Title

Imagine you are in the kitchen, and you are about to take an apple from a fruit bowl. You start reaching out and when you are about to grab your apple someone switches the bowl for oranges. This in a nutshell is the critical section problem in computer science. In this example there are two people both have a job; you need to take fruit from the basket and the other person has the job of switching fruit in the basket. This works perfectly fine normally but when you are both doing it at the same time you could look for a fruit expect to find a fruit then all the fruit change. The fruit bowl in the analogy is the critical section.

A brief history of the critical section problem. When computers were young and dumb this problem did exist. A computer only did one thing at a time so there could not be two programs, people, reaching for the same data, fruit bowl. And so since this wasn’t a problem programmer lived in a happy utopia. Until one day someone much smatter than me said what if my computer could do more than one thing at a time. This was one of the first steps that made computers go from really good calculators to something that was in everyone ones homes. Being able to run multiple programs at a time on a computer open endless opportunities without it your computer can’t have a keyboard or mouse. Of course, with this new found power of flexibility came lots of problems one of which is the critical section problem.

**More explanation of cpu switching and when this switching occurs, my the analogy because a puppet and puppet master.**

Let’s get a definition, the critical section problem arises when shared resources are accessed by concurrent processes. Whoa, what are all those words? What do you mean by share resources? What are concurrent processes? Lets tackle those questions one at a time. Shared resources are things that multiple programs are going to use. In our analogy it’s the fruit bowl. Both you and the person replacing the fruit are going to use the fruit bowl. Ok so that makes a little sense, but I still don’t understand either word in concurrent processes? Concurrent means at the same time and a process is another word for a program. Back to our analogy the two people are doing their task concurrently. You process is taking a fruit the other person’s process is switching the fruit.

So how do we fix this? The first answer is take your fruit really fast. Computers can only run one basic line at a time, so if your action take fruit can be written as one basic line you are fine. Unfortunately taking a fruit sounds complicated, you have to move your arm, grab the fruit with your hand and then pick it up. That is a lot of steps. So unfortunately most actions can’t be done in a single basic action on a computer.

The second answer, that is more useful, is get a fruit pass, this pass lets you interact with the fruit. Importantly, taking the fruit pass is really fast, uses the first method. Since computer’s were having problems with critical section computer designers made a special basic action that lets you check if a pass is available. So how does the fruit pass work. The first thing you do when you want to interact with the fruit is grab the fruit pass, because you shouldn’t grab fruit without it, or the problem can occur. Then once you have the fruit pass you can interact with the fruit, grab a fruit or change the fruit. Once you are done interacting with the fruit put the fruit pass back or you are hogging the fruit pass and no one else can interact with the fruit. If the fruit pass wasn’t there in the first place you have to wait until it shows up to take it.

The last solution fixes the problem of having someone greedily keeping the fruit pass. The solution is getting a butler. The butler is the only person allowed to interact with the fruit. Instead of interacting with the fruit yourself you ask the butler to get your fruit. The other person also asks the butler to swap the fruit. Since there is only one butler he can’t do both at once. If you show up and the butler is already doing something you’ll have to wait until they are not. Also, while the butler is interacting with the fruit for you you’ll have to wait until they says the task is done. In this analogy the butler is a server or monitor and the people are programs.

The critical section problem happens when two programs want to access the same data at the same time. There are lots of ways to solve things and it is a really big thing programers now days have to think about because lots of language let programers do multiple things at the same time.

References yay maybe no References