

CarParkControl - condition synchronization

```
class CarParkControl {
    protected int spaces;
    protected int capacity;

    CarParkControl(int n)
    {capacity = spaces = n;}

    synchronized void arrive() throws InterruptedException {
        while (spaces==0) wait();
        --spaces;
        notifyAll();
    }

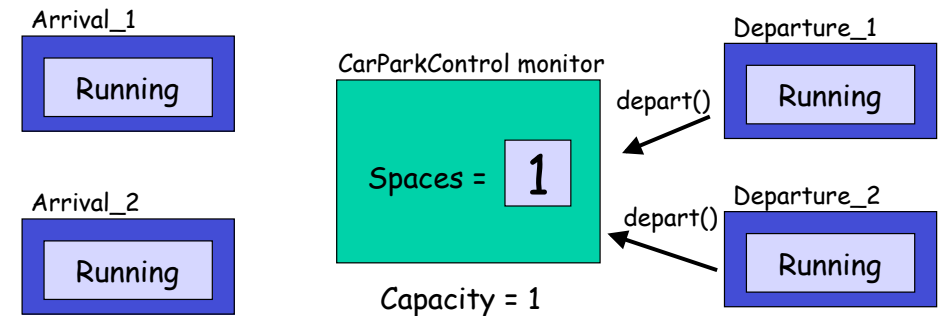
    synchronized void depart() throws InterruptedException {
        while (spaces==capacity) wait();
        ++spaces;
        notifyAll();
    }
}
```

Is it safe to use `notify()` here rather than `notifyAll()`?

2013/2014

©Magee/Kramer 2nd Edition

Notify vs NotifyAll

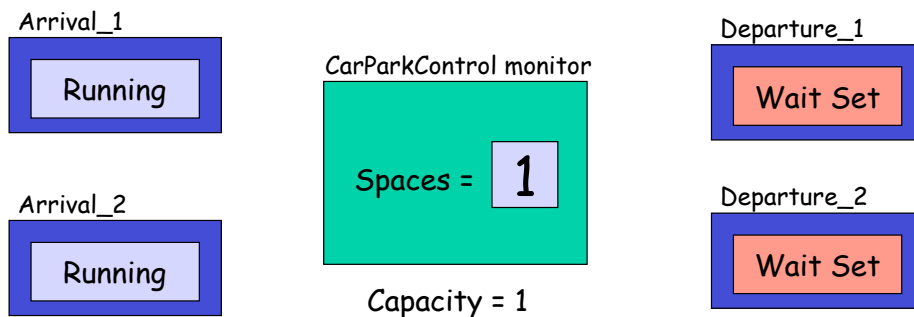


The departure threads call `depart()`...

Concurrency: monitors & condition synchronization
2013/2014

2
©Magee/Kramer 2nd Edition

Notify vs NotifyAll

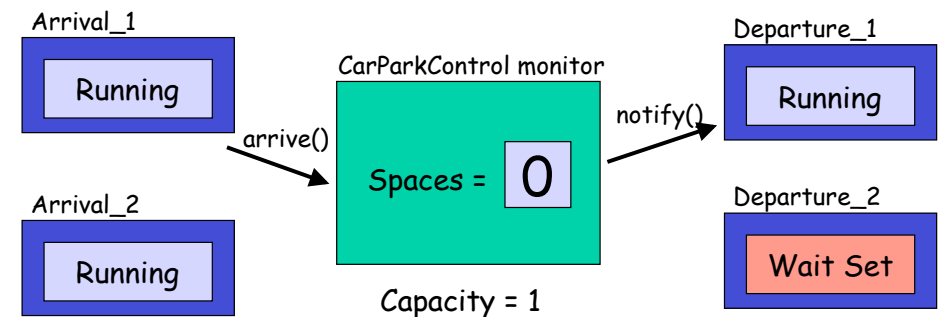


..and because `Spaces == Capacity`, they block.

Concurrency: monitors & condition synchronization
2013/2014

3
©Magee/Kramer 2nd Edition

Notify vs NotifyAll



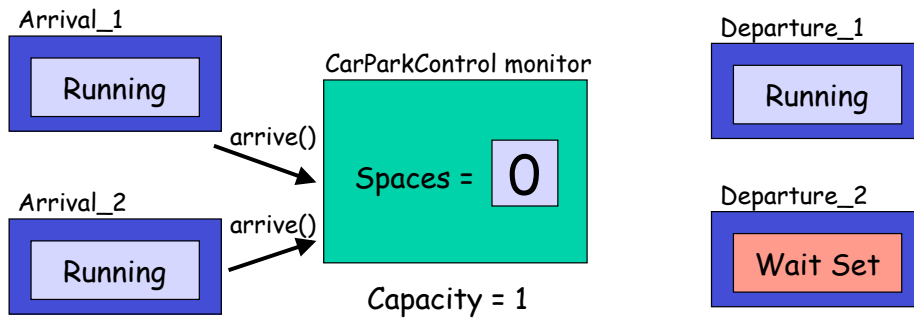
Arrival_1 calls `arrive()` which sets `spaces` to 0 and Departure_1 gets notified.

Before Departure_1 gets a chance to acquire the lock to increment `spaces`...

