Kwangmin Kim

CST-221

06/02/2019

John Zupan

Github: https://github.com/km0312/CST-221

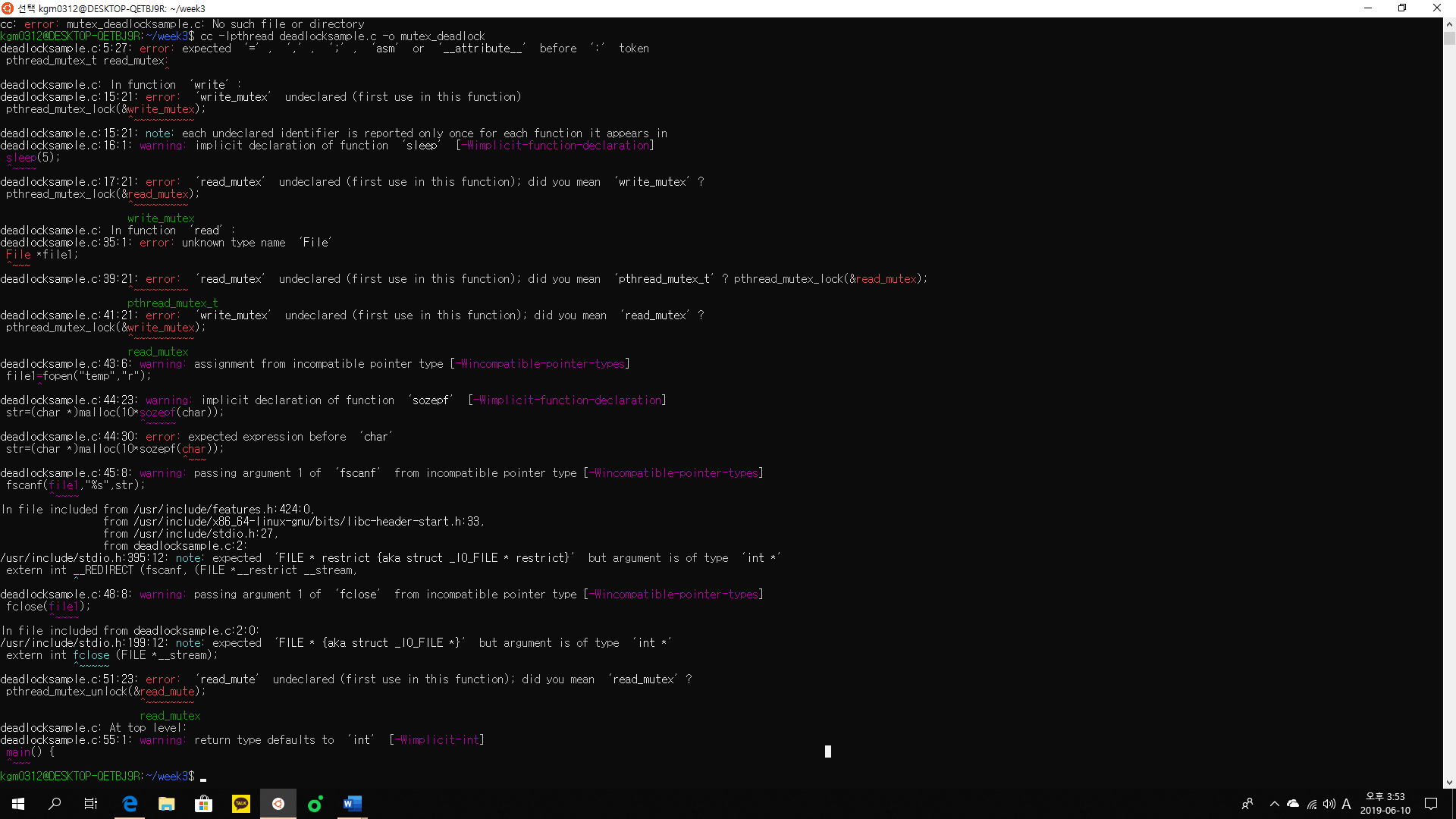
Deadlock avoidance

<scenario>

When several processes compete to access the same resource, it can occur a deadlock. A deadlock is when one and the other thread are waiting for the same resource, so this situation makes both of them to wait for each other forever.

