

## 3.4 openEuler的进程/线程调度

## 3.4.1 openEuler进程调度策略

### ■ openEuler多种调度策略

- 限期进程对应的是限期调度策略，该进程会选择进程截止时间距当前时间点最近的进程来运行。
- 实时进程对应两种调度策略：先进先出调度策略和轮转调度策略。
- 普通进程对应的是标准轮流分时调度策略，采用的是CFS调度（完全公平调度）算法。

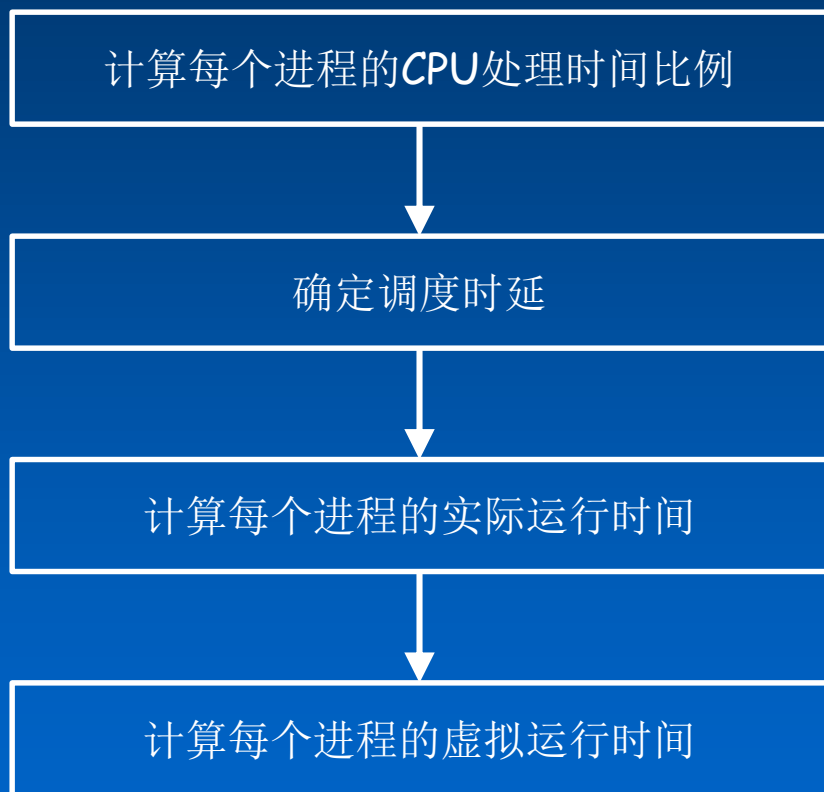
## 3.4.2 CFS调度算法

### ■ CFS调度所解决的问题

- 使用先进先出调度策略与轮转调度策略在实时进程的调度队列中，会出现部分低优先级的队列无法被调用。
- **CFS**调度算法会根据进程优先级和当前系统负载来为每个进程分配一定比例的**CPU**处理时间。保证所有进程都会被调度，只是运行的时间不同。

## 3.4.2 CFS调度算法

### ■ CFS调度策略



CFS调度策略执行流程

(1)计算每个进程的CPU处理时间比例

□ 首先为每个进程分配一个**nice**值,表示相对优先级。并根据**nice**值的权重转换表,将**nice**值转换为对应的权重值。

□ 假设进程*i*的权重为 $w_i$ ,那么*i*所分配的CPU处理时间比例为:

$$P_i = w_i / \sum_{j \in cfs} w_j$$

## 3.4.2 CFS调度算法

(2)确定调度时延(sched\_period)

- 调度时延即进程连续两次获得CPU的时间间隔。
- 则CFS调度算法的调度时延为CFS调度队列上就绪进程数与进程运行的最小粒度时间的乘积。

(3)计算每个进程的实际运行时间( $T_i$ ):

$$T_i = \text{sched\_period} * P_i$$

(4)计算每个进程的虚拟运行时间(vruntime),帮助选择下一个进程。

- 进程i的虚拟运行时间为:(其中 $w_{\text{nice}0}$ 为nice值为0时的权重)

$$V_i = T_i * w_{\text{nice}0} / w_i$$

- CFS调度算法将按照虚拟运行时间从小到大排序，每次选择最小的虚拟运行时间所对应的进程进行调度。
- 随着时间的增长，正在运行进程的vruntime将不断增大。没有得到执行的进程vruntime不变，从而下一次更容易被调用。

