# Interrupt Manager

**Author: Zachary Conner**

## Functions

* run()
  + Inputs: *void*
  + Outputs: vector<pcb\*>
  + Function Purpose: The run function is intended to be ran every execution cycle. It outputs a vector of pointers to job objects that have finished their current IO cycle.
* addToBlocked()
  + Inputs: pcb\*
  + Outputs: *void*
  + Function Purpose: This function takes in a pointer to a job object and adds the job to its blocked queue. It also initializes any job attributes that are related to interrupt tracking.
* getBlockedProcs()
  + Inputs: *void*
  + Outputs: vector<pcb\*>
  + Function Purpose: This is essentially a “getter” function for the blockedQueue variable. It simply outputs the current list of blocked processes.

## Data Structures Used

* Vector<pcb\*> blockedQueue
  + This holds all pointers to jobs that are currently in BLOCKED state. It is a private attribute.

## High-Level Description

The overall function of the Interrupt Manager is to hold and manipulate all jobs in the BLOCKED state. It is passed jobs when their IO requests begin, and it passes jobs back when their IO requests end.

## Low-Level Description

All data is stored in the blockedQueue vector which is manipulated by the run() and addToBlocked() functions. The architecture of the Interrupt Manager assumes that the run() function is ran every execution cycle. The run() function does the following:

1. Checks the blocked queue for any processes that have finished their IO request. It does this by comparing the currentIOCycle variable of the job to jobs current ioRequestLength.
   1. If any finished processes are found, it returns those processes.
2. Increments the current IO cycle count for any other processes that are blocked.

The addToBlocked() function does the following:

1. Takes in a process.
2. Sets the process state to BLOCKED.
3. Increments the nextIORequest attribute of the job.
4. Increments the currentIORequest attribute of the job.
5. Sets the currentIOCycle attribute of the job to 0.
6. Adds the job to the blockedQueue vector.