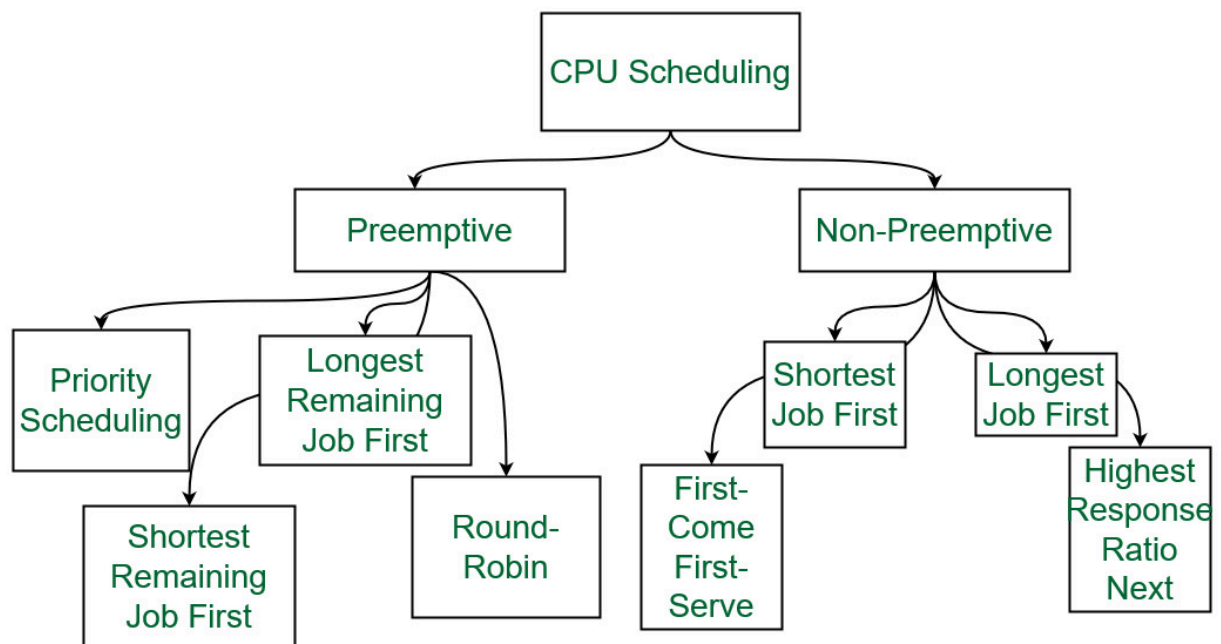


Group Project 4

CPU Scheduler

04/26/2024

CECS 326 - Operating Systems



Brandon Quach (#029133232)

Michael Kim (#025633788)

Project demo:

Link: <https://youtu.be/zGpc9t4hl44?feature=shared>

Project Description:

In this project, we created a program that implements the various process scheduling algorithms. The scheduler will receive a predetermined list of tasks using .txt files and organize them according to the chosen scheduling algorithms. Each task will include an assigned priority and CPU burst just like discussed in class. The priorities will range from 1 to 10 where a higher value indicates a higher relative priority. For round-robin scheduling however, the length of the time quantum will be 10 ms. The project will involve the following scheduling algorithms:

- *First-come, first-served (FCFS)*, which schedules tasks in the order in which they request the CPU.
- *Priority scheduling*, which schedules tasks based on priority.
- *Round-robin (RR) scheduling*, where each task is run for a time quantum (or for the remainder of its CPU burst).

Steps:

1. Download the Algorithm.java, CPU.java, FCFS.java, RR.java, Priority.java, Task.java, Driver.java, and schedule.txt to any folder to your choosing
2. Navigate to the folder using cd where your files are saved and compile them using these commands
 - a. `javac Algorithm.java`
 - b. `javac CPU.java`

- c. javac FCFS.java
 - d. javac RR.java
 - e. javac Priority.java
 - f. javac Task.java
 - g. javac Driver.java
3. Use the command "java Driver [algorithm]" to have the program run the algorithm with a predetermined .txt file.
- a. Replace [algorithm] with FCFS, RR or PRI for the following algorithms:
 - i. FCFS = First Come, First Served
 - ii. RR = Round Robin
 - iii. PRI = Priority

Code analysis:

Schedule function which invokes the process scheduler:

```
// Tasks are executed in the order they were added
@Override
public void schedule() {
    while (!queue.isEmpty()) {
        // Selects the new task from top of queue
        Task nextTask = pickNextTask();
        // CPU.java simulates the task
        CPU.run(nextTask, nextTask.getBurst());
        // Status message to inform task has been completed
        System.out.println("Task " + nextTask.getName() + " completed." + "\n");
        // Completed task is removed from queue
        queue.remove(nextTask);
    }
}
```

pickNextTask() function that determines the tasks selected for execution:

```
40     @Override
41     public Task pickNextTask() {
42         // First task is selected in queue
43         return queue.get(index:0);
44     }
45
```

run() function invokes the execution in CPU.java file:

```
12     public class CPU {
13         // Run the task for the given time.
14         public static void run(Task task, int time) {
15             System.out.println("Running task " + task);
16         }
17     }
```

Interface for handling each CPU scheduler:

```
12     public interface Algorithm
13     {
14         // Calls the scheduler
15         public abstract void schedule();
16
17         // Next task using the chosen scheduling algorithm
18         public abstract Task pickNextTask();
19     }
```

