CS 307 HOMEWORK #2 - DINING PHILOSOPHERS PROBLEM (using Semaphores)

Berna Yıldıran 26431

My program consists of 3 function and a main part.

1 - Barrier_for_Philosophers()

Firstly, I wrote the barrier function for Philosophers which is called “Barrier_for_Philosophers()”. In this function, I up the semaphore n-1 times with a “for loop” and in the second “for loop” down the other semaphores except this semaphore. I used try-catch structure. If an exception is thrown from within the try-block, the catch-block is executed. The catch-block specifies the variable “e” to hold the value of the exception in the scope of the catch-block.

2 - eat()

Secondly, I wrote a function for events happening during the dining. While Philosopher is not eating (are_you_eating != 1) and if both fork is available, then the philosopher starts eating. In order to understand both fork is available, function first checks if left fork is available and make a down on left fork’s semaphore. After philosopher manages to acquire the left fork, function passes to the other if statement which is for right fork. If left fork is available but right fork is not, then philosopher will release the left fork by making a down on its semaphore. If right fork is available, function will make a down on right fork’s semaphore. If philosopher manages to pass these if statements and gets two of the forks, s/he will start eating (philosopher’s state will change to “EATING”). Function will print a statement which indicates that philosopher is eating. Also, following these statements there are statements that indicates the operations of the GUI. When philosopher take the fork it will turn red and when s/he starts eating her/his spaghetti will turn into blue. When s/he puts the fork and stop eating, spaghetti and forks will turn into white. After that, the philosopher will release the left and right fork by making a down on their semaphores.

3 – think()

In this function, philosophers will think for random amount of time in the range of (10 seconds = 10000 milliseconds). State of the philosopher will become “THINKING” and a random number will be generated which determine their sleep (thinking) time. A statement that indicates philosopher is thinking will be printed and function will enter the try-catch block. Inside the try block, philosopher will be sleep for the random amount time that we generated before. After thinking a statement that indicates philosopher is hungry will be printed. Hungry_GUI will change the color of the spaghetti to red and philosophers state will become “HUNGRY”.

4 – run()

A random amount of time for philosophers to arrive to the table will be generated (in the range of 10 seconds). After the arrive time is determined a statement that indicates philosopher is coming to the table in # seconds will be printed. In this arrival time interval, philosopher will be sleeping for rand_arrive_time seconds. When philosopher comes to the table PutPlate_GUI will create a yellow spaghetti. After that the barrier will ensure that philosopher who are currently sitting at the table will wait for other philosophers who are approaching to the table. After all philosophers arrived they all start dining and. Subsequently think() and eat() until program exits.