CS 350: Homework #6

Due 3PM, Tuesday, March 22

Kristel Tan (ktan@bu.edu)

March 14, 2016

Question 1

Part a

The Highest-Slowdown-Next (HSN) scheduler is "unfair" for I/O-bound jobs because it schedules events based on their slowdown, which we know to be the ratio between the job's expected turnaround time upon immediate service and its length. HSN's queue prioritizes jobs with the maximum slowdown first rather than their length. Hence, it does not necessarily prefer jobs with a shorter length. In fact, a job with a shorter length may very well get scheduled before a longer job. For example, even if job A has a longer job length than job B, A will have scheduling priority as long as it has been waiting longer than B.

Part b

Preemption at job arrival times would not be a good solution for an HSN scheduler because jobs would continuously be put on hold as more jobs came in. A preemptive scheduler would stop the current job being served and move on to the next job with the highest remaining slowdown every time a new job entered the system. Not only would this be an expensive process to do, but this would also increase the wait time of the stopped jobs as well as their slowdown ratio. Affecting the slowdown of these jobs in this way would hinder the fidelity of the system and contradict the scheduler's main function of serving jobs with the highest slowdown.

Part c

Preemption using a fixed timeout is a better solution for CPU-bound jobs in the case that we choose a quantum that is larger than the CPU burst for I/O-bound jobs. This is because CPU-bound jobs are more likely to finish their quantum where's I/O-bound ones are not. In other words, a CPU-bound job is likely to monopolize the CPU until it is preempted by the scheduler, whereas an I/O-bound job is likely to surrender the CPU because it blocks for I/O. This would again be problematic for I/O-bound jobs.

Part d

In the preemptive models of the HSN scheduler, we have seen that I/O-bound jobs tend to suffer the most, typically receiving less preference. To "fix" these issues, we can still use preemption with a fixed timeout; however, we should keep track of whether or not the quantum of a process has been used up or not. If the quantum has expired, than the associated job's priority can be considered low since it needs a new quantum and has use up all of its allocated time slice. If the quantum has not expired, then the job should be allowed to run with higher priority than jobs whose quantum has expired. This will ensure that I/O-bound jobs are serviced fairly alongside their rival CPU-bound jobs.

Question 2

Part a

The protocol given in this problem is buggy because it does not guarantee mutual exclusion. Each process is identical in that they only check for the other process's flag once with an 'if' statement instead of continuously checking the flag with a 'while' loop. This while loop is important because it tells a process to wait for the resource being used by the other process. The following instructions are example of when two processes P0 and P1 may both end up in the critical section:

Assume turn = 1 and flag[1] = false...

`flag[0] = true`	
`if (flag[1])`	
	`flag[1] = true`
	`if (flag[0])`
`Enter critical section`	
`turn := 0`	
	`if (turn == 0)`
	`Enter critical section`

Part b

```
Process P0:
repeat
  flag[0]:=true;
  while flag[1]{
    if (turn==1) {
      flag[0]:=false;
      while (turn==1) {};
      flag[0]:=true;
    }
  }
  Critical Section
  turn:=1;
  flag[0]:=false;
     Remainder Section
forever

Process P1:
repeat
  flag[1]:=true;
  while flag[0]{
    if (turn==0) {
      flag[1]:=false;
      while (turn==0) {};
      flag[1]:=true;
    }
  }
  Critical Section
```

```
  turn:=0;
  flag[1]:=false;
     Remainder Section
forever
```

Question 3

Part a

Please find the source code for this problem in the Java file named 'Q3A.java'. The code has been commented for clarity of what each method is doing. The following is a sample output of the program in which the two processes are interleaved:

```
Thread 1 is starting iteration 0
Thread 0 is starting iteration 0
We hold these truths to be self-evident, that all men are created equal,
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 0

Thread 0 is starting iteration 1
Thread 1 is done with iteration 0

We hold these truths to be self-evident, that all men are created equal,
Thread 1 is starting iteration 1
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 1

Thread 0 is starting iteration 2
Thread 1 is done with iteration 1

We hold these truths to be self-evident, that all men are created equal,
Thread 1 is starting iteration 2
that they are endowed by their Creator with certain unalienable Rights,
We hold these truths to be self-evident, that all men are created equal,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 2

that they are endowed by their Creator with certain unalienable Rights,
Thread 0 is starting iteration 3
that among these are Life, Liberty and the pursuit of Happiness.
We hold these truths to be self-evident, that all men are created equal,
Thread 1 is done with iteration 2

that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is starting iteration 3
Thread 0 is done with iteration 3

Thread 0 is starting iteration 4
```

```
We hold these truths to be self-evident, that all men are created equal,
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 4

that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 3

Thread 1 is starting iteration 4
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 4
```

As you can see, it is difficult to tell which output belongs to which process (*i.e. the output is incomprehensible*).

