Nicholas Gregorio – 100514374

Jason Runzer – 100520993

Albert Fung – 100520898

**Functions:** a variety of functions will be used to carry out these tasks. First, each queue will have push() and pop() functions which will control the processes in the queues. Fork() and exec() functions will be used to execute the process binary. The kill() function will be used in conjunction with this, to send signals such as SIGINT, SIGSTP, and SIGCONT. Parent processes will use the wait(&status) to ensure that the child process has been terminated before another process begins. In addition to these functions, the following functions will be implemented:

* A function with the purpose of determining available resources.
* A function to determine if the realtime queue is empty
* Functions that display processes being executed, suspended, completed, terminated, and/or processor time remaining
* Functions to read from the process file.

**Queues:** The HOST will contain one real-time queue, and three priority queues. These queues receive their processes from the dispatcher list. The realtime queue has priority, preempting any other active jobs running. This queue will run on a first-come-first-serve basis. The priority queues will dynamically move processes across levels (priorities) as their operations progress. However, the dispatcher still has the ability to accept jobs at lower priorities, emulating a round robin. At each dispatcher tick, the current process is suspended, to determine the priority of the new process.

**Memory Allocation Algorithm:**  a contiguous block of memory will be allocated to processes. The algorithm to allocate this memory will be first fit. The total memory available will be 1024 MB (1 GB), where 64 MB will be allocated to real time processes, with the remaining 960 MB to be allocated to the user-level priority queues.