ATOPSS (ADAGE TUTORIAL OPERATING SYSTEM SIMULATOR): A COMPUTER GRAPHIC SIMULATION OF A DISCRETE TIME OPERATING SYSTEM FOR INTRODUCING ELEMENTARY CONCEPTS

William J. Tracz

The Pennsylvania State University

ABSTRACT

This paper introduces two films made under a discrete time operating system simulator that is programmed and run on an Adage 30 Graphics Terminal at The Pennsylvania State University.

Using the graphic hardware available on the system, the simulator displays the contents of all major operating system queues along with the contents of memory and active hardware. The observer can then watch the flow of information from one job step to another in the discrete time simulation of different operating system configurations.

Various parameters specified by the user allow the operating system simulator to operate under different hardware configurations and operating system strategies such as multiprogramming and multiprocessing, along with preemption and different memory allocation techniques.

INTRODUCTION

The Adage Tutorial Operating System Simulator is designed to provide both a learning tool and a teaching aid for the presentation of elementary operating system concepts to graduate or advanced undergraduate conputer science students. As a learning tool, the simulator itself offers real time interaction and observation of a parametric, discrete time event oriented operating system. For a teaching aid, a series of short movies has been produced using ATOPSS solely as its basis, demonstrating both the use of the simulator and the different concepts it portrays. These movies thus offer the portability necessary for in-classroom presentation and provide a basis for further discussion.

The main emphasis of this paper is on the tutorial techniques involved in the simulator along with the system components. Also included is a short narration covering the contents of the films to be presented in conjunction with this paper. It has been the philosophy throughout the designing, programming, and filming of ATOPSS to make the system simple, factual, and above all, visually effective for introducing elementary operating system concepts.

ATOPSS OPERATING SYSTEM CONCEPTS

The idea of discete time is often difficult for a student to grasp. Phrases like "multiprogramming", "multiprocessing", and "CPU preemption" can be confusing when presented to a student for the first time. ATOPSS clarifies these issues in two ways. First, it displays on the screen a multitude of possible hardware combinations (ATOPSS allows a variable number of CPU's, channels and 1K memory blocks). The viewer can, for example, select two CPU's for multiprocessing by pushing the appropriate button. Secondly, ATOPSS will simulate discrete time processing of the operating system/hardware configuration chosen by the user. Discrete time is handled by the clock which "skips" time between "events".

There are four "events" simulated by ATOPSS: Job arrival, I/O request, I/O termination and Job termination. Job arrival occurs when the user submits the jobs for the simulated operating system to process. Each job requires the following information: Job identification number; time (how long for which the job will run); priority (necessary to rank jobs in queues); memory size; I/O interval (how long the job will use a CPU before it needs the use of a channel); and I/O length (how long the job will use a channel before it needs the use of a CPU). This information is kept in the job queue element. When jobs are finished being read, all jobs, according to priority, are put in the initiation queue where they wait to be loaded into memory and placed on the ready job queue. The time at which a job is loaded into memory depends on the memory allocation technique, degree of multiprogramming, and the size of the job on the top of the initiation queue. The ready job queue holds jobs awaiting use of the CPU. When a CPU becomes available a job leaves the ready job queue and enters the CPU where one of two possible events will be scheduled and placed in the event queue. First, a job termination event is scheduled if the time remaining is less than the job's I/O interval (found in the Job Queue Element), otherwise an I/O request event is scheduled. When an I/O request event is processed, a job leaves the CPU to use a channel if one is free and an I/O termination event is scheduled. If a channel is not free, the job goes to the I/O wait queue. It should be noted that since all action takes place in discrete time, the clock remains the same during movement between queues and

only changes when a new event is initiated from the event queue.

ATOPSS and its films hold a distinct advantage over other operating system simulators in that they visually present the processing of jobs by a simulated operating system. This and other features of ATOPSS versatility is further demonstrated in the following sections.

ATOPSS VARIABLE SIMULATED HARDWARE CONFIGURATIONS

An ATOPSS user can construct various hardware configurations under which the simulator will run by simply pushing the appropriate "Function Switches" labelled by an overlay.

