CSE 344
SYSTEM PROGRAMMING
HOMEWORK 4 REPORT

OĞUZHAN SEZGİN
1801042005

Program works as a desired. Firstly main thread read students file and finds the student number. Then it create a names char array and 2d attributes integer array and pipe array for giving homework. This arrays stores the students names and attributes. Main thread also stores this attributes and names in struct array for give students threads as a parameter. After that main thread creates students threads. Then it creates the cheater student thread and wait the full semaphore.

Cheater thread read homework file one by one in a infinite loop, then it puts the homeworks in the homework queue. Before it reads the homework it checks the money. If the money is lower then the minimum cost of students, then it break the loop and termite itself. If the money is grater than the minimum cost(**min_cost**), then it continue to read homework. When it checking money it waits the **resource** semaphore and then checks money. After checking money it waits the **queue_per** semaphore. When it take this semaphore it puts the homework in queue, post the **full** semaphore and again wait resource semaphore for check money. This loop continue until the money is over or homework are over. When the homeworks are over then it assign **is_end** value to 1.

After the create students thread and cheater thread, main thread enter in a infinite loop for take homeworks from queue and give the students. Firstly it wait the resource semaphore and checks the money like cheater. Then it waits the **full** semaphore first. If it can takes the **full** semaphore this means that there is at least one homework in queue. Then it waits **queue_per** semaphore and takes a homework from queue. After taking the homework it waits the **empty** semaphore. If the it can takes the empty semaphore this means that at least one student available for doing homework. After taking empty semaphore it waits the resource semaphore for access the attribute arrays and and busy student(**busy_st**) array. This array is for checking students availability. If the student available then it index value will be 0 if she is not then value will be 1. Main thread checks this arrays and find a available student for give homework. When it finds the student, it decrement the money according to chosen student cost and assign **busy_st** array value 1 and writes the homework attribute to her pipe. Then again it waits the checking money until the money is over or homeworks are over. When dequeue function return **Empty** and **is_end** value is 1 it understand that homeworks are over. If money is over or homeworks are over then it writes **Empty** attribute to the all students pipes.

Student thread waits for read her pipe. When main thread writes the her pipe, she read attribute and solves the problem. Then assigns the her **busy_st** array value 0. After that it post the **empty** semaphore for report availability. Then it waits again the reading to her pipes. If she read Empty attribute from her pipe it breaks the loop and terminate itself.

When main thread breaks its loop it waits until the all other thread terminate. After the all threads terminated then it prints the summary of the program, free all resource and terminate itself.

When program running,if user press the Ctrl+C keys then program goes the handler and calls the **error_exit** function and terminate itself. In **error_exit** function call the pthread_cancel function and waits the other threads terminate after that it free all resource and exit.