Part b

Please find the source code for this problem in the Java file named 'Q3B.java'. The code has been commented for clarity of what each method is doing. The following is a sample output of the program in which the two processes are interleaved:

```
Thread 0 is starting iteration 0
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 0

Thread 1 is starting iteration 0
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 0

Thread 0 is starting iteration 1
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 1

Thread 1 is starting iteration 1
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 1

Thread 1 is starting iteration 2
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 2

Thread 1 is starting iteration 3
```

```
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 3

Thread 1 is starting iteration 4
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 4

Thread 0 is starting iteration 2
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 2

Thread 0 is starting iteration 3
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 3

Thread 0 is starting iteration 4
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 4
```

As you can see, the print statements for a process in the critical section are now together for a single iteration. By implementing Dekker's algorithm, the output becomes more comprehensible.

Part c

Please find the source code for this problem in the Java file named 'Q3C.java'. The code has been commented for clarity of what each method is doing. The following is a sample output of the program in which the two processes are interleaved:

```
Thread 0 is starting iteration 0
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 0

Thread 1 is starting iteration 0
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 0

Thread 0 is starting iteration 1
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 1

Thread 1 is starting iteration 1
```

```
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 1

Thread 1 is starting iteration 2
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 2

Thread 1 is starting iteration 3
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 3

Thread 1 is starting iteration 4
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 4

Thread 0 is starting iteration 2
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 2

Thread 0 is starting iteration 3
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 3

Thread 0 is starting iteration 4
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 4
```

As you can see in the highlighted sections of the output, there are a few times when Thread 1 manages to get into the critical section even although it had just exited the critical section. This bug happens due to an 'if' statement in the code that causes Thread 0 to be delayed significantly longer.

Part d

Please find the source code for this problem in the Java file named 'Q3D.java'. The code has been commented for clarity of what each method is doing. The following is a sample output of the program in which the two processes are interleaved:

```
Thread 0 is starting iteration 0
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 0
```

```
Thread 1 is starting iteration 0
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 0

Thread 0 is starting iteration 1
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 1

Thread 1 is starting iteration 1
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 1

Thread 0 is starting iteration 2
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 2

Thread 1 is starting iteration 2
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 2

Thread 0 is starting iteration 3
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 3

Thread 1 is starting iteration 3
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 3

Thread 0 is starting iteration 4
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 0 is done with iteration 4

Thread 1 is starting iteration 4
We hold these truths to be self-evident, that all men are created equal,
that they are endowed by their Creator with certain unalienable Rights,
that among these are Life, Liberty and the pursuit of Happiness.
Thread 1 is done with iteration 4
```

As you can see, the output of this implementation has become identical to that of part (b) although the code still implements the prolonged delay of Thread 0 that caused the bug in part (c). Using Peterson's algorithm here, however, ensures that the same scenario does not occur.

Question 4

Please find the source code for this problem in the Java file named 'Q4.java'. The code has been commented for clarity of what each method is doing. The program writes the average busy waiting time value for each thread to a text file. The following is a sample output of running the experiment 10 times:

```
Experiment 0
Thread 0: 4.40219914E7 | Thread 1: 2.77301682E7

Experiment 1
Thread 0: 8.10850262E7 | Thread 1: 1.17391E7

Experiment 2
Thread 0: 1.112257044E8 | Thread 1: 7.3331007E7

Experiment 3
Thread 0: 6.4191284E7 | Thread 1: 1.09921292E7

Experiment 4
Thread 0: 4.50625566E7 | Thread 1: 3.11753042E7

Experiment 5
Thread 0: 5.83626102E7 | Thread 1: 1.92622284E7

Experiment 6
Thread 0: 9.57362654E7 | Thread 1: 6.37412388E7

Experiment 7
Thread 0: 6.84922346E7 | Thread 1: 1.27517724E7

Experiment 8
Thread 0: 9.18742374E7 | Thread 1: 3.54051792E7

Experiment 9
Thread 0: 9.02587406E7 | Thread 1: 7.22487422E7
```

The mean counter based on these experiments for Thread 0 is about 6.502×10^7 . The standard deviation for Thread 0 is about 2.522×10^7. This makes the 95% confidence interval 4.698×10^7 to 8.306×10^7.

The mean counter based on these experiments for Thread 1 is about and about 3.584×10^7. The standard deviation for Thread 1 is about 2.495×10^7. This makes the 95% confidence interval 1.799×10^7 to 5.369×10^7.