	17	13	6.5

- 平均周转时间　T=(3＋18＋9＋17+13)/5=12
- 平均带权周转时间 W=(1＋3＋2.25＋3.4+6.5)/5=3.23

时间片大小的选择

- 若时间片太大，所有进程都能在一个时间片内完成，则时间片轮转算法退化为先来先服务；
- 若时间片太小，则进程调度频繁，系统开销增加。
- 因此时间片大小选择应适当。

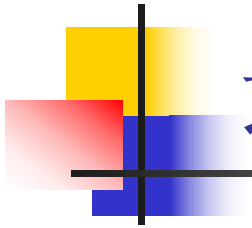
确定时间片大小应考虑的因素

- 系统对响应时间的要求：响应时间=时间片*进程数。进程数一定，则时间片与系统响应时间成正比。
- 就绪队列中的进程数目：时间片与就绪进程数成反比。
- 系统处理能力：人所能承受的响应时间一定，系统速度快则时间片可增长。

4.优先权调度算法

调度队列按照调度对象的优先权从高到低排队

- 在作业调度中，从后备作业队列中选择若干优先权高的作业调入内存。
- 在进程调度中，将处理机分配给就绪队列中优先权最高的进程。
- 优先权通常用优先数来衡量。在某些系统中，优先数越大优先权越高；而在另一些系统中，优先数越大优先权越小。



进程调度方式

- 进程调度有两种方式:
  - 非抢占方式
  - 抢占方式

非抢占方式

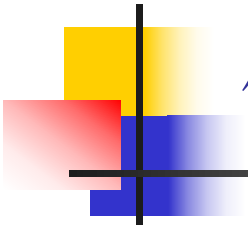
- 非抢占方式：又称非剥夺方式、不可剥夺方式、不可抢占方式。这种调度方式是指一旦将处理机分配给某进程后，便让该进程一直执行，直到该进程完成或发生某事件而进入阻塞状态，才把处理机分配给其他进程。
- 非抢占方式中引起进程调度的因素有：进程结束、因某种原因而阻塞、执行同步原语等。
- 特点：简单，系统开销小，但无法处理紧急任务。

抢占方式

- 抢占方式：又称剥夺方式、可剥夺方式。这种调度方式是指允许调度程序根据某种原则去停止正在执行的进程，将已分配给该进程的处理机重新分配给其他进程。
- 抢占原则有：优先权、最短剩余时间优先、时间片。

按调度方式对优先权调度算法分类

- 非抢占式优先权算法：系统一旦将处理机分配给就绪队列中优先权最高的进程后，该进程便一直运行下去，直到完成或因发生某事件使该进程放弃处理机时，系统才将处理机分配给另一个更高优先权的进程（主动释放CPU）。
- 抢占式优先权调度算法：将处理机分配给优先权最高的进程，使之运行。在进程运行过程中，一旦出现了另一个优先权更高的进程时，进程调度程序就停止原运行进程，而将处理机分配给新出现的高优先权进程(**被动释放CPU**)。



优先权的类型

- 优先权分为两种：
  - 静态优先权
  - 动态优先权

静态优先权

- 静态优先权是在创建进程时确定的，确定之后在整个进程运行期间不再改变。
- 确定依据有：
  - 进程类型：系统，用户
  - 进程对资源的需求：执行时间，资源数量
  - 用户要求：紧迫程度
  - 到达时间：先到则优先权高
- 特点：简单易行，系统开销小，但不精确。

动态优先权

- 动态优先权是指在创建进程时，根据进程的特点及相关情况确定一个优先权，在进程运行过程中再**根据情况的变化调整**优先权。
- 确定原则有：占用CPU时间，等待时间。
- 例：优先数=CPU使用时间/2+基本优先数
  - CPU使用时间衰减函数:
  - Decay(CPU)=CPU/2

5.最高响应比优先调度算法

- 最高响应比优先调度算法是对短作业优先调度算法和先来先服务调度算法的一种综合。

- 响应比＝（等待时间+运行时间）/运行时间

调度队列按照调度对象的**响应比**降序排列

最高响应比优先调度算法思想

- 在每次调度时，先计算调度队列中每个对象的响应比并排序，然后挑选响应比最高者投入运行。
- 特点：
  - 等待时间相同，短作业优先-----有利于短作业，
  - 运行时间相同，等待时间长的作业优先运行----先来先服务。
  - 长进程不会饥饿

