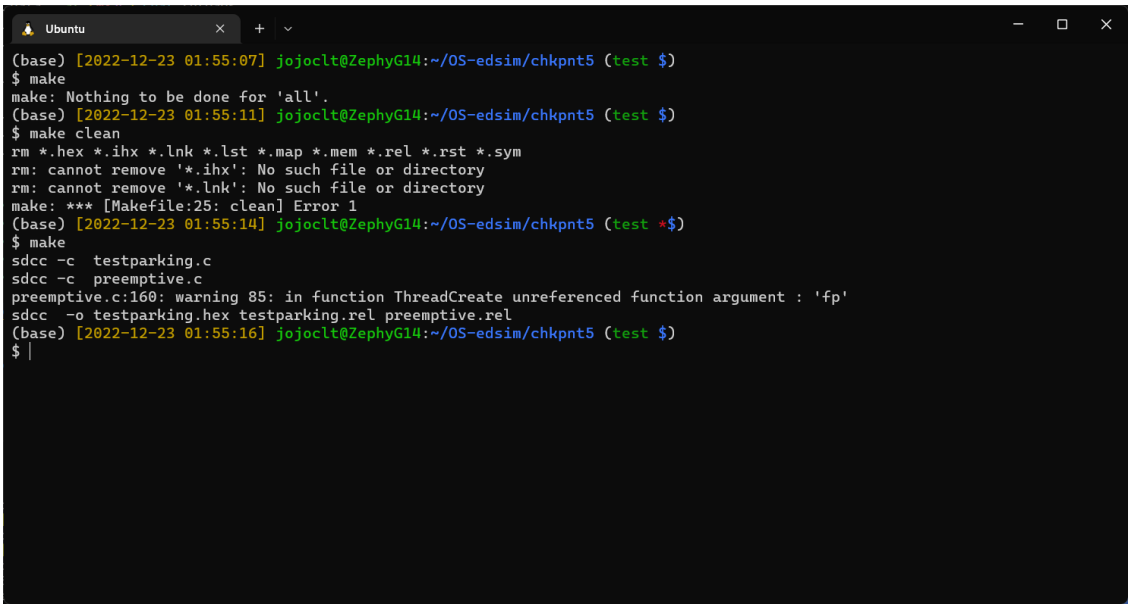
# Programming Project Checkpoint#5

姚林飛 109006243

## [1] Typescript for compilation



**[2] Screenshots and explanation**

1) delay(n), now()

Delay implementation - When it is called, the thread is delayed by n time units (current_time + n), where the time units are based on the timer-0 ISR.

When the timer-0 ISR is called, it will increase the global timer when the thread is at thread#0.

If all the threads call delay() and finish their delays all at the same time, it will use the first found thread that is free.

Note: Everytime Thread is Exit, it will go through all threads (car) again, to check whether there is a car that wants to go in, but not yet knows that the lot is free.

2) Thread Termination and Creation

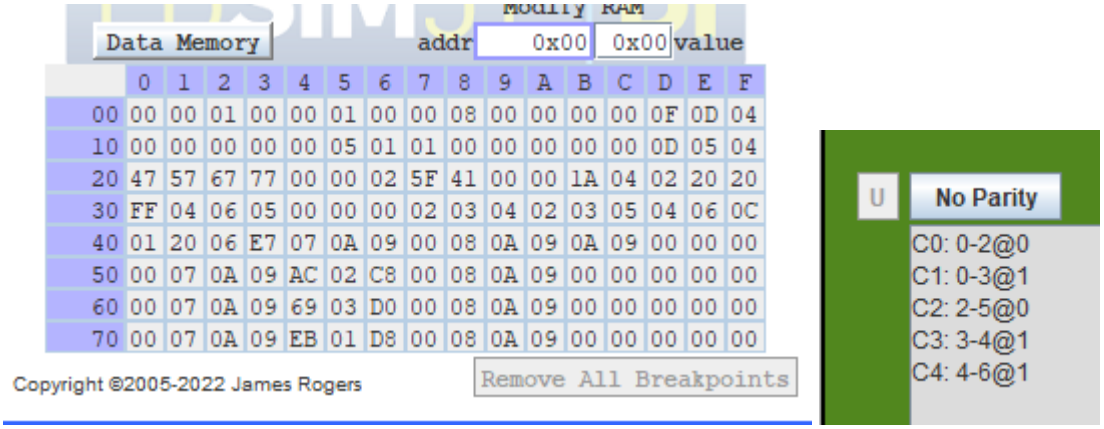
I added the Semaphores for Thread Creation and Termination.

```
ThreadID ThreadCreate(FunctionPtr fp) {
    SemaphoreWait(threads);
    EA = 0;

    for (createdThread = 0; createdThread < MAXTHREADS; createdThread++) {
        tmp = bitmap & (1 << createdThread);
        if (tmp) continue;
        break;
    }
```

```
void ThreadExit(void) {
    __critical {
        bitmap = bitmap ^ (1 << currentThread);
        SemaphoreSignal(threads);
    }
    EA = 0;
    currentThread = 0;
```

3. Parking Lot Example




Where 0x35-0x39 is arrival time and 0x3A to 0x3E is departure time, and 0x2B is the parking_lot status by bit.