**PROJECT 2 REPORT**

In this Project I used Linux to compile c code with virtual box. In the code I gave a solution for the dining philosopher problem. Fort his I used semaphores. For example, there are sticks for philosophers to eat their meal but one must use 2 stick. As a solution while a philosopher using sticks the one next to him must wait. After waiting He will get the turn to use sticks and eat. They have also dining time which is used to calculate how much time did it take to finish meal for each philosopher. They have also a thinking time. Those times are chosen randomly in my code. Also there is a distribution time and my code gets it from user. If it is exponential I used this algorithm:

thinkingTime = expoDistribute((double) (a->minThink + a->maxThink) / 2);

diningTime = expoDistribute( (double) ((a->minDine + a->maxDine) / 2));

And if it is uniform I used this algorithm:

thinkingTime = uniDistribute(a->minThink, a->maxThink);

diningTime = uniDistribute(a->minDine, a->maxDine);

My output with 5 philosophers:

Philosopher 1 complete dining and he waited for 20.171686ms

Philosopher 3 complete dining and he waited for 20.223709ms

Philosopher 2 complete dining and he waited for 20.311350ms

Philosopher 4 complete dining and he waited for 84.412236ms

Philosopher 0 complete dining and he waited for 100.848010ms

**...Program finished with exit code 0**

**Press ENTER to exit console.**

Melik BOZKURT