

Unix Lab 6 - Producer Consumer problem with threads

You are asked to solve the following producer consumer problem:

A gas station has a large underground tank of gas which is currently empty.

A tanker truck is filling the underground tank at a rate of 15 liters a second.

At the same time 2 cars are trying to fill up their tanks with gas. They cannot remove fuel from the filling station unless the station has enough gas to satisfy the car's entire request. For example If the station has only 30 liters of gas and a car requests 45 liters, then the car must wait.

There are two cars waiting for fuel at the same time. You must prevent both cars from attempting to fill themselves up when there is only enough fuel for one.

You need to use `pthread_cond_wait` to make sure the thread representing the car does not take gas when there is not enough. Use `pthread_mutex_unlock` to make sure the car and the tanker truck are not accessing simultaneously the variable that represents the current fuel level in the station.

You must not have any "busy wait" in your solution.

Here is a sample run when there is only one car.

No fuel. Waiting...

Filled fuel... 15

No fuel. Waiting...

Filled fuel... 30

No fuel. Waiting...

Filled fuel... 45

Got fuel. Now left: 5

Filled fuel... 20

Filled fuel... 35