Introducing CPU Scheduling Algorithms: The Photocopier Scenario

John K. Estell
Bluffton College

Retention is obtained from a combination of repetition and association. One methodology for promoting retention is to introduce a topic by first alluding to an association to which students can relate. This serves as a foundation upon which the technical material can then be based, making the learning of the involved concepts easier. Unfortunately, in the area of operating systems there is little time for repetition in the lecture and few materials are available that show associations. This paper is a first step in providing materials suitable for an operating systems course that illustrate concepts through associations.

The subject of CPU (central processing unit) scheduling algorithms is not one that students can easily comprehend in the abstract. Normally students are presented in lecture with one algorithm after another, and as the concept of a process is often new and fuzzy the students have no intuitive grasp of the material. However, there are associations between a CPU and a photocopying machine that can be used to assist in the teaching of CPU scheduling algorithms. Both devices can be used by at most one entity at a time and there is often a queue awaiting access. Students can readily relate to waiting in line to use the photocopier, and because of this familiarity the associations can be presented in a short amount of time. By introducing the concepts behind the various types of scheduling algorithms using concrete examples in this context, students are given the framework that allows them to understand how the algorithms work. When the actual scheduling algorithms are then introduced using the traditional methodology, the concepts have already been explained, and the associations allow them to remember the technical material.

The lectures on this topic begin with some background explanatory material. One of the fundamental actions of a multitasking operating system is the scheduling of resources. The CPU is one of the primary resources available on a computer system; the goal of CPU scheduling is to assign processes to be executed by the CPU over time in such a way that system objectives are met. There are a variety of schedulers that affect the long-term, medium-term, and short-term performance; the focus of this paper is on the short-term CPU scheduling algorithms performed by the dispatcher. The success of CPU scheduling depends on the property that process execution consists of a cycle of CPU execution and I/O wait; when a process is in an I/O wait state, another process should be allowed to use the currently idle CPU to increase the utilization of the resource. Ideally, the allocation of the CPU will be performed so as to optimize one or more aspects of system behavior, such as minimal response time, minimal overhead, maximal resource utilization, or maximal throughput.

Once the basic definitions are presented, the concepts behind some well-known CPU scheduling algorithms can now be introduced by stating the following problem: In a library we have several people and one photocopier. Each person has a variety of items to photocopy - some have one page, others a few pages out of several books, and there are also those who want to copy an entire book. How do we allocate access to the photocopier? Before continuing on, it can be briefly shown to the class that the persons in the example correspond to the various types of processes encountered on a computer system. Each individual represents a process. There are short processes (copying one page) and long processes (copying entire book). There are CPU-bound processes that will perform many computations without interruption (copying a sequence of pages from one book) and there are I/O-bound processes that can perform only a few computations before an interruption occurs (copying a few pages each from several books, or single pages scattered throughout a single book). The CPU scheduling algorithms are presented by their actual name, but are explained only in terms of the photocopier scenario. To further illustrate the concepts, clip art graphics are used to visually show what is being discussed for each algorithm. The basic illustration, which is what students are most familiar with, is shown in Figure 1: a line of individuals waiting to use the photocopier.

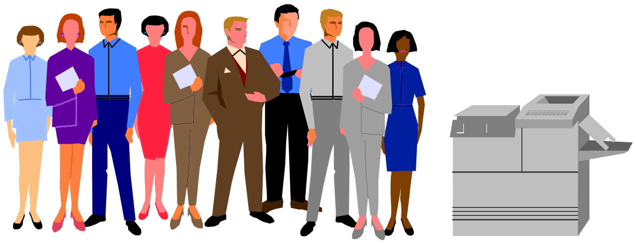

Figure 1. Representation of First Come First Served scheduling algorithm.

This represents the FCFS (first come, first served) scheduling algorithm, where whoever arrives at the photocopier first gets to use it to make as many copies as desired. A distinction is made between the person using the photocopier, which corresponds to the active CPU process, and the people waiting in line to use the photocopier, which corresponds to the ready queue containing processes that are ready to use the CPU once access is obtained. The class discussion attempts to get the students involved by relating occurrences of having just a couple of pages to copy but having to wait a long time because the person using the photocopier was copying tens or hundreds of pages. This is something that almost every student has experienced. The class readily comes to the conclusion that while this is a simple mechanism for organizing photocopier use it results in a lot of unhappy people waiting for service, which is an unproductive use of their time. This sets the stage for discussing ways to prevent someone from being a resource hog by monopolizing the use of the photocopier.

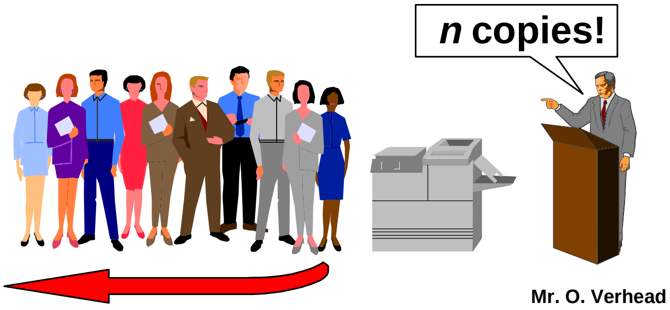

Figure 2. Representation of Round Robin scheduling algorithm.

