CSCI 344/715
Lecturer: Dr. Simina Fluture

Lecture #9

Topics:

Lamport's Bakery Algorithm
Two assertions necessary for the Mutual Exclusion proof

Readings: **Lecture on the web** Bibliography: [SH] 3.5.2.
Class notes [GA] 3.3.3., 3.6.1

Software Implementation – Lamport's Bakery Algorithm

CPUi

Shared variables: choosing, number (integer arrays[1..N]);
Initialized to false and 0.

```
While(true) {

L1:     choosing[i] = 1;
        number[i] = 1 + max(number[1], …, number[N]);
        choosing[i] = 0;
        for (int j = 1; j <= N; j++) {
L2:         while (choosing[j] = 1) {}
L3:         while(number[j] <>0 and (number[j],j) < (number[i], i)) {}
        }

        Critical Section

        number[i] = 0;
}
```

Note: **(a, b) > (c, d) iff (a > c) OR [(a == c) AND (b > d)]**

Assertions that are used in proving **Mutual Exclusion** for the Bakery Algorithm

Assertion 1: if processes *i* and *k* are in the bakery, and *i* entered the bakery before *k* entered the doorway,
then **number[i] < number[k]** (1)

Assertion 2: if process *i* executes its Critical Section, and process *k* is in the bakery (k !=i)
then **(number[i], i) < (number[k], k)** (2)