6.多级队列调度算法

- 实现思想：根据作业性质或类型不同，将进程就绪队列分为多个，每个队列采用不同的调度算法。
- 例如：终端型作业为前台作业，批处理作业为后台作业。前台采用时间片轮转算法，后台采用先来先服务，前台作业的优先级高。

7.多级反馈队列调度算法（1）

- 应设置多个就绪队列，并为每个队列赋予不同的优先级和时间片。
- 其中：P1>P2>...>Pn
- Q1<Q2<...<Qn

多级反馈队列调度算法（2）

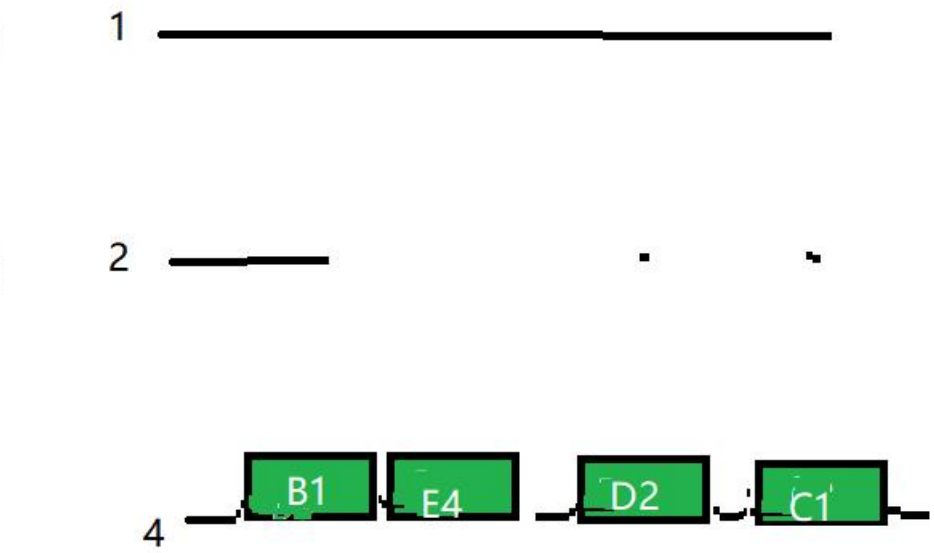
- 当一个新进程进入系统时，首先将它放入第1个队列的末尾，按先来先服务的原则排队等待调度。当轮到该进程执行时，
- （1）在此时间片内完成，便可准备撤离系统；
  （2）它在一个时间片结束时尚未完成，调度程序便将该进程转入下一级队列的末尾，再同样地按先来先服务原则等待调度执行。
- （3）时间片未用完即要进行IO,待IO完成被唤醒后入原队列队尾

多级反馈队列调度算法（3）

- 仅当第1个队列为空时，调度程序才调度第2队列中的进程运行；仅当第1个至第（i－1）个队列均为空时，才会调度第i个队列中的进程运行。
- 当处理机正在为第i个队列中的某进程服务时，若又有新进程进入优先级较高的队列中，则此时新进程将抢占正在运行进程的处理机，即由调度程序把正在执行进程放回第i个队列末尾，重新将处理机分配给新进程。

多级反馈队列调度算法例

- 设有A、B、C、D、E五个进程，其到达时间分别为0、1、3、4、5，要求运行时间依次为3、8、4、5、7，采用多级反馈队列调度算法，系统中共有3个队列，其时间片依次为1、2和4，试计算其平均周转时间和平均带权周转时间。



调度分析

- A、B、C、D、E到达时间依次为0、1、3、4、5，要求运行时间依次为3、8、4、5、7

0：A运行；
1：B运行，A等待；
2：A运行，B等待；
3：C运行，BA等待；
4：D运行，BAC等待；
5：E运行，BACD等待；
6：BB运行，ACDE等待；
8：A运行，CDE等待；B等待
9：CC运行，DE等待；B等待
11：DD运行，E等待；BC等待
13：EE运行，BCD等待；
15：BBBB运行，CDE等待；
19：C运行，DEB等待；
20：DD运行，EB等待；
22：EEEE运行，B等待；
26：B运行。

多级反馈队列调度算法的性能

- 多级反馈队列调度算法能较好满足各类用户的需求：
  - 终端型用户：大多能在一个时间片内完成，响应时间较短；
  - 短批处理作业用户：能在前几个队列完成，周转时间较短；
  - 长批处理作业用户：依次在1～n队列中运行，不会长时间得不到处理。