Interactive CPU Scheduling

Alex Ekstrom – g42x664

Kade Pitsch – t12r458

Robert Jenko – g65n539

Introduction and Background

Our group wanted to focus our project on learning scheduling algorithms. The pros and cons of each algorithm, and how to implement them. We chose five different scheduling algorithms that we wanted to model at a high level through a web page.

First Come First Serve

- Simplest scheduling algorithm
- Executes each process based on order it arrives at CPU
- Performs much better for large processes
- Can create inefficient use between I/O and processor

Round Robin

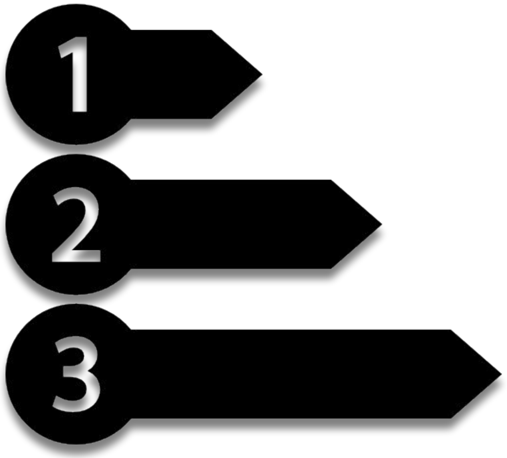
- Prevents short jobs from suffering like FCFS
- Allows equal run time between processes
- Uses FCFS to start each process
- Not very efficient

Shortest Job First

- Prevents short jobs from suffering like FCFS
- Process runs based upon expected processing time
- Small jobs receive priority
- Variability of response time is large

Priority Scheduling

- Executes processes based on assigned priority
- Allows most important processes to run first
- Not very efficient



Multi-Level Queue

- Combines multiple scheduling algorithms
- Executes queues based on priority
- Programmer chooses algorithm for each queue
- Processes can starve in lower priority queues

Implementation

For the back-end we chose a python framework called Django. This framework handles all the HTML templating and URL routing. The framework also gives us a lot of flexibility if we ever want to host the website. The frontend is written in HTML and JavaScript.

Website

Here is the demo portion.

We did not want to deploy the website so this link will only work for us during the presentation. Just a heads up if you're reading this after the fact. :)

Future Work

Possibly host our website through AWS so others can take a look at our work.

Resources

- Stallings, William. "Scheduling Algorithms." *Operating Systems Internals and Design Principles*, Ninth Edition, Pearson, 2018, pp. 430-452.