CSE 344
SYSTEM PROGRAMMING
HOMEWORK 3 REPORT

OĞUZHAN SEZGİN
1801042005

First of all, homework does not work as desired. Sometimes deadlocks occur. The synchronization problem has been tried to be solved by using fifos and named semaphore.

First of all, each process reads the fifo name from the file. While creating the Fifo, it checks whether it has been created before. If it is created, it understands that it has been received by another procces and passes to the other fifo name. It continues until find a fifo that has not been created before. Then, the process that creates its fifo saves the pid number and value of the potato in its hand in the shared memory and goes to wait to read the fifo. After the last process reading the file generates its own fifos, it sends the "start" message to all other fifos. So other waiting processes get activated and send their potatoes to a randomly selected fifo. After sending, they start to wait to read the incoming message. After the message arrives, they read the message and decrease its value by one to cool the incoming potato. After reducing, they wait to decrease the semaphore. The process that decreases the semaphore value writes the new value of the potato to the shared memory and increases the semaphore value. Then he checks whether the potato in your hand is cold or not. If it is cold, it waits for other potatoes to come to it without sending the potato, if it is not, it sends it by selecting a random fifo.

This cycle continues until all the pataes are cooled. The potato cooling process in its hand checks whether all the other potatoes are cold. If all of them are cooled, it writes the message "0-0" to all the waiting fifos and terminates itself. The processes that read this reduce the running process value in shared memory and terminate itself. Each process thus checks whether there are any other running processes before it terminates. The last process understands that it is the last running process from this value and unlink the shared memory and semaphore, then terminates itself.

However, in this process, when different processes write messages to the same fifo consecutively, some messages disappear and this causes a deadlock.