

实验五 进程调度

一. 实验目的

1. 深入了解进程调度的策略和机制
2. 掌握各种调度算法的基本原理和评价指标
3. 实现几种常用的调度算法

二. 实验工具与设备

装有 Linux 操作系统的计算机。

三. 实验内容

1. 掌握进程调度的调度策略和调度机制

处理机（CPU）是指在一组就绪的进程中进行 CPU 的分配。概念上，操作系统调度程序是由用于上下文切换的机制和确定就绪进程分配 CPU 顺序的策略构成的。现代操作系统中，硬件和进程管理器的数据结构和算法实现了机制。处理机调度所追求的性能主要是响应时间短、周转时间快和对截止完成时间的保证。

CPU 调度涉及到三个层次的调度，即高级调度（或**作业调度**）、中级调度（或**进程对换**）和低级调度（或处理机调度、CPU 调度、**进程调度**）。这里仅讨论低级调度。低级调度的对象是进程。对于单处理机系统，处理机调度，是从内存的就绪队列中挑选一个进程送到处理机上运行。它采用多道程序设计的操作系统中，在一段时间间隔内，允许多个进程加载到主存中。当一个进程执行结束、或等待某事件发生而进入阻塞状态时，由调度程序将 CPU 分配给另一个进程。

操作系统的调度程序有两项任务：调度和分派。前者实现调度策略，确定就绪进程竞争使用 CPU 的次序的裁决原则，即进程何时应放弃 CPU 和选择哪个进程来执行；后者实现调度机制，确定何时复用 CPU，处理上下文切换细节，完成进程和 CPU 的绑定与放弃的具体工作。

处理机有两种基本的调度策略，即抢占式（或称剥夺式）和非抢占式（或称非剥夺式）。在抢占方式下，操作系统有权按某种原则剥夺一个正在运行的进程的处理机。通常高优先级进程剥夺低优先级进程，或当运行进程时间片用完后被剥夺；而在非抢占式方式下，除非进程主动放弃，否则操作系统不得剥夺一个正在运行的进程的处理机。

处理机调度机制由 3 个逻辑功能程序模块组成：队列管理程序（将 PCB 指针放入等待 CPU 和进程列表中，计算进程的优先级）、上下文切换程序（将当前执行进程的上下文信息保存到其 PCB 中）和分配程序（从就绪队列中选择进程，把 CPU 出让给选中的进程）。

2. 掌握评价调度算法的性能指标

（1）**周转时间**：进程（或作业）从提交到完成（得到结果）所经历的时间。包括：在收容队列中等待，CPU 上执行，就绪队列和阻塞队列中等待，结果输出等待。批处理。包括：平均周转时间 T 和平均带权周转时间（带权周转时间 W 是 $T(\text{周转})/T(\text{CPU 执行})$ ）。

（2）**响应时间**：用户输入一个请求（如击键）到系统给出首次响应（如屏幕显示）的时间。分时系统。

（3）**截止时间**：开始截止时间和完成截止时间，与周转时间有些相似。该指标针对实时调度。

（4）**公平性**：不因进程（或作业）本身的特性而使上述指标过分恶化。如长进程等待很长时间。

（5）**优先级**：可以使关键任务达到更好的指标。

(6) 吞吐量：单位时间内所完成的进程（或作业）数，跟进程（或作业）本身特性和调度算法都有关系。

(7) 设备的利用率和均衡性：如 CPU 繁忙的进程（或作业）和 I/O 繁忙（指次数多，每次时间短）的进程（或作业）搭配。

3. 掌握调度算法的基本原理

处理机调度算法的核心问题就是采用什么算法将处理机分配给进程。处理机调度算法主要有，先来先服务（FCFS）算法、短进程优先（SPN）算法、最短剩余时间进程优先算法（SRT）、时间片轮转（RR）算法、高优先权优先（HPF）算法、高响应比优先调度（HRRN）算法和多级反馈队列调度（FB）算法等。本实验只涉及三种处理机调度算法，即 FCFS、SPN 和 RR 等三种调度算法。

(1) 先来先服务调度算法（FCFS）

该算法采用非剥夺策略，算法按照进程提交或进程变为就绪状态的先后次序，分派 CPU。当前进程占用 CPU，直到执行完或阻塞，才出让 CPU（非抢占方式）。在进程唤醒后（如 I/O 完成），并不立即恢复执行，通常等到当前进程出让 CPU。这是最简单的调度算法，比较有利于长进程，而不利于短进程，有利于 CPU 繁忙的进程，而不利于 I/O 繁忙的进程。

