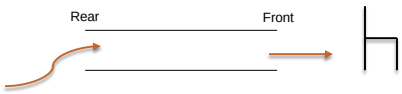
# Queue Applications

Discrete Event Simulation

## Discrete Event Simulation

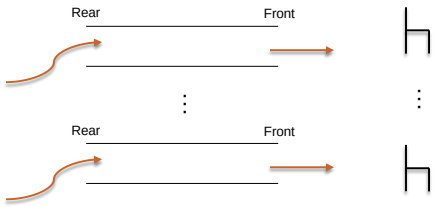


- Single Queue, single server




- Single Queue, multiple servers

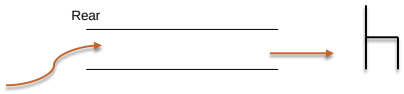

- Multiple Queue, multiple servers



# Example: Single Queue, Single Server

3

- 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

4

| Customer | Inter-arrival Time | Service Time (min) |
|---|---|---|
| C1 | 1.9 | 1.7 |
| C2 | 1.3 | 1.8 |
| C3 | 1.1 | 1.5 |
| C4 | 1.0 | 0.9 |

**Queue Simulation**

| T | Arrival | Queue | Server | Depart |
|---|---|---|---|---|
| 0 | | [] | Idle | |
| 1.9 | C1 | [] | C1 | |
| 3.2 | C2 | [C2] | C1 | |
| 3.6 | | [] | C2 | C1 |
| 4.3 | C3 | [C3] | C2 | |
| 5.3 | C4 | [C4, C3] | C2 | |
| 5.4 | | [C4] | C3 | C2 |
| 6.9 | | [] | C4 | C2 |
| 7.8 | | [] | | C4 |

# Application: Lab Printer Simulation

5

- 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 upto 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

6

- 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 upto 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?

$$\frac{20}{1\ hour} * \frac{1\ hour}{60\ min} * \frac{1\ min}{60\ sec} = \frac{20}{3600} = \frac{1\ task}{180\ sec}$$

# Application: Lab Printer Simulation

7

- 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 upto twice in an hour
- Print jobs are 1-20 pages long
- ∴ 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

8

- 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

9

**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

10

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

# Task Implementation

11

```
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
```

**Task**
- int timestamp
- int pages (1-20)

+ Task(currentTime)
  // Constructor

+ int getTimeStamp()
  // Accessor

+ int getPages()
  // Accessor

+ int waitTime(currentTime)
  // Compute wait time

+ String toString()
  // Print method

# Printer Implementation

12

```
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
```

**Printer**

- int pageRate (ppm)
- Task task
- int timeRemaining

+ Printer(ppm)
  // Constructor

+ void tick()
  // Do 1 sec of printing

+ boolean busy()
  // Is printer busy?

+ void startNextTask(newTask)
  // Schedule newTask on printer

# Simulation: Main Loop

13

**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

14

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

   // 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
      }
   } // simulate()

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