

Matthew Austin

Principles of Operations

Dr. Tian

29 March 2018

Trace-Drive Modeling

In this article, authors Sherman, Baskett, and Browne analyze CPU Scheduling in a multiprogramming system. When analyzing CPU scheduling, the microscopic level job stream data is obtained in an event driven software probe, which is used to drive a multiprogramming computers system. In order to get the information, an accurate replica of a production environment for the testing of variations in the system is needed, thus a technique called trace-driven modeling is used. The schedule methods tested are: round robin, first-come-first-served, dynamic predictors, etc. Out of these best methods, the one that is preemptive and prevents a job from holding CPU for too long will be successful.

The primary purpose of a multiprogrammed operating system is to maximize throughput by balancing the demand for processing and I/O facilities. Keeping in mind that the character of resource demand of a given job may fluctuate dramatically, thus making a valid comparison of any algorithm will require a large amount of labor. By comparing the effects of the are isolated and changes in the methods become very easy to make, an effectively comparison between the algorithms can be made. The data that is used on the comparison include factors such as CPU burst times and I/O service times. The job stream is then put through the model system with the set of algorithms that are being compared. The most successful scheduling algorithms are a pre-emptive mechanism and a bound on the CPU burst time.

In conclusion, a good scheduling method must be preemptive and must prevent any job from capturing the CPU for too long. Out of all the other methods, the data from the trace-driven model

showed that the performance of the round-robin is better while predictive methods were the worst.

Round-robin method is preferred in a system where fairness is important.