An effective method of presenting the various hardware configurations is accomplished by comparing the simplest with the most complex. Figure 1 shows the default hardware selection list; and Figure 2 the corresponding simulated hardware configuration. Figure 3 shows the user modified selection list and Figure 4 the appropriate hardware configuration.

Figure 1

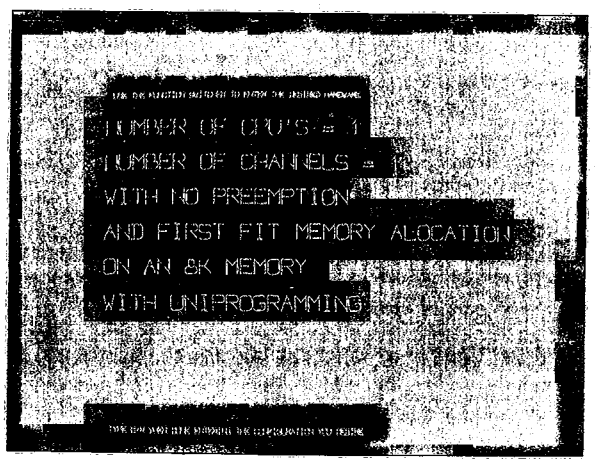


Figure 2

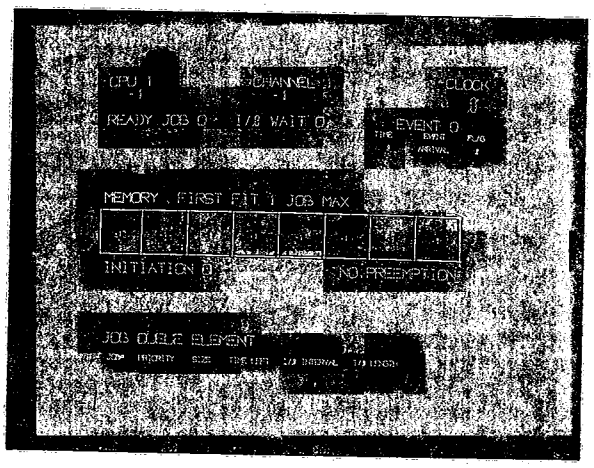


Figure 3

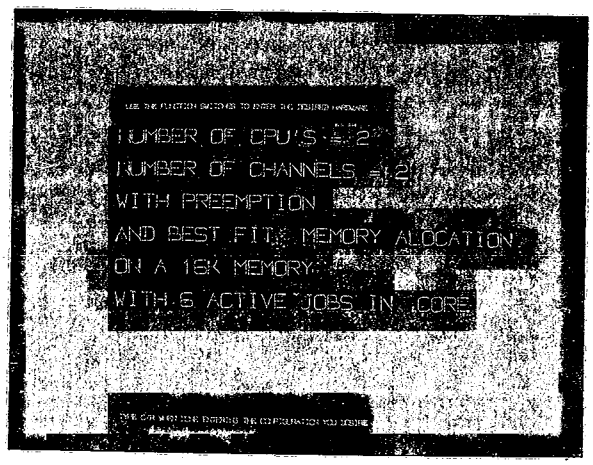
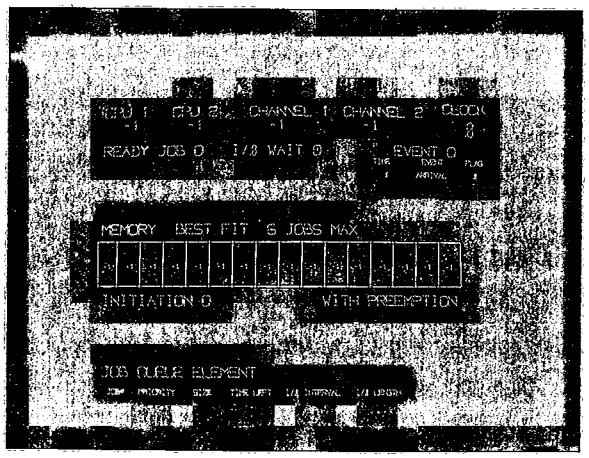


Figure 4



It is evident that the simulator offers the following choices: one or two CPU's; one or two channels; CPU preemption or no CPU preemption; 8K or 16K memory; first fit or best fit memory allocation; and uniprogramming contrasted with a limit of up to six jobs simultaneously being multiprogrammed.

