ECS150
WQ2004
Homework #1
Due: Thurs. 1/22, 4:45PM, HW Box in Kemper Hall

1. Write a formula to compute the cpu utilization for open shop. We define the **utilization** *u* as the fraction of time used for computation. Assume a typical job requires $r = 10$ seconds to read in from cards, $c = 3$ seconds to compute and $p = 30$ seconds to print the results on paper. Programmers sign up for 15-minute slots and run their programs twice per slot. For these values compute the utilization using your formula.

2. Write formula to compute the cpu utilization for same situation using operator-driven shop. Assume that an operator takes $s = 30$ seconds to remove the output from one job and set up the next job. There are enough jobs to ensure that the operator always has another job to start as soon as one finishes. For this value compute the utilization using your formula.

3. Write a formula to compute the cpu utilization for the same situation using offline transput. Assume it takes only 1/100 as long to read information from tape as from cards and only 1/100 as long to write information to tape as to paper. The resident monitor takes $s = 0.1$ seconds to reset the machine between jobs. The operator spends 60 seconds to unload and load tapes after every ten jobs. There are several offline computers, so there is no bottleneck reading jobs and printing results. For these values compute the utilization using your formula.

4. Write a formula to compute the utilization for the same situation using spooling. Assume that the computer has enough card readers and printers so that there are always jobs waiting to be run and printing is not a bottleneck. It takes only 1/100 as long to read or write information from or to disk as from or to cards or paper. The computer spends 1 percent of its time servicing interrupts for transput (I/O); this is not counted as computation on behalf of jobs. It takes $s = 0.01$ seconds to reset the machine between jobs.

5. An I/O-bound process is one that, if run alone, would spend more time waiting for I/O than using the processor. A processor-bound process is the opposite. Suppose a short-term scheduling algorithm (the algorithm that decides what process to run, among those capable of running, when a process performs an I/O operation or terminates) favors those processes that have used little processor time in the recent past. Explain why this algorithm favors I/O-bound processes and yet does not permanently deny processor time to processor-bound processes. A simple picture will help you with your answer.

6. Tanenbaum: Chapter 1, problem 3

7. Tanenbaum: Chapter 1, problem 6

8. Tanenbaum: Chapter 1, problem 8

9. Tanenbaum: Chapter 1, problem 9