The simplest method of allowing fair access to resources is the Round Robin algorithm. Illustrated in Figure 2, a person is allowed to use the photocopier once the head of the line is reached. However, there is now a “copier monitor” named O. Verhead (to indicate that the monitoring process introduces overhead into the computer system) that will allow the person using the photocopier to make at most a pre-determined number of copies. Once that limit is reached, the person is required by the monitor, who has been carefully counting the number of pages copied, to go back to the end of the line. In technical terms, the monitor preempted the user, thereby interrupting the current process. This requires the user to somehow bookmark the next page to be copied from the book so that the photocopying process can resume where it left off. One of the key things to discuss is to what the value of ***n*** - the maximum number of pages that can be copied before being preempted - should be set. If ***n*** is too small then the people in line waste a lot of time thrashing about in getting their materials set up for copying then having to put things away after just a few copies. If many pages need to be copied then this can result in spending a significant amount of time in preparing to use the photocopier, which takes away from actual photocopier use. If ***n*** is too large then the system degrades back to the FCFS algorithm and people in line will be forced to wait while someone is able to copy an entire book ahead of them. The class now discusses ways to handle this problem, which allows the instructor to introduce other scheduling methods.

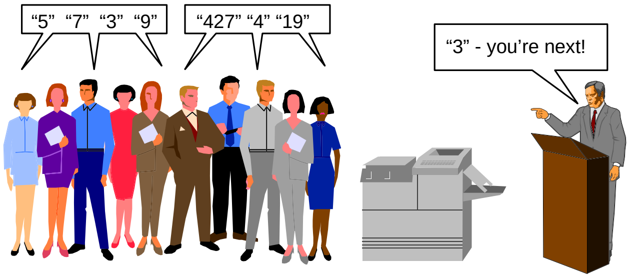

Figure 3. Representation of Shortest Job First scheduling algorithm.

With the Shortest Job First algorithm shown in Figure 3, the monitor checks the line each time the photocopier is available for use and asks each person how many pages he or she needs to copy. The monitor then selects the person specifying the fewest number of pages to be the next user of the photocopier. This method has the benefit of allowing persons who require just a few copies ready access to the photocopier. However, there are also problems. It is possible that someone might undercount the number of pages to be copied and unfairly be placed in front of other users. If a person has a large number of copies to make, that individual might never gain access to the photocopier if people with fewer pages to copy continue to enter the line. Once a person with a large number of copies gains access to the photocopier the machine is tied up until all of the copies are made; in the meantime, a line of increasingly annoyed users can form behind the person at the photocopier. This particular problem is addressed by the Shortest Remaining Time (SRT) algorithm, illustrated in Figure 4.

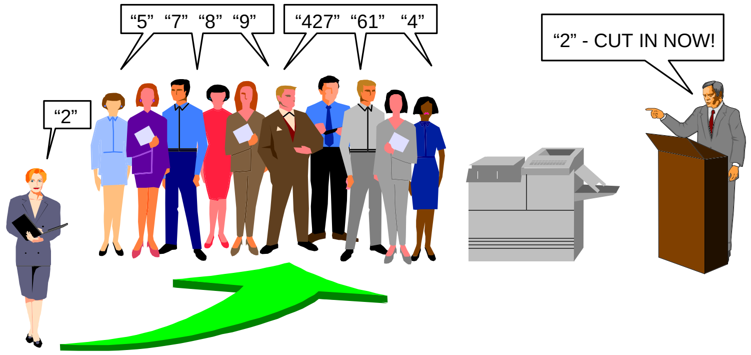

Figure 4. Representation of Shortest Remaining Time scheduling algorithm.

In the SRT algorithm the line is check by the monitor whenever a person joins the line. If that person has fewer pages to copy than the current user of the photocopier, then the current user is preempted and the new person is given access to the photocopier. While this successfully addresses the problem of several users being stuck behind someone doing a large copying job, the role of the monitor is now increased, thereby adding to the overhead. Additionally, it is still possible that some users will encounter what is known as starvation, which is when one is denied access to the resource because of continuously receiving low priority evaluations from the monitor. After some more class discussion, ways to guarantee that large copying jobs eventually receive photocopier access is presented.

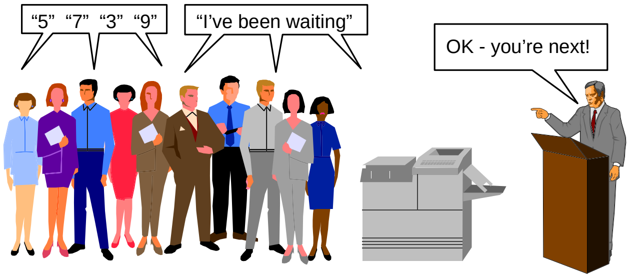

Figure 5. Representation of Highest Response Ratio Next scheduling algorithm.

