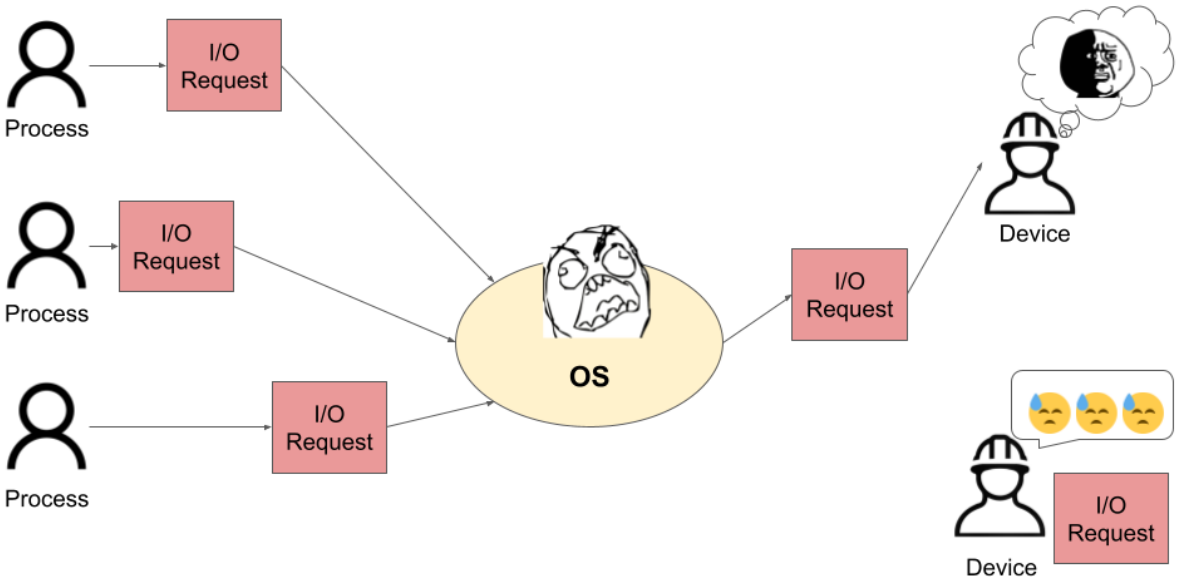
Project 1 of ITCS 343

Synchronization in Multi-threaded Programs

Due: March 13, 2022	**Team**: 1 - 3 members

Background



In a computer system, there can be a limited number of devices that can perform some I/O operations. For example, a printer can handle a printing job one at a time. Thus, some kind of policies must be in place to handle many I/O requests to a few devices. One of the common ways to manage this is to place all requests into a task queue where a device can take a request off and work on it one request at a time. When a task queue is full, there are different ways to handle this:

1. Let processes wait for an empty spot
2. Drop all incoming requests
3. Replace old requests with incoming ones

Instructions

In this project, you will implement a simulation of this situation. Your C program must simulate processes and devices on threads. Your program must take **inputs** in the command line for

1. A number of simulated processes
2. A number of simulated devices
3. A number of total requests
4. Minimum time to process a request in milliseconds
5. Maximum time to process a request in milliseconds
6. Action when the queue is full

We will assume that the simulated process will issue a request every 100 to 500 milliseconds at random. The devices will select a request by **FIFO policy**.

During the simulation, *your program can print anything that can help you debug and to show instructors that your program is working correctly*. By the end of the simulation, you should **output**

1. Average waiting time (time from a request is issued to the request is either dropped or started)
2. Percentage of dropped requests
3. Total time of the simulation

Hint: This is similar to the "producer-consumer" problem, but there are many producers and consumers.

Presentation

After the project is due, you will have to present your work with the following topics:

1. Introduction: who doing what part of the project.
2. Overview of your project
3. A case to show that your program works correctly
4. A comparison of the simulation in different settings
   a. Relax workload: a few processes and a lot of devices
      (wait vs drop vs replace)
   b. Heavy workload: a lot of processes and a few devices
      (wait vs drop vs replace)

   If you are working on the extra point, please also compare between the basic setting and the extra point setting.
5. Q&A

Criteria

Program (10 points)

1. [1pt] Create threads for the simulated processes and devices
2. [2pt] Use mutex and semaphore
3. [3pt] Implement all the wait, drop, and replace actions correctly.
4. [2pt] Synchronize between the queue, processes, and devices correctly
5. [2pt] Show the output correctly

Presentation (5 points)

1. [1pt] Demo works
2. [2pt] Show comparisons
3. [2pt] Q&A

Extra Points (2 points)

You may select ***one*** of the following extra conditions to get additional points:

1. [2pt] Change the devices’ policy from FIFO to SJF.
2. [2pt] Spin up one more device every time the queue is full.