CMPSC 400
Operating Systems
Spring 2020

Participation 3 and 4: Comparing and Contrasting Scheduling Algorithms

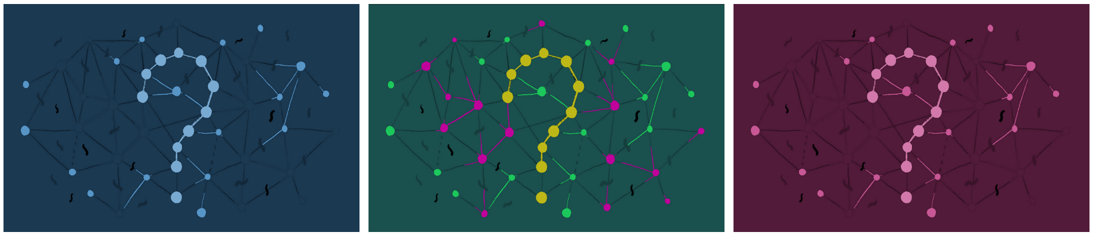


Figure 1: Each process-scheduling algorithm has different practical uses. Can you determine some of these applications in the context of best- and worst-case scenarios?

Summary

This task counts for two checks toward your grade.

Participation 3, writing for Tuesday:

In this participation, you are given the option of using emulators that demonstrate the running-time, performance and priority-systems of different process-scheduling algorithms. You are to discuss several selected algorithms in terms of its mechanism, advantages and disadvantages. In relation to the some of the others (listed below) you are to compare and contrast. The details of the task are listed below.

Participation 4, presenting for Thursday: For our next class on Thursday, please come prepared to present a *lightning-talk* of your analysis of your studied algorithms. The details for this task are below.

GitHub Starter Link

General Participation Repository
`https://classroom.github.com/a/S8lbI9Z5`

To use this link, please follow the steps below.

- Click on the link and accept the assignment.
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab.
- Clone this repository (bearing your name) and work on the practical locally.
- As you are working on your practical, you are to commit and push regularly. You can use the following commands to add a single file, you must be in the directory where the file is located (or add the path to the file in the command):

- `git commit <nameOfFile> -m ‘‘Your notes about commit here’’`
- `git push`

Alternatively, you can use the following commands to add multiple files from your repository:

- `git add -A`
- `git commit -m ‘‘Your notes about commit here’’`
- `git push`

Directory Instructions

We will be using a previous participation repository. Please copy the `03-04part/` directory from `classDocs` into your working participations repository. Please do not forget to push your work. We will not be using GatorGrader for this work.

Participation 3: What TODO

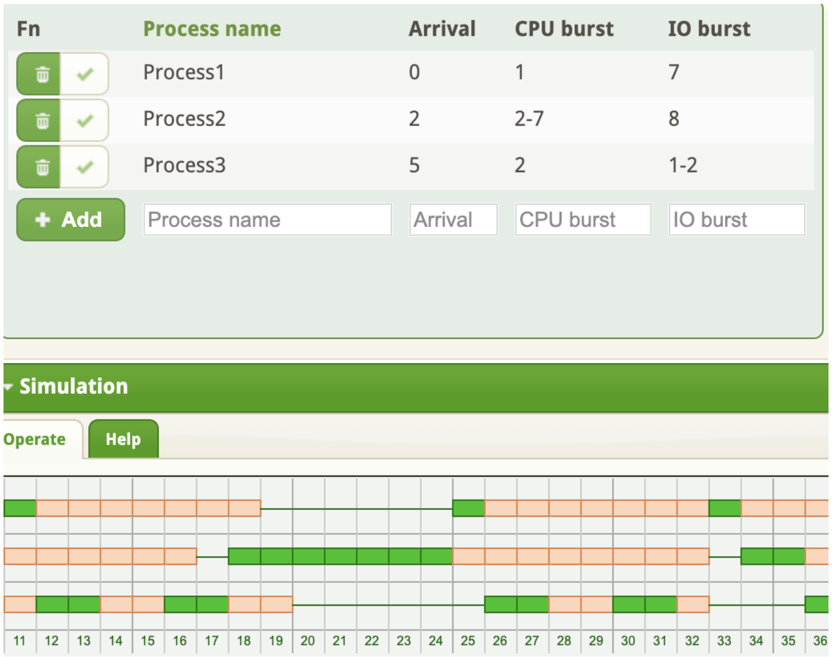
You are to select **three** of the above scheduling algorithms. For each, please study the algorithm using an emulator or code to be able respond to the below Questions in Blue. Your responses are to be written into your `writing/report.md` document. Your discussion should be written in clear and meaningful language.

