

University of Jordan  
Computer Engineering Department  
CPE702  
Computer Performance Evaluation

Assignment 3  
Introduction to SimPack and Discrete-Event Simulation

Objective:

The objective of this experiment is to introduce you to the SimPack toolkit and discrete-event simulation. For more information about this toolkit, please refer to <http://www.cs.sunysb.edu/~algorithm/implement/simpack/implement.shtml>.

Brief Introduction:

SimPack is a toolkit that can be compiled using the GNU C compiler, which is designed to work under both the UNIX and the Windows operating systems. In this experiment, it is required that you experiment with a CPU-Disk simulation example program that uses SimPack's event scheduling routines. This program finds performance parameters by simulating 9 tasks running on a system with one CPU and 4 disks.

Procedure:

1. First you need to have GNU C compiler installed.
2. Next you need to install the SimPack toolkit and the CPU-Disk example from the course homepage. Download the file named Assignment3.zip and unzip it to your hard disk. Six files will be unzipped to a folder named simpack1. The file queuing.c contains the needed SimPack routines, the file cpudisk.c contains the model for the CPU-Disk problem, and the file readme.txt describes this problem and the SimPack routines. Please read this file thoroughly before proceeding.
3. The file makefile contains rules for compiling the C files and generating executable file named cpudisk.exe. When you execute the command **make**, these rules are executed and if there are no errors, the cpudisk.exe is generated.
4. Execute cpudisk.exe and observe its results.
5. After you study the files cpudisk.c and makefile, you need to solve the cpu-disk-disk simulation problem described below by performing the following procedure:
  - Create a new folder and copy the files makefile, queuing.c, and queuing.h to it.
  - Modify makefile by replacing every "cpudisk" by "cpudisk2".

- Create a new file named cpudisk2.c similar to the file named cpudisk.c to model the cpu-disk-disk problem.
- Compile the sources to generate cpudisk2.exe.
- Run cpudisk2.exe
- Submit the source code of cpudisk2.c and the results of running cpudisk2.exe

CPU-Disk-Disk Problem:

This is similar to the CPU-Disk Problem described in the class. The only difference is that each tour consists of one CPU computation, and two successive disk I/Os instead of one disk I/O.