The standard operating procedure consists of the participant first choosing an appropriate hardware configuration under which to run and, second, submitting jobs via the keyboard (at present the simulator is programmed to run only in batch mode; future releases will offer a job arrival rate parameter so that jobs may be submitted at a given rate during the simulation). ATOPSS then simulates the job processing in one of two modes-manual or automatic. In manual mode the viewer is allowed to proceed at his own pace by depressing carriage return on the keyboard to complete each job step. While in automatic mode, an evenly paced presentation of the processing is carried out in the operating system without any external assistance.

Other features displayed in Figure 4 are the queues: initiation queue, ready job queue, I/O wait queue, and event queue. The event queue is broken down further into 3 components: the time of the event, the type of event (job arrival, I/O request, I/O termination and job termination), and the event flag (showing which CPU or channel is to be acted upon).

ATOPSS SIMULATION DISPLAY

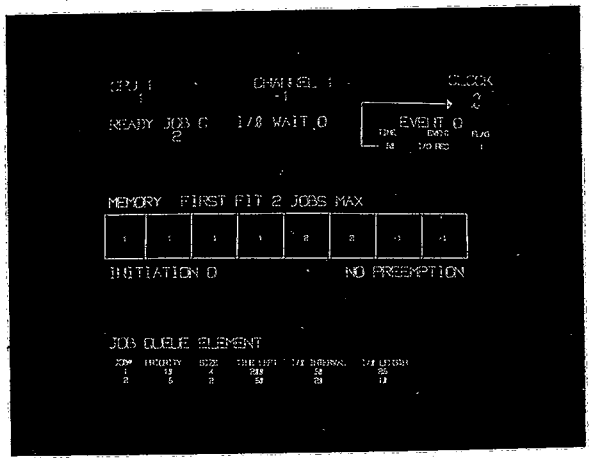
The run time display of ATOPSS is the main reason for its distinct advantage as an operating system simulator. Through successive suggestive images displayed on the screen, the observer is able to view the necessary decisions and the resulting actions occurring during the simulated processing of the operating system.

The decision making process is shown by highlighting certain queues, or hardware components and their contents. This occurs when through normal operating sequence, the availability or status of a queue or hardware device must be considered before effecting the next job step or action.

The results of these actions are shown in two ways: first through the use of "Flag Arrows" and secondly, through the changing of the contents of queues and hardware components. Another important graphic feature used by the simulator calls attention to both the discrete time and the event orientation of the system. Whenever an event is added to the event queue, it is emphasized by a flashing arrow that appears beside its position in the queue. A good example showing all concepts would be the I/O request event. This job step would proceed as follows:

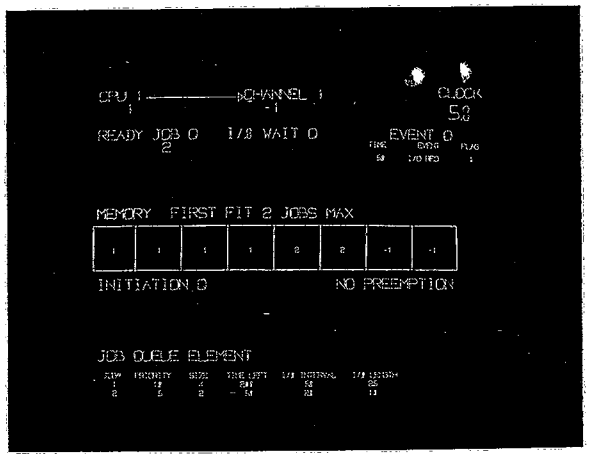
Step 1: The event queue is highlighted (indicating pending action).
Step 2: The clock is highlighted (indicating an upcoming change).
Step 3: A flow arrow appears between the clock and event queue (showing the flow of information and emphasizing the discrete time nature of the simulator (Figure 5)).

Figure 5



Step 4: The time for this event is transferred to the clock from the event queue time element.
Step 5: CPU 1 is highlighted.
Step 6: Channel 1 is highlighted.
Step 7: If channel 1 is empty (ie. its contents are -1) then go to Step 8, else go to Step 13.
Step 8: A flow arrow appears between CPU 1 and channel 1 (Figure 6).