### 3.4.3 多核调度策略

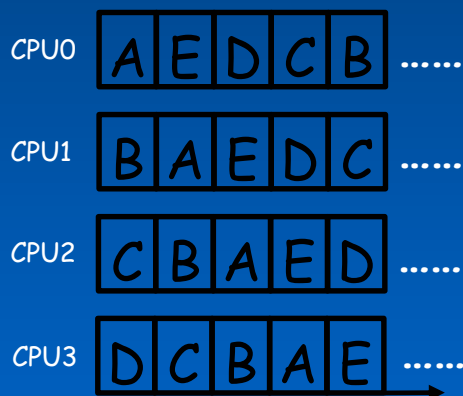
- 多核处理器缓存和主存的关系发生变化之后，系统主要会面临如下问题：
  - 缓存一致性问题；
  - 缓存亲和性问题；
  - 核间数据共享；
  - 负载均衡。
- 针对这些存在的问题，现有的多核调度策略主要分为两类：
  - 单队列调度策略
  - 多队列调度策略

## 3.4.3 多核调度策略

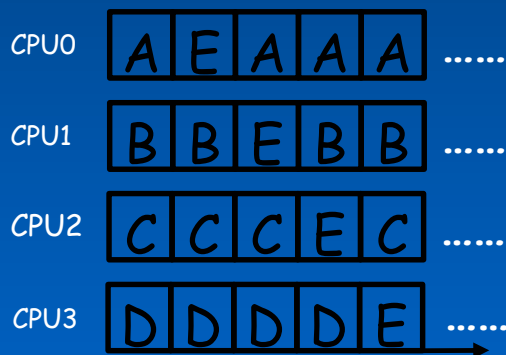
### 3.4.3.1 单队列调度策略



调度队列情况：5个就绪进程



单队列调度情况(策略一)



考虑亲和性后的单队列调度情况(策略二)

两种策略的不足：

**策略一：**每个CPU使用轮转调度算法选择下一个要执行的进程。进程会在不同CPU间转移，违背了缓存亲和性。

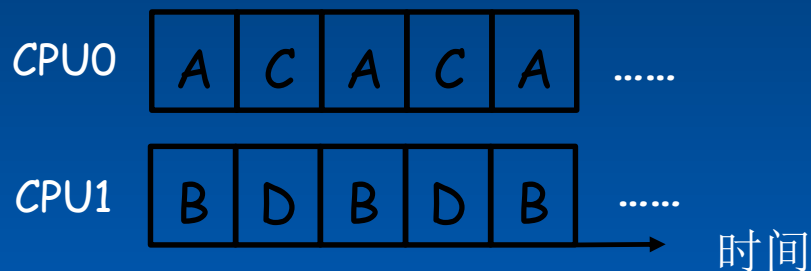
**策略二：**引入亲和度机制尽可能地让进程在同一个CPU上执行。牺牲了某些进程的亲和性。

## 3.4.3 多核调度策略

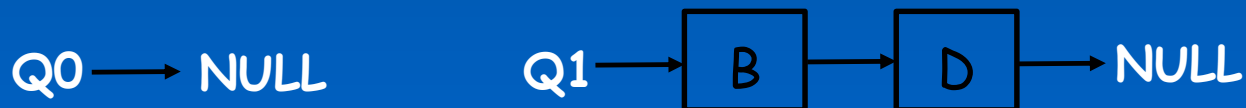
### 3.4.3.2 多队列调度策略



openEuler使用多队列调度策略。图中队列Q0由CPU0维护，Q1由CPU1维护。



采用轮转调度算法，调度之初的情况。



一段时间后，假设A、C两个进程都结束，此时负载失衡。解决多CPU负载不均衡问题最直接的方法就是让就绪进程跨CPU迁移。比如上图中将进程D迁移到CPU0上运行后，CPU0和CPU1则实现了负载均衡。



## 3.4.3 多核调度策略

### 3.4.3.2 多队列调度策略

- 多CPU负载不均衡解决方法:

在openEuler中，每个处理器都有一个迁移线程（称为migration/CPU-ID），每个迁移线程都有一个由函数组成的停机工作队列。迁移线程每次都从停机工作队列中取出一个函数执行。只要停机工作队列上有待执行的函数，迁移线程就会一直运行直至所有的函数都执行完毕。这些函数负责将进程从一个CPU迁移到另一个CPU。另外，迁移线程可以抢占所有其它进程，但是其它进程不可抢占迁移线程。

上图的典型场景是：

- CPU0向CPU1的停机工作队列中添加一个工作函数，并唤醒CPU1上的迁移线程；
- 该迁移线程不会被其他进程抢占，故其第一时间从停机工作队列中取出函数执行，将进程从CPU1迁移到CPU0，实现负载均衡。