Sleeping Barber

1. There is a barber’s shop where there is only one barber.
2. One barber chair and a number of waiting chairs for the customers.
3. When there are no customers the barber sits on the barber chair and sleeps.
4. When a customer arrives he awakes the barber or waits in one of the vacant chairs if the barber is cutting someone else’s hair.
5. When all the chairs are full, the newly arrived customer simply leaves.

Problems

Deadlock:

There might be a scenario in which the customer ends up waiting on a barber and a barber waiting on the customer, which would result to a deadlock.

Starvation:

Then there might also be a case of starvation when the customers don’t follow an order to cut hair, as some people won’t get a haircut even though they had been waiting long.

Busy Waiting:

Busy waiting or spinning is a technique in which a process repeatedly checks to see if a condition is true, such as whether keyboard input or a lock is available. Spinning can also be used to generate an arbitrary time delay, a technique that was necessary on systems that lacked a method of waiting a specific length of time. Processor speeds vary greatly from computer to computer, especially as some processors are designed to dynamically adjust speed based on external factors, such as the load on the operating system.

Use of Semaphores

Mutex:

The solution to these problems involves the use of three semaphores out of which one is a mutex (binary semaphore). They are:

Customers: Helps count the waiting customers.

Barber: To check the status of the barber, if he is idle or not.

accessSeats: A mutex which allows the customers to get exclusive access to the number of free seats and allows them to increase or decrease the number.

NumberOfFreeSeats: To keep the count of the available seats, so that the customer can either decide to wait if there is a seat free or leave if there are none.

Monitors:

A high level abstraction that provides a convenient and effective mechanism for process synchronization.

The Procedure:

When the barber first comes to the shop, he looks out for any customers i.e. calls P(Customers), if there are none he goes to sleep.

Now when a customer arrives, the customer tries to get access to the accessSeats mutex i.e. he calls P(accessSeats), thereby setting a lock.

If no free seat (barber chair and waiting chairs) is available he releases the lock i.e. does a V(accessSeats) and leaves the shop.

If there is a free seat he first decreases the number of free seats by one and he calls V(Customers) to notify the barber that he wants to cut.

Then the customer releases the lock on the accessSeats mutex by calling V(accessSeats).

Meanwhile when V(Customers) was called the barber awakes.

The barber locks the accessSeats mutex as he wants to increase the number of free seats available, as the just arrived customer may sit on the barber’s chair if that is free.

Now the barber releases the lock on the accessSeats mutex so that other customers can access it to the see the status of the free seats.

The barber now calls a V(Barber), i.e. he tells the customer that he is available to cut.

The customer now calls a P(Barber), i.e. he tries to get exclusive access of the barber to cut his hair.

The customer gets a haircut from the barber and as soon as he is done, the barber goes back to sleep if there are no customers or waits for the next customer to alert him.

When another customer arrives, he repeats the above procedure again.

If the barber is busy then the customer decides to wait on one of the free waiting seats.

If there are no customers, then the barber goes back to sleep.