

# Bakery Problem

## Problem Overview

In this exercise, you must implement the following “simulation”. In a bakery that exclusively bakes chocolate chip cookies we have several bakers (i.e., threads). Each baker has its own oven that is always hot, its own set of supplies and work area. Each baker follows the following pseudo-code:

```
// Baker thread
for (int i=0; i < 10; i++) {
    work (i.e., prepare dough, cut cookies, put them on a baking sheet)
    get oven mitts from the oven mitt rack
    put cookies in the oven
    wait for cookies to be baked
    remove cookies from the oven
    put oven mitts back onto the oven mitt rack
}
```

In the pseudo-code above the “work” and “wait for cookies to be baked” operations simply print a message to standard out and then sleep a random number of microseconds (using `usleep`) between 2 and 5 seconds. The “put cookies in the oven” and “remove cookies from the oven” operations simply print a message to the console. The point of the exercise is to implement the “get oven mitts” and “put back oven mitts” operations, which should also print messages, as seen in the sample output below.

## Problem Details

The bakery is dysfunctional in that the owner does not want to buy oven mitts for each baker. Instead, there are:

- 3 left-handed oven mitts
- 3 right-handed oven mitts

We have three kinds of bakers:

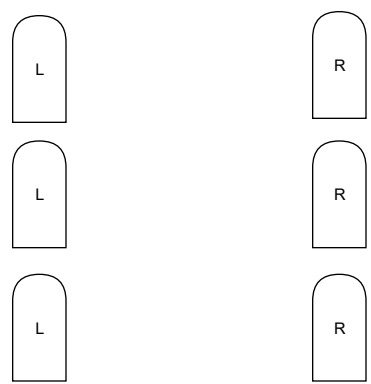
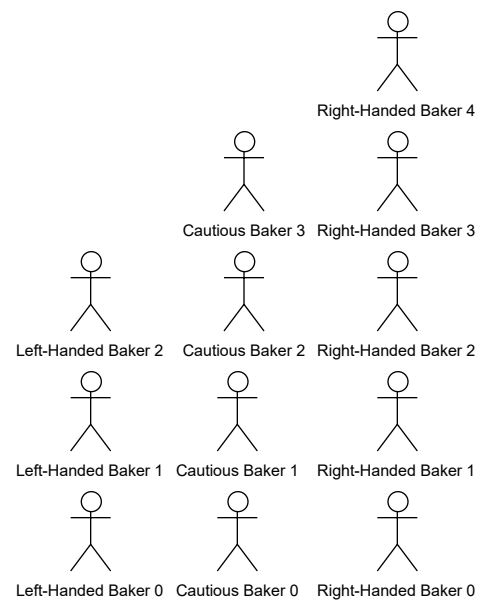
- Left-handed bakers who require only one left-handed oven mitt
  - Right-handed bakers who require only one right-handed oven mitt
  - Cautious bakers who require both a left-handed and a right-handed oven mitt.
- (A cautious baker first takes a left-handed mitt, and then takes a right-handed mitt. )

The bakers are named/numbered as follows:

- Left-handed baker 0 Left-handed baker 1, Left-handed baker 2, ...
- Right-handed baker 0, Right-handed baker 1, Right-handed baker 2, ...
- Cautious baker 0, Cautious baker 1, Cautious baker 2, ...

All bakers must be able to bake, and they cannot use the same oven mitts at the same time. In other words, the shared resources are the oven mitts and the bakers contend for them to do their job.

1. Initialized Bakery with 3 left-handed bakers, 4 cautious bakers, and 5 right-handed bakers. These bakers represent one thread doing some kind of computations in parallel.

