

( picture is from <https://www.geeksforgeeks.org> )

In the “Sleeping Barber” problem, we need to consider several issues as listed below:

* The number of customers cannot be bigger than 27 and should be odd number(2k+1).
* The number of customers represents the number of threads in the program.
* Max-arrival-time is the maximum amount of time is the time passes between each arrival of customers.
* Max-haircut-duration is the maximum time for a haircut.
* Max-haircut-repetition is repetition of cuts for a customer.
* We need to stop the program when all the customers have their haircuts.
* Additionally, I kept a fixed value for number of chairs since it was not included in the sample input from the report.

The algorithm I followed:

Here, the barber can sleep after finishing the cut. If the customer founds a sleeping barber, he wakes him up and gets his haircut.



Done for the day

Cut hair.

Sleeping barber

Start

I used several websites during this homework:

1. <https://www.geeksforgeeks.org/sleeping-barber-problem-in-process-synchronization/>
2. <https://www.baeldung.com/cs/sleeping-barber-problem>
3. <https://examradar.com/sleeping-barber-problem/>

In the solution, I used semaphores to synchronize the customers’ waiting and barber’s cutting tasks.

* <https://www.calculator.net/standard-deviation-calculator.html> (I used this website to calculate the waiting duration for each customer.
* I showed a couple of examples of how to find the waiting duration for each customer and also added average values as well as other statics.)

1. **./barber 2 2 2 2**

Customer 0 waits 0 seconds since he will come to the barber and wake him up and then gets his hair cut also, the other customers will enter to the waiting room & Customer 1 waits (total haircut time for 0) – (arrive time for 1) = 2\*2 – 2 = 2 seconds

* + Population => (0,2)
  + Mean => 1
  + Standard deviation => 1

1. ./ **barber 3 2 2 2**

Customer 0 waits 0 seconds& Customer 1 waits 4 -2 = 2 seconds & Customer 2 waits 6 – 2 = 4 seconds.

* + Population => (0,2,4)
  + Mean => 2
  + Std => 1.63

1. ./ **barber 2 2 3 2**

Customer 0 waits 0 seconds & Customer 1 waits 6-2 = 4 seconds.

* + Population => (0,4)
  + Mean => 2
  + Std => 2

1. **./barber 3 3 3 2**

Customer 0 waits 0 seconds & Customer 1 waits 6 – 3 = 3 seconds & Customer 3 waits 6\*2 – 3 = 9 seconds.

* + Population => (0,3,9)
  + Mean => 4
  + Std => 3.74