Concurrency: monitors & condition synchronization
2013/2014

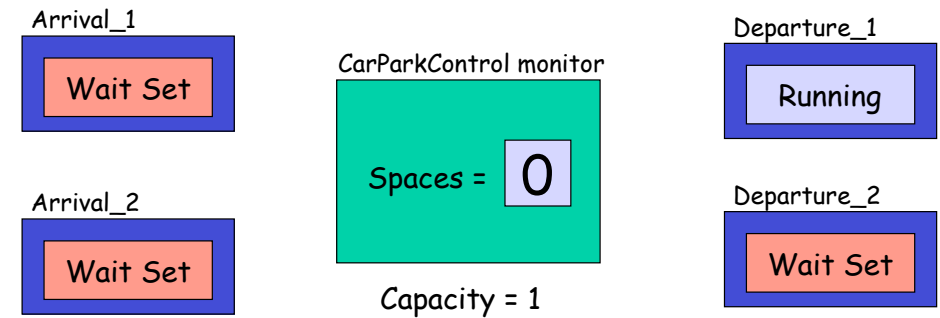
4
©Magee/Kramer 2nd Edition

Notify vs NotifyAll



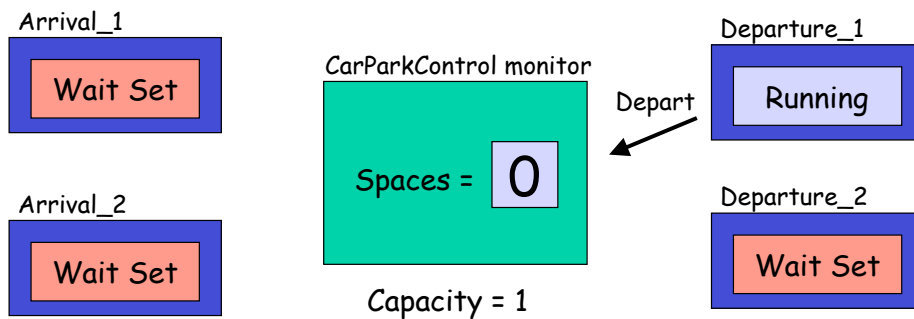
...both arrivals call `arrive()`...

Notify vs NotifyAll



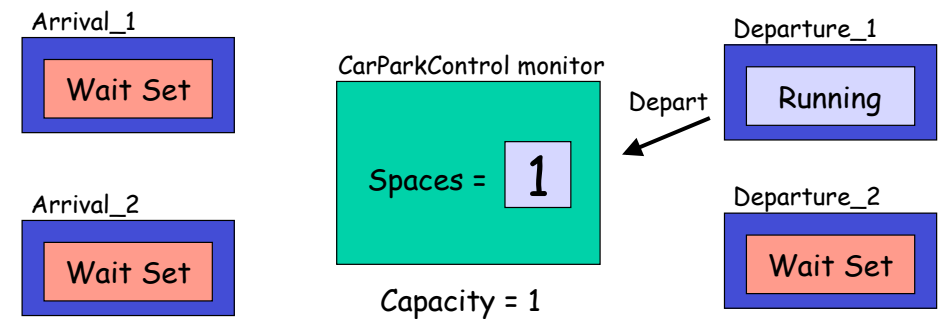
... and block because `Spaces==0`.

Notify vs NotifyAll



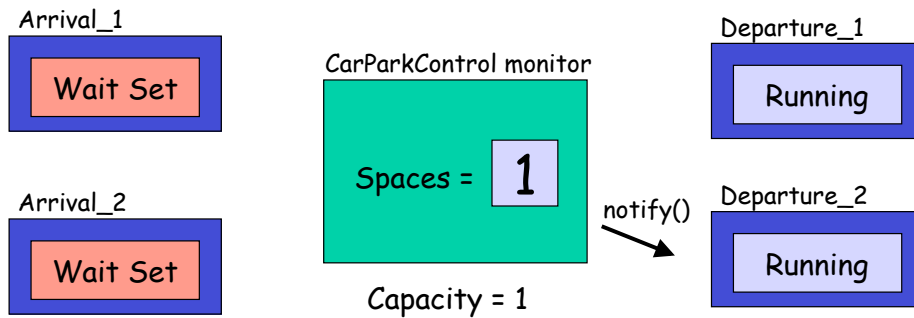
Now `Departure_1` gets the lock on the monitor...

Notify vs NotifyAll



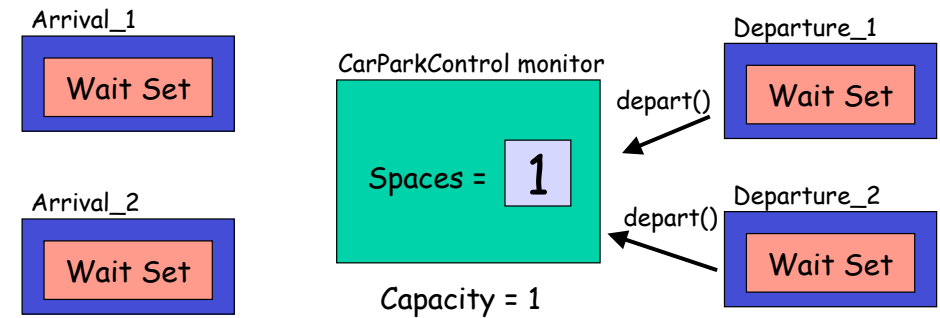
... increments spaces...

Notify vs NotifyAll



... and notifies **Departure_2**.

Notify vs NotifyAll



Now, both departures call **depart()** and block because **Spaces == Capacity**

DEADLOCK!