

Bakery Algorithm

Lamport's Bakery Algorithm

Bakery Algorithm

Analogy

- Bakery With A Numbering Machine
- Each Customer Receives Unique Number
 - Numbers Increase By One As Customers Enter
- Global Counter Displays Number Of Customer Being Served Currently
 - All Others Wait In Queue
- After Baker Is Done Serving Customer The Next Number Is Displayed
- Served Customer Leaves

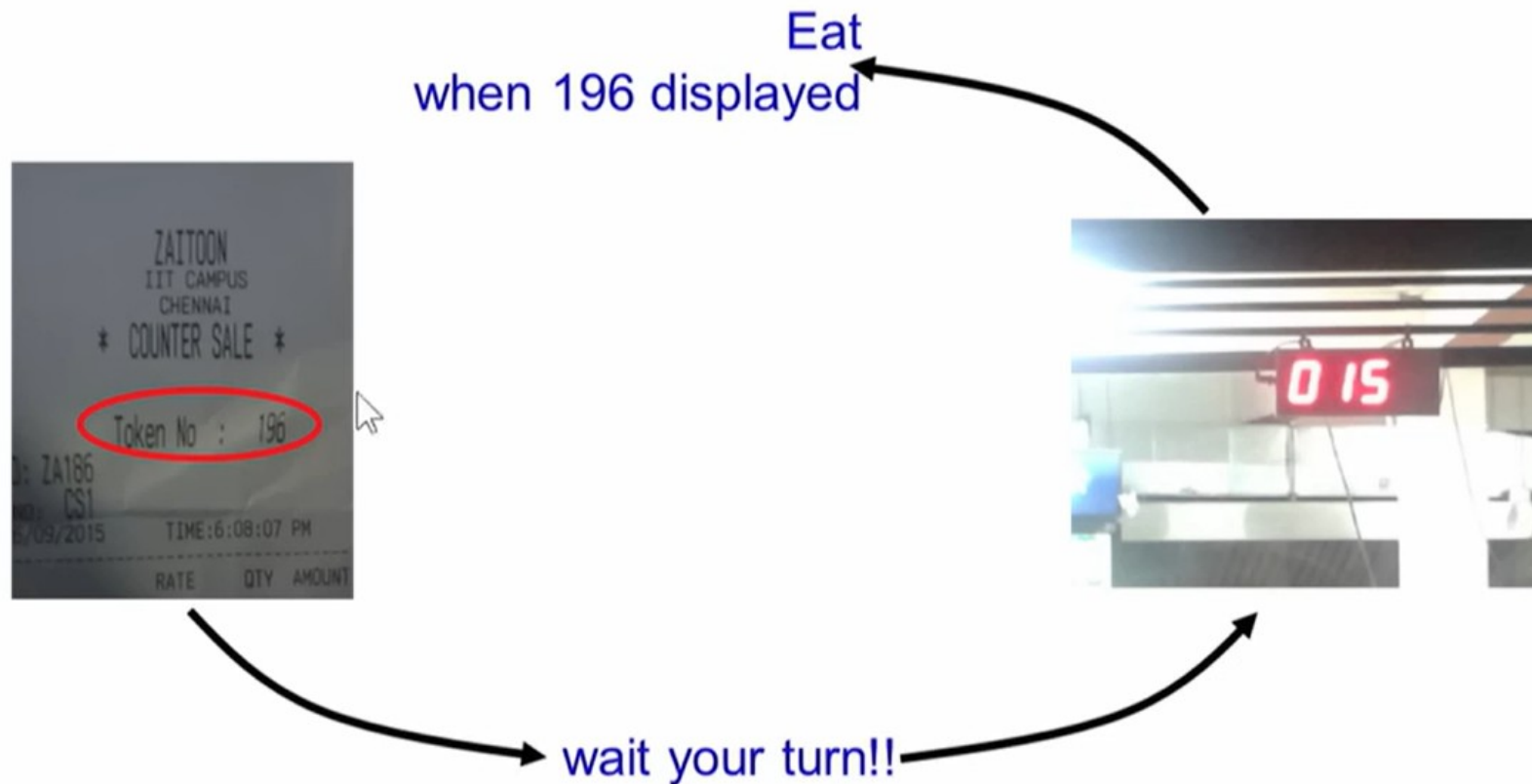
Bakery Algorithm

threads and bakery analogy

- when thread wants to enter critical section it has to make sure it has the smallest number.
 - however, with threads it may not be true that only one thread gets the same number
 - e.g., if number operation is non-atomic
 - if more than one thread has the smallest number then the thread with lowest id can enter
 - use pair (number, ID)
 - In this context $(a,b) < (c,d)$ is equivalent to
 - $(a < c)$ or $((a == c) \text{ and } (b < d))$

Bakery Algorithm

- Synchronization between $N > 2$ processes
- By Leslie Lamport



Simplified Bakery Algorithm

- Processes numbered 0 to N-1
- num is an array N integers (initially 0).
 - Each entry corresponds to a process

```
lock(i){  
    num[i] = MAX(num[0], num[1], ..., num[N-1]) + 1  
    for(p = 0; p < N; ++p){  
        while (num[p] != 0 and num[p] < num[i]);  
    }  
}
```

critical section

```
unlock(i){  
    num[i] = 0;  
}
```

Original Bakery Algorithm

- Without atomic operation assumptions
- Introduce an array of N Booleans: *choosing*, initially all values False.

```
lock(i){  
    choosing[i] = True  
    num[i] = MAX(num[0], num[1], ..., num[N-1]) + 1  
    choosing[i] = False  
    for(p = 0; p < N; ++p){  
        while (choosing[p]);  
        while (num[p] != 0 and (num[p],p)<(num[i],i));  
    }  
}
```

doorway

critical section

```
unlock(i){  
    num[i] = 0;  
}
```

Choosing ensures that a process
is not at the doorway
i.e., the process is not 'choosing'
a value for num

(a, b) < (c, d) which is equivalent to: (a < c) or ((a == c) and (b < d))

Refernces

- <https://lamport.azurewebsites.net/pubs/bakery.pdf>
- <https://www.cs.colostate.edu/~cs551/CourseNotes/BakeryAlg.html>
- <https://nptel.ac.in/courses/106106144>
- https://en.wikipedia.org/wiki/Lamport%27s_bakery_algorithm