

作业调度 进程调度 先来的先分配处理机 优点: 算法简单 对长作业有利 有利于CPU繁忙型作业(计算型) 先来先服务算法 缺点:效率低不利于短作业 不利于IO繁忙型作业 不会导致饥饿 非抢占式的算法 讲程调度 优先选择预计运行时间最短的进程 优点:平均等待时间 平均周转时间最短 短作业优先算法 缺点:对长作业不利,造成饥饿现象,没有考虑作业的紧迫性,用户可能可以缩短作业预估时间,使得无法做到短作业优 产生"饥饿"现象。如果一直得不到服务,则称为"饿死" SJF和SPF(短进程优先(SPF)算法)是非抢占式的算法。但是也有抢占式的版本——最短剩余时间优先算法 作业调度 进程调度 剥夺型 立即停止当前运行进程,将处理机分配给更高优先级进程 分类 非剥夺型 等待当前进程运行完成,然后将处理机分配给更高优先级进程 静态优先级:进程创建后无法对优先级进行修改 优先级分类 动态优先级:可以根据进程运行状态,对进程优先级进行动态调 优先级调度算法 系统进程>用户进程 优先级设置原则 交互型进程>非交互型进程 I/O进程>计算型进程(cpu繁忙型) 产生"饥饿"现象 有抢占式的, 也有非 抢占式的 响应比=(等待时间+要求服务时间)/要求服务时间=1+等待时间/要求服务时间 等待时间相同情况下,要求服务时间越短响应比越大,有利于短作业进程 要求服务时间相同,作业响应比由其等待时间决定,等待时间越长响应比越高,实现先来先服务 高响应比调度算法 对于长作业,作业的响应比可以随等待时间的增加而提高,等待时间足够长时,其响应比可以升到很高,从而获得处理 机 不会导致饥饿 非抢占式的算法 使用与分时系统,使用时间片,就绪进程按照到达先后排成队列,依次在时间片内占用处理机,时间片到达时就释放处理机 时间片选择很重要,过大就变成了先来先服务,过短又变成了短作业优先 时间片轮转算法 时间片影响因素:系统响应时间,就绪队列中的进程数目和系统的处理能力 不会导致饥饿 抢占式 设置多个就绪队列,为每个队列设置不同的优先级,优先级依次递减

实现思想

优点

抢占式

产生"饥饿"现象

多级反馈队列调度算法(融合前面几种算法)

每个队列中的时间片各不相同,时间片依次递增

终端型作业用户:短作业优先

短批处理作业用户:周转时间较短

只有到高级队列为空的时候, 低等级队列才能开始调度。

长批处理作业用户:讲过前面几个队列得到部分执行,不会长期得不到处理

每个队列按照先来先服务原则进行进程排队,若规定时间片内没有完成,就将进程放入下一级队列

2.2处理机调度(下)

典型调度算法