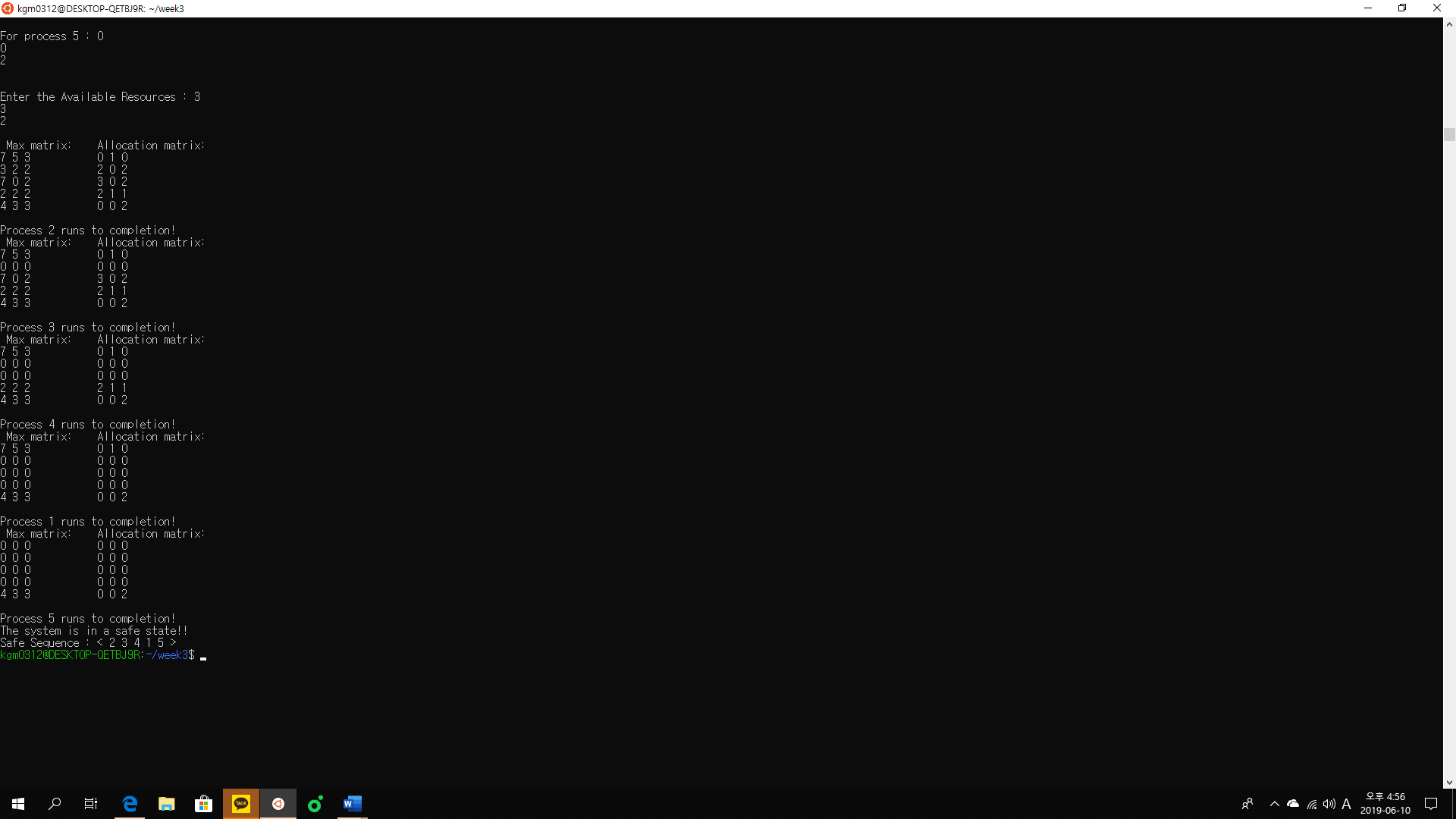
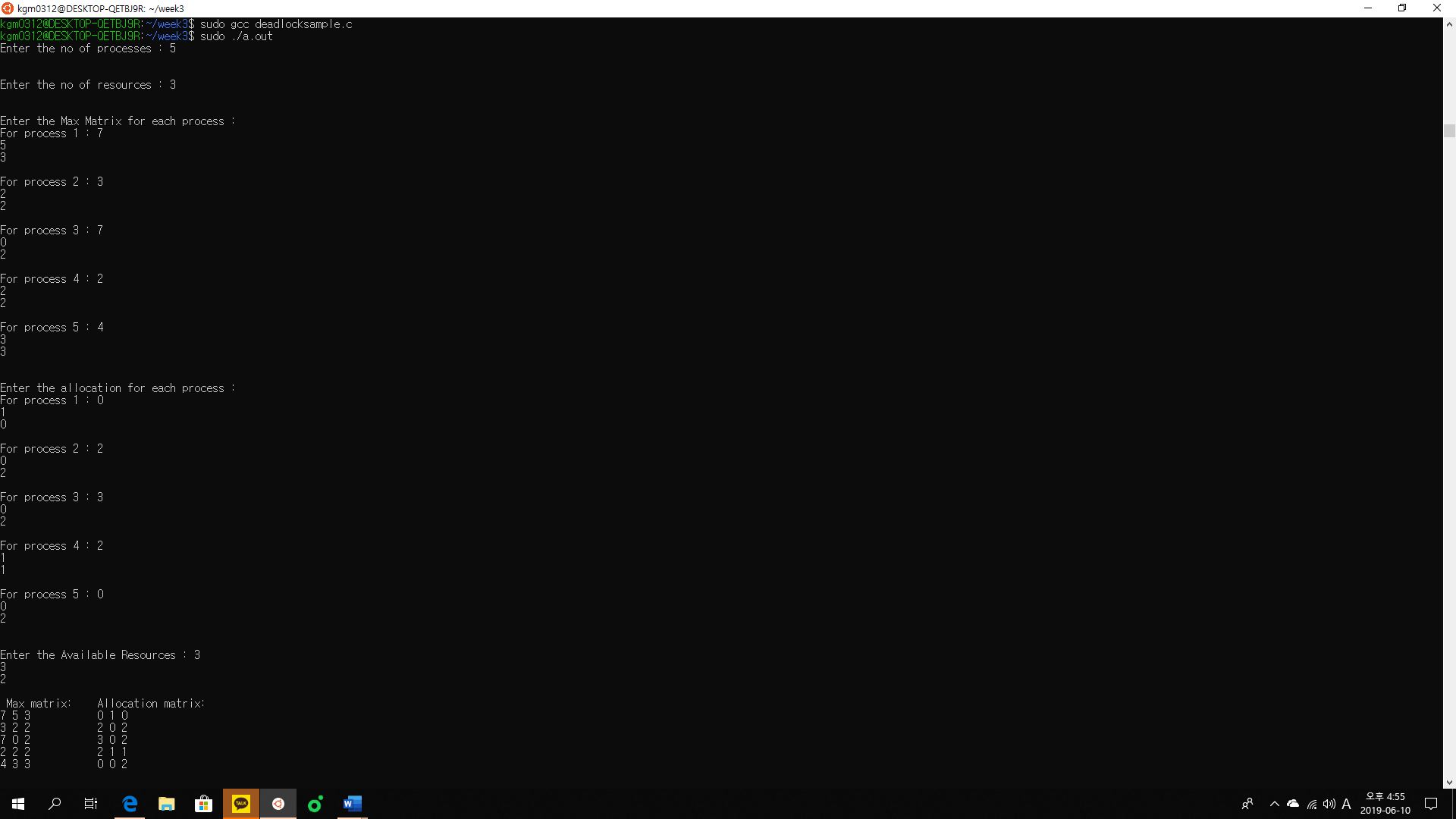
<approach>

To not end up in a deadlock, there would be various way, but I am going to use mutexes so that many threads won’t use the same resource at the same time.



<correction>

Somehow mutex doesn’t let me compile the c program. So, I guess mutex doesn’t work to avoid deadlock state. I tried to find new way to avoid deadlock and I found Banker’s algorithm for deadlock avoidance.



<analysis>

To avoid deadlock state, there would be various way to do like mutex, Banker’s algorithm, and clock timer. But I have chosen mutex and banker’s algorithm but only banker’s algorithm worked properly. Since the scenario was that a deadlock occurred when several processes were accessing the same resource at the same time, I think banker’s algorithm would be better way to use. It keeps safe state and when it is unsafe it creates deadlock and wait until it recovers safe state.

<reference>

Studentcpu. (n.d). BANKER’S ALGORITHM Avoidance of Deadlock Operating Systems lap Linux based. Retrieved from:

https://www.studentcpu.com/2009/10/bankers-algorithm-avoidance-of-deadlock.html.