The Highest Response Ratio Next (HRRN) algorithm shown in Figure 5 requires the monitor to keep track of both the length of wait ***w*** and the time needed to make copies ***s*** for each person in line. When the photocopier becomes available, the response ratio ***(w + s)/s*** is calculated for every user. The user with the highest response ratio is selected by the monitor to use the photocopier. While this method favors short copying jobs, by taking into account the waiting time the longer copying jobs will eventually receive access.

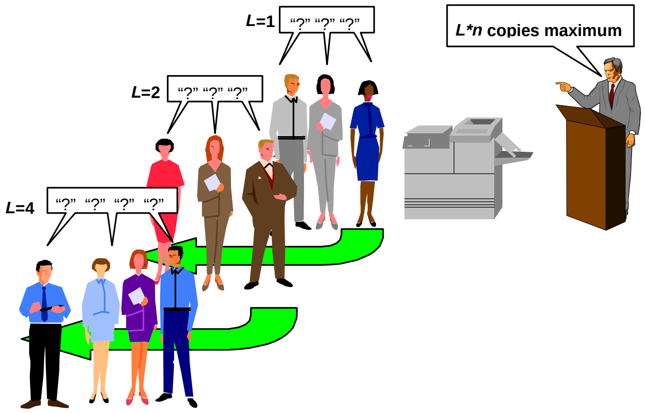

Figure 6. Representation of Multilevel Feedback scheduling algorithm.

The multilevel feedback algorithm shown in Figure 6 establishes priority based on the number of photocopies already made instead of an estimated total number of copies. This is done by creating a number of lines, each with their own priority level. When a person wants to use the photocopier, he or she is placed by the monitor into the highest priority line. The individuals in the line access the photocopier using the FCFS algorithm, but when they receive access to the photocopier a limit is placed on the number of copies that can be made at that time. When users are preempted the monitor directs them to enter the next lower priority line. Users in the lower priority lines can access the photocopier only when all of the higher priority lines are empty; however, once access is obtained they are able to make more photocopies in this session than they were in previous sessions. The ***L*** values in the diagram indicate the relative maximum number of copies allowed for a person in line to make when at the photocopier; the ***L*** values are set so as to prevent starvation. One of the benefits of this approach is that it simplifies the role of the monitor. There is no need to know how many copies each user wants to make; the line that a person comes from to use the photocopier will determine the number of copies that can be made. This reduces the role of the monitor to keeping track of the lines, starting with the one with highest priority, and allowing the first person found to have access to the photocopier.

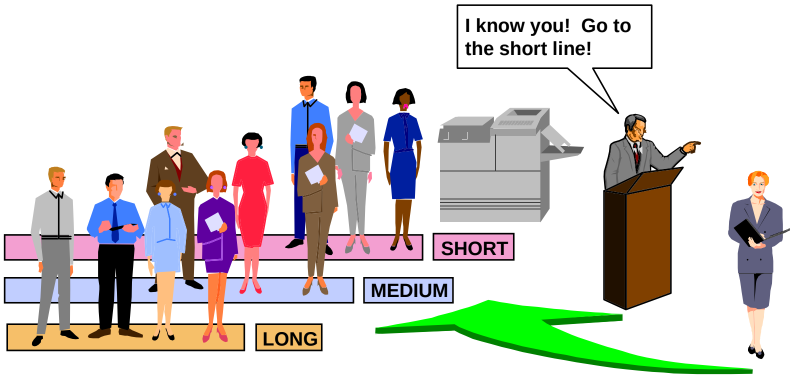

Figure 7. Representation of Fair-Share scheduling algorithm.

The Fair-Share algorithm shown in Figure 7 keeps a history of recent photocopier use by each user. The monitor grants priority to those individuals who historically make the fewest copies while other users are placed into lines that reflect their photocopier usage. In this example the users are separated into having short, medium, or long photocopying jobs. The lines are treated in a similar manner as that used in the multilevel feedback algorithm, with the exception that if a user is preempted by the monitor or makes fewer copies than expected, then the next time the person comes to use the photocopier the monitor can reassign the individual to a different line.

Now that the various scheduling algorithms have been introduced to the students, they can be revisited and explained using operating systems terminology. The associations that the students have been presented give them something to which to relate. This enables them to better understand what is transpiring with the dispatcher when it is attempting to select the next process to use the CPU and what the general benefits and drawbacks are for each algorithm. This method has been successfully used in our operating systems course.

JOHN K. ESTELL joined Bluffton College as an associate professor of computer science in 1996. He was previously an associate professor in the EECS Department at The University of Toledo. He received a BS (1984) degree in computer science and engineering from Toledo and received both his MS (1987) and PhD (1991) degrees in computer science from the University of Illinois. His areas of interest include program development tools, robotics and interface design. Dr. Estell is a member of ACM, ASEE, IEEE, Tau Beta Pi, and Eta Kappa Nu.