Error handling for program:

```
18 // Error handling
19 Run | Debug
20 public static void main(String[] args) throws IOException {
21     if (args.length != 2) {
22         System.err.println(x:"Usage: java Driver <algorithm> <schedule>");
23         System.exit(status:0);
24     }
25     BufferedReader inFile = new BufferedReader(new FileReader(args[1]));
26
27     String schedule;
28
29     // Array list for the queue of tasks
30     List<Task> queue = new ArrayList<Task>();
31
32     // Opens text file to read in the tasks and initialize the ready queue
33     while ((schedule = inFile.readLine()) != null) {
34         String[] param = schedule.split(regex:"\\s*");
35         queue.add(new Task(param[0], Integer.parseInt(param[1]), Integer.parseInt(param[2])));
36     }
37
38     // Closes the file
39     inFile.close();
40
41     Algorithm scheduler = null;
42     String algchoice = args[0].toUpperCase();
```

Switch casing for each CPU scheduler:

```
45     switch(algchoice) {
46         // First-come, First-served
47         case "FCFS":
48             scheduler = new FCFS(queue);
49             break;
50         // Priority scheduling
51         case "PRI":
52             scheduler = new Priority(queue);
53             break;
54         // Round-robin
55         case "RR":
56             scheduler = new RR(queue);
57             break;
58         // Default case
59         default:
60             System.err.println(x:"Invalid algorithm. Please try again.");
61             System.exit(status:0);
62     }
```

Program output:

FCFS scheduling output:

```
PS C:\Users\brand\Desktop\CECS326\Project4> java Driver FCFS schedule.txt
Running task Name: T1
Tid: 0
Priority: 4
Burst: 20

Task T1 completed.

Running task Name: T2
Tid: 1
Priority: 3
Burst: 25

Task T2 completed.

Running task Name: T3
Tid: 2
Priority: 3
Burst: 25

Task T3 completed.

Running task Name: T4
Tid: 3
Priority: 5
Burst: 15

Task T4 completed.

Running task Name: T5
Tid: 4
Priority: 5
Burst: 20

Task T5 completed.

Running task Name: T6
Tid: 5
Priority: 1
Burst: 10

Task T6 completed.

Running task Name: T7
```

```
Running task Name: T7  
Tid: 6  
Priority: 3  
Burst: 30
```

Task T7 completed.

```
Running task Name: T8  
Tid: 7  
Priority: 10  
Burst: 25
```

Task T8 completed.

Round Robin Scheduling Output:

```
PS C:\Users\brand\Desktop\CECS326\Project4> java Driver RR schedule.txt  
Running task Name: T1  
Tid: 0  
Priority: 4  
Burst: 20  
  
Running task Name: T2  
Tid: 1  
Priority: 3  
Burst: 25  
  
Running task Name: T3  
Tid: 2  
Priority: 3  
Burst: 25  
  
Running task Name: T4  
Tid: 3  
Priority: 5  
Burst: 15  
  
Running task Name: T5  
Tid: 4  
Priority: 5  
Burst: 20  
  
Running task Name: T6  
Tid: 5  
Priority: 1  
Burst: 10  
  
Task T6 finished.  
  
Running task Name: T7  
Tid: 6  
Priority: 3  
Burst: 30  
  
Running task Name: T8  
Tid: 7  
Priority: 10  
Burst: 25  
  
Running task Name: T1
```

Running task Name: T1
Tid: 0
Priority: 4
Burst: 10

Task T1 finished.

Running task Name: T2
Tid: 1
Priority: 3
Burst: 15

Running task Name: T3
Tid: 2
Priority: 3
Burst: 15

Running task Name: T4
Tid: 3
Priority: 5
Burst: 5

Task T4 finished.

Running task Name: T5
Tid: 4
Priority: 5
Burst: 10

Task T5 finished.

Running task Name: T7
Tid: 6
Priority: 3
Burst: 20

Running task Name: T8
Tid: 7
Priority: 10
Burst: 15

Running task Name: T2
Tid: 1
Priority: 3


```
Running task Name: T2
Tid: 1
Priority: 3
Burst: 5

Task T2 finished.

Running task Name: T3
Tid: 2
Burst: 5

Task T3 finished.

Running task Name: T7
Tid: 6
Priority: 3
Burst: 10

Task T7 finished.

Running task Name: T8
Tid: 7
Priority: 10
Burst: 5

Task T8 finished.
```

Priority scheduler Output:

```
PS C:\Users\brand\Desktop\CECS326\Project4> java Driver PRI schedule.txt
Running task Name: T8
Tid: 7
Priority: 10
Burst: 25

Task T8 finished.

Running task Name: T4
Tid: 3
Priority: 5
Burst: 15

Task T4 finished.

Running task Name: T5
Tid: 4
Priority: 5
Burst: 20

Task T5 finished.

Running task Name: T1
Tid: 0
Priority: 4
Burst: 20

Task T1 finished.

Running task Name: T2
Tid: 1
Priority: 3
Burst: 25

Task T2 finished.

Running task Name: T3
Tid: 2
Priority: 3
Burst: 25

Task T3 finished.

Running task Name: T7
Tid: 6
Priority: 3
Burst: 30

Task T7 finished.

Running task Name: T6
```

```
Running task Name: T6
Tid: 5
Priority: 1
Burst: 10

Task T6 finished.
```

What each individual did:

For this project we kept the roles the same to avoid any confusion.

Brandon: Brandon took the role in the completion of the lab report and ensuring that all project deliverables were included. In addition to his lab report responsibilities, he also worked on the code with Michael and input the proper comments to show the design.

Michael: Michael took a technical role within the group, with a primary focus on both software. His responsibilities encompassed software development, creation, and software components. In addition to his software responsibilities, he also focused on the video.

Work split was mostly 50/50 when it came to the software and its deliverables.

References:

<https://www.geeksforgeeks.org/cpu-scheduling-in-operating-systems/>

<https://www.geeksforgeeks.org/program-for-fcfs-cpu-scheduling-set-1/>