Figure 6

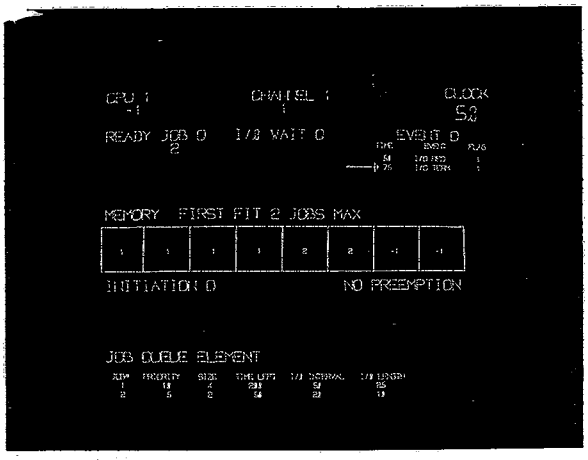


Step 9: The contents of CPU 1 is transferred to channel 1.
Step 10: I/O termination event is added to the event queue (calculated from the I/O

length in the Job Queue Element for the job now entering the channel).

Step 11: A flashing arrow points to the newly added event (Figure 7).

Figure 7

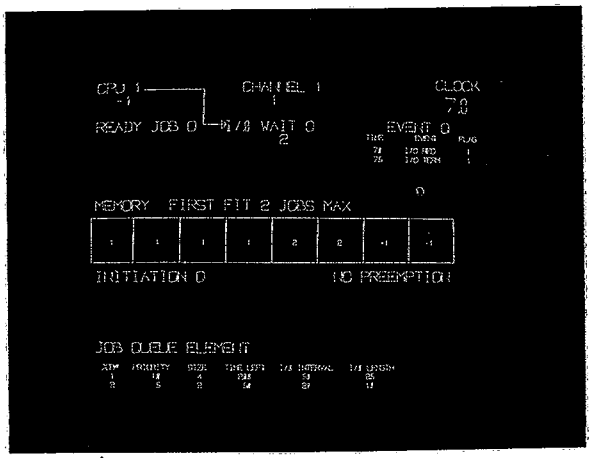


Step 12: The next event is processed.

Step 13: A flow arrow appears between CPU 1 and I/O wait queue.

Step 14: The contents of CPU 1 is transferred to the I/O wait queue (Figure 8).

Figure 8



Step 15: The next event is processed.

In summary, through the techniques outlined in this section, the author has attempted to preserve a tutorial attitude in the graphic presentation of the system and maintain a proper emphasis on actual operating system functions.

ATOPSS TUTORIAL FILMS

Two movies have been produced using ATOPSS as their basis and are to be shown in conjunction with this paper. The first movie is titled "ATOPSS Introduction to Operating System Concepts-Hardware Configurations". This movie shows the construction of different hardware configurations and the resulting operating systems generated by them. Particular attention is paid to the job flow within the simulator and the relation of the hardware components to the various queues. Also stressed is the event orientation of the system. The film is instructional in nature and is directed toward students with a marginal background in computer science.

The second film is titled "ATOPSS Introduction to Operating System Concepts-A Basic Simulation". This movie presents a complete simulation of jobs processed through normal execution of ATOPSS. The discrete time concept of the simulator in emphasized along with multiprocessing and multiprogramming. The film starts out by simulating a basic hardware configuration (one CPU and one channel) and works up to a more complex configuration (two CPU's and two channels). This film is also instructional in nature and, although designed as a sequal to the first, may be shown alone to audiences with some background in operating systems.

REMARKS

ATOPSS is a program written in Adage Fortran and run under the Amos 2 Monitor on an Adage 30 Graphics Terminal. Because of the dual processing nature of the machine, the graphic hardware and the image can be controlled independently of the executing main program. All figures furnished in this paper are taken directly from the screen using a Polaroid camera. All movies were filmed using a Maurer 16 mm camera.

The basic simulator presented here was derived from another operating system simulator studied in a graduate course given by the Computer Science Department at The Pennsylvania State University. ATOPSS is in its first release and is scheduled for major renovations such as dynamic priority revision along with improved job statistics in the Fall of 1973.