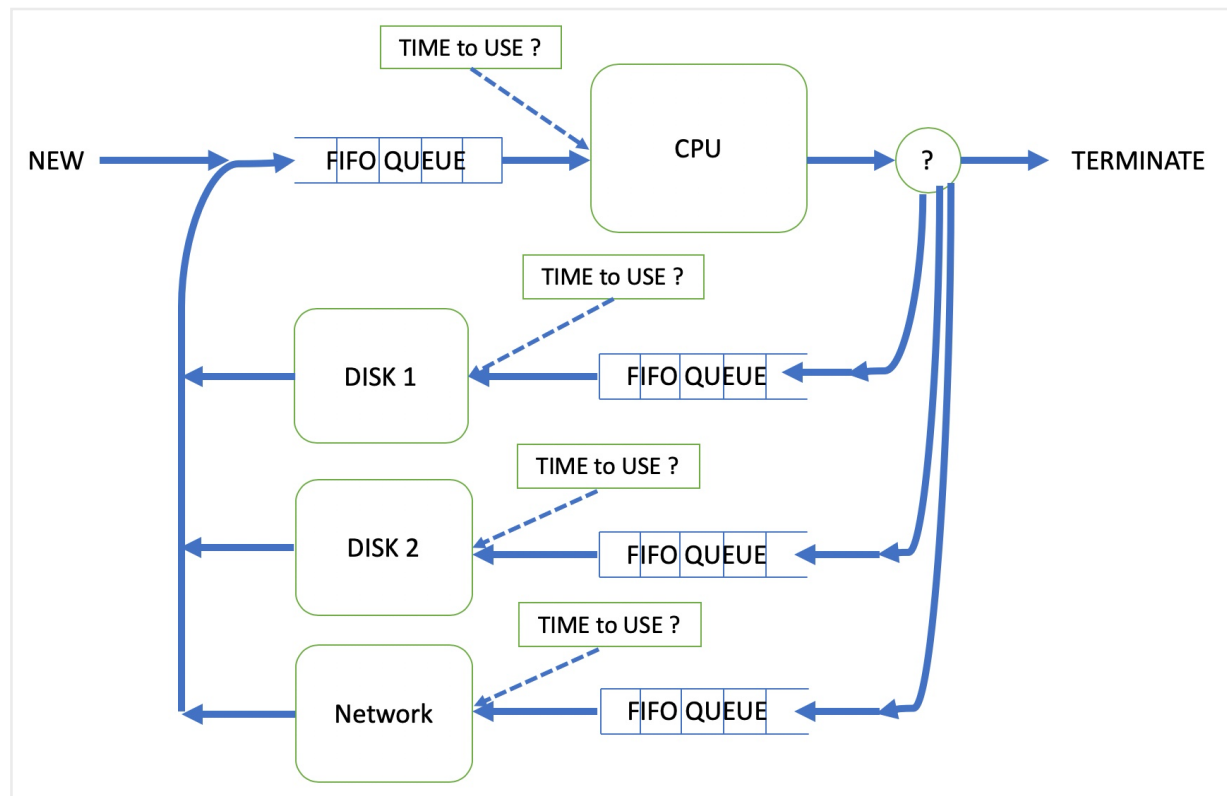


# Discrete Event Simulator

CIS 3207 Section 2

Lab 1

Dalerjon Murodov



This program is a simulation of the machine shown in the graph above. The machine models an operating system and stimulates how it allocates resources as jobs arrive to the CPU and the various states that they go through, from arrival to exit. As jobs enter the system, they will be placed in a FIFO queue and wait there until the CPU is free. When jobs enter the CPU, they will either then continue executing or exit the system. The probability of exiting the system will be given in the config file under the name QUIT\_PROB. The time that each job spends in the CPU will be randomly generated, and the lower and upper bounds will be defined in the config file. If the simulation decided to continue doing the job, then the job will either go to the network or disk. The probability that it goes to the network will be defined in the config file under NETWORK\_PROB. There is only one network, but there are two discs so the job will enter the disk which has the shorter queue, and will

go to disk1 if both queues are equal size. Disk1, disk2, and network will all have their own FIFO queue in order to keep track of which jobs need to be serviced first. The time that they spend in the disk/network will be a randomly generated number, and the upper and lower bounds will be defined in the config file. Afterwards, they will enter the CPU queue and the cycle repeats for each job until they exit. We will have a priority queue in order to keep track of all the events that happen. A priority queue is a data structure that can be used to keep track of elements that all have a different priority. In our case, the jobs that entered the the system first have the highest priority because they have been waiting the longest, so they will be ordered by the time when the entered the system. Jobs will be added to the priority queue every x seconds, where x is a randomly generated integer between ARRIVE\_MIN and ARRIVE\_MAX, both of which will be defined in the config file. The simulation will continue until FIN\_TIME has passed, which will also be defined in the config file.

The program is compromised of 3 helper files and 1 main file that brings everything together. The queue.c file defines what a queue is, and has two different types of push functions: one that pushes by priority, and one that uses FIFO. The config file defines various constants of the simulation, and the readConfig.c file helps read the config file. The main file is compromised of several functions. The CPU, DISK1/DISK2, and NETWORK are all defined as queues that can only have one thing inside at a time. There are 3 helper functions that make sure that jobs can leave the queue only if the CPU, DISK1/DISK2, or NETWORK is empty. randomNumberGenerator returns a random int between two integers, which it takes as a parameter.