Ahmad Adil 12/3/22
Computer Science 332: Operating Systems Laboratory
Professor Debnath

Semaphores:

Output for semaphores.c:

```
adil@adil-VirtualBox:~/Documents/CSC332-hw/lab7$ ./semaphores
Put match and tobacco on the table, Smoker # 2
Smoker #2 with paper has picked up match and tobacco!
Put paper and match on the table, Smoker # 1
Smoker #1 with tobacco has picked up paper and match
Put tobacco and paper on the table, Smoker # 3
Smoker #3 with match has picked up tobacco and paper!
Put match and tobacco on the table, Smoker # 2
Smoker #2 with paper has picked up match and tobacco!
Put paper and match on the table, Smoker # 1
Smoker #1 with tobacco has picked up paper and match
Put paper and match on the table, Smoker # 1
Smoker #1 with tobacco has picked up paper and match
Put tobacco and paper on the table, Smoker # 3
Smoker #3 with match has picked up tobacco and paper!
Put tobacco and paper on the table, Smoker # 3
Smoker #3 with match has picked up tobacco and paper!
Put match and tobacco on the table, Smoker # 2
Smoker #2 with paper has picked up match and tobacco!
Put tobacco and paper on the table, Smoker # 3
Smoker #3 with match has picked up tobacco and paper!
adil@adil-VirtualBox:~/Documents/CSC332-hw/lab7$
```

For the semaphores.c file, I first started out by creating and initializing the semaphores for each of the processes. These processes are Lock, Agent, smoker_Tobacco, smoker_Paper, and smoker_Match. Then in order to ensure the smokers don't run without the agent supplying the ingredients, the semaphores for the smokers were initialized to 0. The agent was also initialized to 0 to make sure that it will sleep after supplying the ingredients.

The lock was initialized to 1, this is to make it possible for the agent to run the first loop and grant permission to the corresponding smokers after putting down the ingredients. The three smokers start off sleeping but are woken up when the agent supplies correct the ingredients. After putting down the ingredients, the agent goes back to sleep and is woken up when the smokers have used the ingredients.

Ahmad Adil 12/3/22
Computer Science 332: Operating Systems Laboratory
Professor Debnath

Output for Pthreads.c:

```
adil@adil-VirtualBox:~/Documents/CSC332-hw/lab7$ gcc pthreads.c -o pthreads
adil@adil-VirtualBox:~/Documents/CSC332-hw/lab7$ ./pthreads
Put tobacco and paper on the table, Smoker # 3
This is Smoker # 3, Smoker with MATCH picked up tobacco and paper.
Put paper and match on the table, Smoker # 1
This is Smoker # 1, Smoker with TOBACCO picked up paper and match.
Put tobacco and paper on the table, Smoker # 3
This is Smoker # 3, Smoker with MATCH picked up tobacco and paper.
Put tobacco and paper on the table, Smoker # 3
This is Smoker # 3, Smoker with MATCH picked up tobacco and paper.
Put paper and match on the table, Smoker # 1
This is Smoker # 1, Smoker with TOBACCO picked up paper and match.
Put paper and match on the table, Smoker # 1
This is Smoker # 1, Smoker with TOBACCO picked up paper and match.
Put match and tobacco on the table, Smoker # 2
This is Smoker # 2, Smoker with PAPER picked up match and tobacco.
Put match and tobacco on the table, Smoker # 2
This is Smoker # 2, Smoker with PAPER picked up match and tobacco.
Put tobacco and paper on the table, Smoker # 3
This is Smoker # 3, Smoker with MATCH picked up tobacco and paper.
Put paper and match on the table, Smoker # 1
This is Smoker # 1, Smoker with TOBACCO picked up paper and match.
adil@adil-VirtualBox:~/Documents/CSC332-hw/lab7$
```

For the pthreads, first the function pointers for the agent, which prints when the agent puts an ingredient down for a corresponding smoker. Then the function pointer for each of the smokers (paper, match, and tobacco) is created.Along with this, threads are initialized (agent, smoker_paper, smoker_match, and smoker_tobacco). Finally a for loop is used to create new threads and waits for the thread to terminate before moving onto the next.