CS307 HW2 Report

Mehmet Enes Battal – 26354

In this homework a solution for the dining philosophers problem was implemented. First thing was to make philosophers come to the table and put their plate on. A random time was selected so that philosophers would arrive at different times. But in order to start eating every philosopher should have been present at the table so the ones who arrived first needed to wait. Since each philosopher has its own thread, these threads had to wait each other in order to continue execution. This was achieved by implementing a barrier. The barrier algorithm that was used can be found below.

```
for(int i=0; i < N-1; i++)
   barriers[id].release();

for(int i=0; i < N; i++) {
   if(i != id)
      barriers[i].acquire();
}
```

Thread makes an up operation on its barrier semaphore N-1 times where there are N concurrently running threads. In this problem N is 5. Then it makes a down operation to all the other threads' barrier semaphores. This way all the threads wait for each other.

After barrier every philosopher is present at the table so they can start the think-eat cycle. A philosopher thinks for a random amount of time between 0 and 10 seconds. Then it tries to take the forks to eat. If it can it starts eating but if it can't, it waits for the forks to be available in hungry state. When available, it eats and puts the forks down and starts to think again for a random amount of time. This cycle continues.