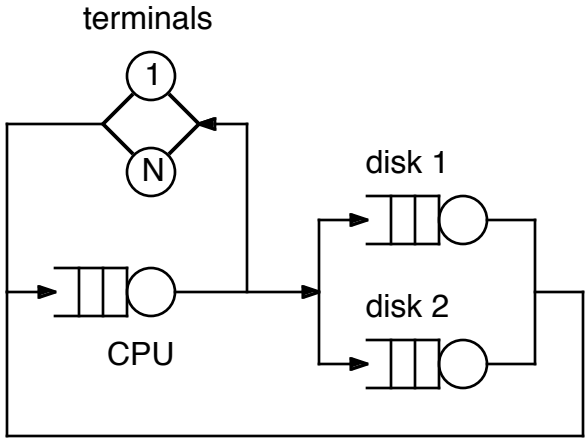
Project #1
Simulation of an Interactive Computer System

Due: Wednesday, Feb. 8, 2006

Write a Yacsim program to simulate the following models of an interactive, multiprogrammed computer system with two disks.



The system has a total of *N* jobs, one job associated with each terminal. A job that completes its "think time" at its terminal goes to the CPU for service.

You are to investigate two job routing strategies.

Strategy A: Each time a job completes service at the CPU, the job goes to disk 1 with probability 0.65, to disk 2 with probability 0.3, and back to its terminal with probability 0.05. The CPU, disk 1, and disk 2 can each service one job at a time, and each uses a FCFS, non-preemptive scheduling policy.

Strategy B: Each time a job completes service at the CPU, it returns to its terminal with probability 0.05. Otherwise, it goes to the disk with the smaller number of jobs already waiting or in service. In case of a tie, the job goes to disk 1 or disk 2 with probability 0.5 each.

The service time at the CPU has a skewed-left triangular distribution between 5 and 20 ms (see the text, Section 3.3, for a discussion of skewed-left triangular distributions). The disks have uniformly distributed service times between 0 and 30 ms. Think times are exponentially distributed with mean 1000 ms. Functions that generate random numbers for all of these distributions are found in Appendix C of the text. Run the simulation for 1,000,000 ms (1,000 seconds) of simulation time.

Turn in a project report with a cover page and three sections (plus an appendix). The cover page should state, on separate lines,

ELEC 428
Spring Semester 2006
Project 1

(date submitted)

(your name)

The body of the report should consist of the following sections, each clearly named:

1. An introduction that includes
   - A summary of the project objectives (write this, not as a lab report for a course, but as a research project)
   - A *brief* overview of the model – do not assume that someone has read this assignment handout
   - A summary of the major results

2. A discussion of the implementation of the project that includes a description of the model in sufficient detail that someone else who implemented the model based on this description would get the same results. Also, a brief discussion of your implementation of the model as a Yacsim program. The latter should include information on what kind of Yacsim constructs you used (resources, sempahores, etc.), how you used them, and how the simulation is initialized and terminated. Do not include names of variables you use.

3. A presentation of the results of the simulation. This must include a table for each strategy with the following format:

	cpu			disk 1			disk 2		
nterms	qlength	rtime	util	qlength	rtime	util	qlength	rtime	util
5									
10									
15									
20									
25									

qlength is the average number of jobs at the cpu or disk, including the job (if any) in service. *rtime* is the average total time a job spends at the cpu or disk each visit, including waiting time and time in service. *util* is the utilization over the entire simulation. *nterms* is N, the number of terminals. <u>All times should be in units of seconds</u>.

Also provide a single graph that plots the system response time (the average time spent by a job at the CPU and disks "per throughput") for each strategy, as a function of the number of terminals. On the same graph, show two sets of upper and lower asymptotes for response time, based on the demands in Strategy A and what you expect to be the demands in Strategy B.

Put the tables and graph in context with a brief and concise description of what they are and what the results they show indicate about the system's performance under each strategy. Explain the difference in system response times shown in the graph. If there are any results that you believe are

unexpected, point them out and explain them. If you believe that some of your results are wrong, state why you are suspicious and identify the probable source(s) of the error(s).

For this project, attach as an appendix your **well-commented** simulation program or programs (you may choose to have separate simulation programs for the two strategies).

The project will be graded as follows:

Correctness of simulation results:	40%
Asymptotic results:	10%
Report:	
Organization and completeness:	10%
Evaluation/explanation	20%
Quality*:	20%

Correctness of simulation results seems fairly obvious. Partial credit may be given when results are incorrect **IF** you point out those results that you believe are in error and explain why. (Of course, credit may be deducted if you point out results that you say are incorrect and are in fact correct.) See Evaluation/explanation.

Asymptotic results: Are the asymptotes/bounds on the graph of system response time correct? Assuming they are, they may go a long way towards helping you argue about the possible correctness of your results.

Organization and completeness of the report: Did you follow the directions regarding the project? Are all of the sections present, clearly delineated, and complete? Did you include the tables, graph, and appendix?

Evaluation/explanation: Does the text of the report provide all of the requested information? Is the explanation of the results reasonable? Did you adequately explain seemingly anomalous results? Did you fail to note suspicious or totally outrageous results? Basically, does the report demonstrate that you reflected on the results and came to justifiable conclusions about how to interpret them?

Quality relates to a number of objective and subjective aspects of the report: grammar, spelling, presentation (formatting), clarity, conciseness, and appropriateness for the intended audience. The audience you should target for this report is someone who knows what an interactive, multiprogrammed computer system is and who has a basic understanding of Yacsim, operational analysis, and performance evaluation in general. That is, your audience is someone like you. Think of how you would explain the model and your results to one of your classmates if she or he were not doing the same project.

All reports must be printed and spell-checked. The tables and graph must be computer-generated, although you may want to annotate or elaborate the graph by hand (those coloring skills you developed in kindergarten may come in handy, but any hand-written annotations should be made with precision and care). Graphs can be produced easily using a variety of tools, including spreadsheets (I recommend Excel) or tools such as Matlab or gnuplot on Owlnet. If you want to include a copy of the figure at the beginning of this document and don't know

how to generate one yourself, you will find GIF and JPEG versions in the course web pages. Drawing programs are also available on Owlnet.

As will always be the case in this class, clarity, correctness, and completeness, not volume, are what count in the project report.

You may compare your results for this project with those obtained by other students in the class, and you may help one another with any programming or Yacsim-related problems. You may not copy another student's program(s), results, or report. With this understanding, don't forget to

SIGN THE PLEDGE