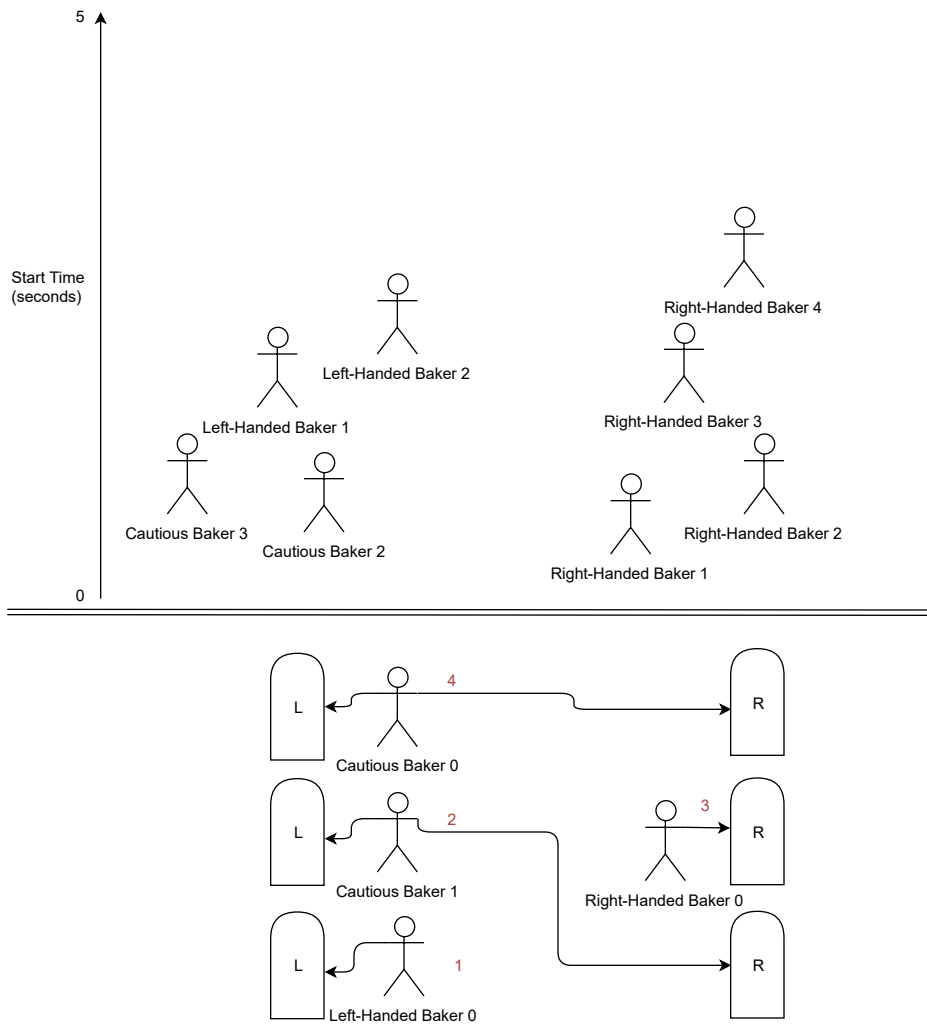


2. We have the bakers "race" to the critical section, where each thread is entering non-deterministically. Before we race to the critical section, we must check the conditional variable to see if there is a non-zero amount of oven mitts left. If there are no oven mitts left then the baker thread must wait for a mitt to become available.



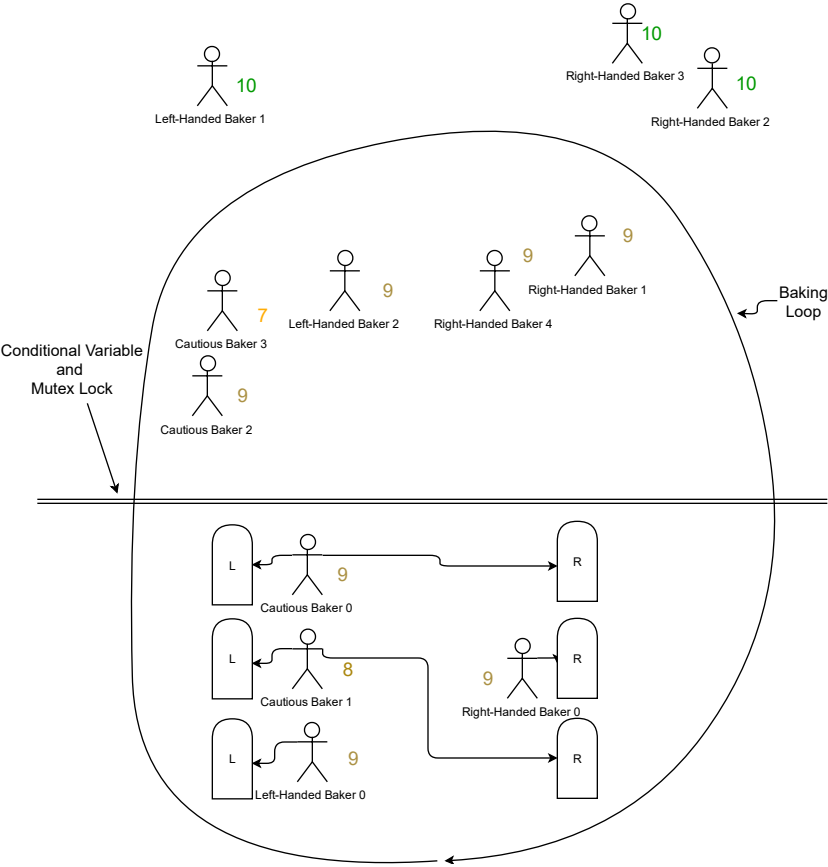
3. Here we see that all of the mitts have been occupied. The bakers with the mitts required are doing some computational work\* while the other bakers are waiting for a mitt to become available.

\*In this project instead of doing computational work, we just want the bakers to sleep for 2 to 5 seconds to *simulate* them doing computational work.



4. Once the bakers get some time with their required oven mitts, they signal that there is an available oven mitt and then they get back in line to make a total of 10 baked items.

Here, we see that the bakers with 10 items are outside of the loop, 3 of the 4 bakers currently working are about to finish.



5. Finally, each baker will have 10 baked items and there will be the same amount of oven mitts as when we started.