(2) 短进程优先调度算法（SPN）

该算法也采用非剥夺策略，对预计执行时间短的进程优先分派处理机。通常后来的短进程不抢先正在执行的进程。相比 FCFS 算法，该算法可改善平均周转时间和平均带权周转时间，缩短进程的等待时间，提高系统的吞吐量。缺点是对长进程非常不利，可能长时间得不到执行，且未能依据进程的紧迫程度来划分执行的优先级，以及难以准确估计进程的执行时间，从而影响调度性能。

(3) 时间片轮转算法（RR）

该算法采用剥夺策略。让就绪进程以 FCFS 的方式按时间片轮流使用 CPU 的调度方式，即将系统中所有的就绪进程按照 FCFS 原则，排成一个队列，每次调度时将 CPU 分派给队首进程，让其执行一个时间片，时间片的长度从几个 ms 到几百 ms。在一个时间片结束时，发生时钟中断，调度程序据此暂停当前进程的执行，将其送到就绪队列的末尾，并通过上下文切换执行当前的队首进程，进程可以未使用完一个时间片，就出让 CPU（如阻塞）。时间片轮转调度算法的特点是简单易行、平均响应时间短，但不利于处理紧急作业。在时间片轮转算法中，时间片的大小对系统性能的影响很大，因此时间片的大小应选择恰当。

4. 用 C 语言实现 FCFS、SPN 和 RR 算法

四. 实验指导

1. 调度算法的性能指标

进程周转时间 = 进程完成时间 - 进程提交时间

平均周转时间 = 所有进程的周转时间之和/进程数

带权周转时间 = 进程周转时间/进程运行时间

平均带权周转时间 = 所有进程的带权周转时间之和/进程数

2. 算法实现

实现时，每个进程由一个进程控制块（PCB）表示，进程控制块可包含如下信息：进程名、到达时间、完成时间、需要运行的时间、已用 CPU 时间、周转时间、带权周转时间、进程状态等。进程的到达时间为进程输入的时间，进程的运行时间按时间单位计算，每个进程的状态可以是就绪 W（Wait）、运行 R（Run）和完成 F（Finish）三种状态之一。

(1) FCFS 算法:

FCFS 算法的基本思想就是谁先来,就先服务谁。即每次调度是从就绪队列中选择一个最先进入就绪队列的进程,把处理器分配给该进程,使之得到执行。该进程一旦占有了处理机,就一直运行下去,直到该进程完成或因发生事件而阻塞,才退出处理机。实现时,可按请求处理机的次序(进程到来的时间戳)指定进程的优先顺序,让分派程序选择时间戳最早的进程,或者将就绪队列组织成一个简单的先进先出的数据结构,新进来的进程排到队尾,而分派程序从队列的头移走进程。先来先服务原则包含了日常生活中的公平观念,如排队买东西、在银行存取款等就体现了这种原则。

先来先服务算法的一个隐含条件是不能抢占,一个程序一旦启动就一直运行到结束或者受阻塞为止。这是因为一旦允许抢占,就破坏了先来先服务的原则的。

(2) SPN 算法:

FCFS 算法实现起来比较容易,但该算法的缺点也是明显的,即短的工作有可能变得很慢,因为其前面有很长的工作,这样造成用户的交互式体验就比较差。就像我们排队办事情,你要办的事情只要几分钟就可办好,而你前面的一个人办理的事情因为复杂需要一个小时,这时候你要等在他后面就十分不高兴。短进程优先调度算法(SPN)是对 FCFS 算法的改进,其目标是减少平均周转时间。算法对预计执行时间短的进程优先分派处理机,通常后来的短进程不抢先正在执行的进程。

SPN 调度算法采用非抢占策略,一旦将处理机分给某个进程,则在该进程运行完之前,不能剥夺其对处理机的控制权。

(3) RR 算法:

时间片轮转算法主要用于进程调度。采用此算法的系统,其就绪队列往往按进程到达的时间来排序。进程调度按一定时间片 q 轮番运行各个进程。进程按到达时间在就绪队列中排队,调度程序每次把 CPU 分配给就绪队列首进程使用一个时间片,运行完一个时间片释放 CPU,排到就绪队列末尾参加下一轮调度, CPU 分配给就绪队列的首进程。

实现中,可采用固定时间片和可变时间片轮转法。本实验只实现固定时间片轮转算法,对于该算法,其步骤如下:

- a. 所有就绪进程按 FCFS 规则排队;
- b. 处理机总是分配给就绪队列的队首进程;
- c. 如果运行的进程用完时间片,则系统就把该进程送回就绪队列的队尾,重新排队;
- d. 因等待某事件而阻塞的进程送到阻塞队列;
- e. 系统把被唤醒的进程送到就绪队列的队尾。

五. 思考题

1. 在实际的进程调度中,除了按调度算法选择下一个执行的进程外,还应该处理哪些工作?
2. 几种进程调度算法有何区别与联系?
3. 在 RR 调度算法中,如果将时间定量增长为一个任意大的数目,那么会产生什么影响?