

# CSE 4034 Project 2 Report

## Group Members:

Yiğit Göksel 150119053

Berk Yıldız 150117052

```
for(i=0;i<typeA_input;i++){
    status=pthread_create( newthread: &threadA[i], attr: NULL, start_routine: typeA_work, arg: NULL);
    if (status != 0)
    {
        fprintf( stream: stderr, format: "Failed to create thread %d: %s\n", i , strerror( errnum: status));
        return 0;
    }
}
for(i=0;i<typeB_input;i++){
    status=pthread_create( newthread: &threadB[i], attr: NULL, start_routine: typeB_work, arg: NULL);
    if (status != 0)
    {
        fprintf( stream: stderr, format: "Failed to create thread %d: %s\n", i , strerror( errnum: status));
        return 0;
    }
}
for(i=0;i<typeC_input;i++){
    status=pthread_create( newthread: &threadC[i], attr: NULL, start_routine: typeC_work, arg: NULL);
    if (status != 0)
    {
        fprintf( stream: stderr, format: "Failed to create thread %d: %s\n", i , strerror( errnum: status));
        return 0;
    }
}
for(i=0;i<typeD_input;i++){
    status=pthread_create( newthread: &threadD[i], attr: NULL, start_routine: typeD_work, arg: NULL);
    if (status != 0)
    {
        fprintf( stream: stderr, format: "Failed to create thread %d: %s\n", i , strerror( errnum: status));
        return 0;
    }
}
```

After reading the text file, we created threads according to input values. Four different for loops for four different types of technicians. All of them have their own functions to execute.

```

void * typeA_work(){
    sem_wait( sem: &sem_no0f_tires);
    sem_wait( sem: &sem_chassis);
    tires();
    sem_post( sem: &sem_tires);
    sem_post( sem: &sem_chassis);

    sem_wait( sem: &sem_no0f_painting);
    sem_wait( sem: &sem_top);
    painting();
}

void * typeB_work(){
    sem_wait( sem: &sem_no0f_chassis);
    chassis();
    sem_post( sem: &sem_chassis);
}

void * typeC_work(){
    sem_wait( sem: &sem_no0f_seats);
    sem_wait( sem: &sem_chassis);
    seats();
    sem_post( sem: &sem_seats);
    sem_post( sem: &sem_chassis);
}

void * typeD_work(){
    sem_wait( sem: &sem_no0f_engine);
    sem_wait( sem: &sem_chassis);
    engine();
    sem_post( sem: &sem_engine);
    sem_post( sem: &sem_chassis);

    sem_wait( sem: &sem_no0f_top);
    sem_wait( sem: &sem_tires);
    sem_wait( sem: &sem_seats);
    sem_wait( sem: &sem_engine);
    top();
    sem_post( sem: &sem_top);
}

```

Type A technicians firstly have sem\_wait that prevent them from exceeding the total number of tires. Then they wait for the chassis to add tires. That's why we have sem\_wait for that. After adding tires sem\_post for tires and chassis. For painting also there is sem\_wait that prevents them from exceeding the total number of paint operations. Then there is a sem\_wait that waits for the top.

Type B technicians firstly have sem\_wait that prevent them from exceeding the total number of chassis. After putting the chassis it should sem\_post so that other operations that wait for the chassis can continue.

Type C technicians firstly have sem\_wait that prevent them from exceeding the total number of seats. Then they wait for the chassis. After attaching the seats it should sem\_post so that other operations that wait for the seats can continue. Also there is sem\_post for chassis because tires and engine are also waiting for chassis.

Type D technicians firstly have sem\_wait that prevent them from exceeding the total number of engines. Then they wait for the chassis. After adding the engine it should sem\_post so that other operations that wait for the engines can continue. Also there is sem\_post for chassis because tires and seats are also waiting for chassis. Then it has sem\_wait that prevents them from exceeding the total number of car tops. It waits for tires, seats and engines. After adding the top cover it has sem\_post so painting can be implemented.