1. Which three scheduling algorithms did you choose?
2. Using code or a Gantt chart, describe how each algorithm works. Offer a step-by-step diagram of functionality using sample processes.
3. What are advantages of the scheduling algorithm?
4. What are disadvantages of the scheduling algorithm?
5. Describe scenarios in which the algorithm will perform best?
6. Describe scenarios in which the algorithm will not perform well?

Algorithms

The algorithms of your study may be selected from the list below. If you cannot find an algorithm in your emulator, then you will have to conduct online research to locate code or another emulator to facilitate your study. If you choose another resource, please cite the resource.

- First-Come First-Serve Scheduling, FCFS
- Shortest-Job-First Scheduling, SJF
- Shortest Process Next (Knowledge), SPNK
- Shortest Process Next (Prediction), SPNP



Fn	Process name	Arrival	CPU burst	IO burst
	Process1	0	1	7
	Process2	2	2-7	8
	Process3	5	2	1-2
+ Add	Process name	Arrival	CPU burst	IO burst

Figure 2: A demonstration of scheduling algorithms by *AnimOS CPU-Scheduling*. We note the that processes and their run-times according to the scheduling algorithm.

- Shortest Remaining Time (Knowledge), SRTK
- Shortest Remaining Time (Prediction), SRTP
- Highest Response Ratio Next (Knowledge), HRRH
- Highest Response Ratio Next (Prediction), HRRP
- Feedback, F
- Fixed Priority (Non-preemptive), FP
- Priority Scheduling, PS
- Round Robin Scheduling, RR
- Virtual Round Robin, VRR
- Multilevel Queue Scheduling, MQS
- Multilevel Feedback-Queue Scheduling, MFQS
- Thread Scheduling, TS
- Multiple-Processor Scheduling, MPS
- Other type, Ot

Emulators

Since we are studying the performance of scheduling algorithms, an emulator is often very helpful to determine some of the basic uses of a particular algorithm. Below are two examples of good emulators that you may choose to help you explore your chosen scheduling algorithms. Note, the below list is certainly not inclusive and you are strongly encouraged to find other working emulators (or source code) online which may provide you with a better means to study the algorithms. As always, please be sure to cite your findings in your submission.

- CPU Scheduler Application
  - `http://jimweller.com/jim-weller/jim/java_proc_sched/`
- AnimOS CPU-Scheduling
  - `https://ess.cs.tu-dortmund.de/Software/AnimOS/CPU-Scheduling/`

Participation 4: What TODO

Figure 3: Come prepared on Thursday to discuss your findings during your *lightning-talk*.

Choose one your selected scheduling algorithms from Participation 3 to present to the class as a lightning talk (no more than five minutes and several slides in length). In your talk, you are to introduce the scheduling algorithm and discuss how it works using a Gantt diagram with sample processes.

Summery of Deliverables

1. Copy the directory called `03-04part/` from your `classdocs` into your `participations/` repository. Begin editing your files from the participation repository (and not in `classDocs`.) Do not forget to push your work.
2. File `writing/report.md`: In this file, all your questions-in-blue from above are to be answered.

3. Directory `src/*`: Add any code that you wrote or obtained from an online source to investigate your algorithm

4. Slides: Prepare your slides using a tool of your choosing to present your lightning talk on Thursday.