Queue Applications

Discrete Event Simulation

- **Single Queue, single server**
  - Rear → Front
  - $h$

- **Single Queue, multiple servers**
  - Rear → Front
  - $h$
  - $h$

- **Multiple Queue, multiple servers**
  - Rear → Front
  - $h$
  - $h$
  - $h$
Example: Single Queue, Single Server

- Arrival process
  - How customers arrive: What is inter-arrival time? E.g. between 1-3 min
  - Service mechanism: How long will service take? E.g. 0.5 to 2.0 min
  - Queue characteristics: FIFO

Example Data

<table>
<thead>
<tr>
<th>Customer</th>
<th>Inter-arrival Time</th>
<th>Service Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>1.9</td>
<td>1.7</td>
</tr>
<tr>
<td>C2</td>
<td>1.3</td>
<td>1.8</td>
</tr>
<tr>
<td>C3</td>
<td>1.1</td>
<td>1.5</td>
</tr>
<tr>
<td>C4</td>
<td>1.0</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Queue Simulation

<table>
<thead>
<tr>
<th>T</th>
<th>Arrival</th>
<th>Queue</th>
<th>Server</th>
<th>Depart</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>Idle</td>
<td></td>
</tr>
<tr>
<td>1.9</td>
<td>C1</td>
<td>[]</td>
<td>C1</td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>C2</td>
<td>[C2]</td>
<td>C1</td>
<td></td>
</tr>
<tr>
<td>3.6</td>
<td></td>
<td>[]</td>
<td>C2</td>
<td>C1</td>
</tr>
<tr>
<td>4.3</td>
<td>C3</td>
<td>[C3]</td>
<td>C2</td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>C4</td>
<td>[C4, C3]</td>
<td>C2</td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td></td>
<td>[C4]</td>
<td>C3</td>
<td>C2</td>
</tr>
<tr>
<td>6.9</td>
<td></td>
<td>[]</td>
<td>C4</td>
<td>C2</td>
</tr>
<tr>
<td>7.8</td>
<td></td>
<td>[]</td>
<td>C4</td>
<td></td>
</tr>
</tbody>
</table>
Application: Lab Printer Simulation

☐ There is one printer in the Computer Science Lab
☐ At any given time, there may be as many as 10 students working in the lab
☐ Each student may print up to twice in an hour
☐ Print jobs are 1-20 pages long
☐ ∴ There are up to 20 print jobs in an hour
☐ Question: What is the chance that in any given second there will be a print job scheduled?

Application: Lab Printer Simulation

☐ There is one printer in the Computer Science Lab
☐ At any given time, there may be as many as 10 students working in the lab
☐ Each student may print up to twice in an hour
☐ Print jobs are 1-20 pages long
☐ ∴ There are up to 20 print jobs in an hour
☐ Question: What is the chance that in any given second there will be a print job scheduled?

\[
\frac{20}{1 \text{ hour}} \cdot \frac{1 \text{ hour}}{60 \text{ min}} \cdot \frac{1 \text{ min}}{60 \text{ sec}} = \frac{20}{3600} = \frac{1 \text{ task}}{180 \text{ sec}}
\]
Application: Lab Printer Simulation

- There is one printer in the CS Lab (10 ppm)
- At any given time, there may be as many as 10 students working in the lab
- Each student may print up to twice in an hour
- Print jobs are 1-20 pages long
- Therefore, there are up to 20 print jobs in an hour
- Question: What will the average wait time be for students to receive their printouts?
- Question: What would the average wait time be if the printer were upgraded to 20 ppp?

Lab Printer Simulation

- Need representation of a task:
  - time stamp – time when the print job arrives
  - # pages - # pages in the print task
- Need a printer queue
  - keeps track of print jobs
Simulation: Main Loop

for each tick (second)
1. Is there a new task? 1/180 chance!
   if so, create a new task and add it to the printer queue

2. Do we need to schedule a new task on printer?
   if printer is idle and there is a task on printer queue
      a. remove next task from printer queue and assign it to printer
      b. record wait time for this task [current time – time stamp]
      c. how long will task take?

3. Printer prints for 1 second
   a. take off 1 sec from task time
   b. if task completed, set printer to idle

Finally, compute average wait time & print results

Modeling the Task & Printer

<table>
<thead>
<tr>
<th>Task</th>
<th>Printer</th>
</tr>
</thead>
</table>
| - int timestamp  
- int pages (1-20) | - int pageRate (ppm)  
- Task task  
- int timeRemaining |
| Task(task)  
// Constructor  
+ int getTimeStamp()  
// Accessor  
+ int getPages()  
// Accessor  
+ int waitTime(task)  
// Compute wait time  
+ String toString()  
// Print method | Printer(ppm)  
// Constructor  
+ void tick()  
// Do 1 sec of printing  
+ boolean busy()  
// Is printer busy?  
+ void startNextTask(newTask)  
// Schedule newTask on printer |
Task Implementation

public class Task {
    private int timestamp; // When a print request arrives (in ticks since 0)
    private int pages;     // # pages of the print job [1..20]

    public Task(int t) { // Constructor. T is time stamp (seconds)
        timestamp = t;
        pages = (int) (1 + 20 * Math.random());
    } // Task()

    public int getTimeStamp() {
        return timestamp;
    } // getTimeStamp()

    public int getPages() {
        return pages;
    } // getPages()

    public int waitTime(int currentTime) {
        return currentTime - timestamp;
    } // waitTime()

    public String toString() {
        return "Task" + "@" + timestamp + ", " + pages + " pages."
    } // toString()
} // class Task

Printer Implementation

public class Printer {
    private int pageRate; // pages per minute (ppm)
    private Task currentTask;  // current task on printer
    private int timeRemaining; // time remaining on current task

    public Printer(int ppm) {
        pageRate = ppm;
        currentTask = null;
        timeRemaining = 0;
    } // Printer()

    public void tick() {
        if (currentTask != null) {
            timeRemaining--;
            if (timeRemaining == 0)
                currentTask = null;
        }
    } // tick()

    public boolean busy() {
        return currentTask != null;
    } // busy()

    public void startNext(Task newTask) {
        currentTask = newTask;
        timeRemaining = newTask.getPages() * 60 / pageRate;
    } // startNext()
} // class Printer
Simulation: Main Loop

for each tick (second)
1. Is there a new task? 1/180 chance!
   if so, create a new task and add it to the printer queue

2. Do we need to schedule a new task on printer?
   if printer is idle and there is a task on printer queue
      a. remove next task from printer queue and assign it to printer
      b. record wait time for this task [current time – time stamp]
      c. how long will task take?

3. Printer prints for 1 second
   a. take off 1 sec from task time
   b. if task completed, set printer to idle

Finally, compute average wait time & print results

Skeletal Main Program

```java
public class PQSim {
    // Simulate for time (Seconds) for a printer speed (ppm)
    public static void simulate(int seconds, int ppm) {
        Printer = labPrinter = new Printer(ppm); // Create printer with ppm speed
        Queue<Task> printerQueue = new Queue<Task>(); // The printer queue
        int totalWaitTime = 0;
        int nTasks = 0;
        for (int tick = 0; tick < seconds; tick++) {
            // your code here
        }
    }

    private static boolean newPrintTask() {
        return (180 == ((int) (1 + 180 * Math.random())));
    }
}
```

```java
public class PQSim {
    public static void main(String[] args) {
        simulate(3600, 5);
    }
}
```