ECE 353: Systems Software
Lecture 23

Parallelization Example

1.0.1

Jon Eyolfson
March 8, 2023

You Do NOT Want Deadlocks

The more locks you have, the more you have to worry about deadlocks

Conditions for deadlocking:

1. Mutual Exclusion (of course for simple locks)
2. Hold and Wait (you have a lock and try to acquire another)
3. No Preemption (we can’t take simple locks away)
4. Circular Wait (waiting for a lock held by another process)

A Simple Deadlock Example

Consider two processors trying to get two *locks*:

Thread 1

```
Get Lock 1
Get Lock 2
Release Lock 2
Release Lock 1
```

Thread 2

```
Get Lock 2
Get Lock 1
Release Lock 1
Release Lock 2
```

Thread 1 gets Lock 1, then Thread 2 gets Lock 2, now they both wait for each other

Deadlock

You Can Ensure Order to Prevent Deadlocks

```
void f1() {
    locktype_lock(&l1);
    locktype_lock(&l2);
    // protected code
    locktype_unlock(&l2);
    locktype_unlock(&l1);
}
```

This code will not deadlock, you can only get l2 if you have l1

You Could Also Prevent A Deadlock by Using `trylock`

Remember, for pthread there's `trylock` that returns 0 if it gets the lock

```
void f2() {
    locktype_lock(&l1);
    while (locktype_trylock(&l2) != 0) {
        locktype_unlock(&l1);
        // wait
        locktype_lock(&l1);
    }
    // protected code
    locktype_unlock(&l2);
    locktype_unlock(&l1);
}
```

This code will not deadlock, it will give up `l1` if it can't get `l2`

We Explored More Advanced Locking

We have another tool to ensure order

- Condition variables are clearer for complex condition signaling
- Locking granularity matters
- You must prevent deadlocks

We’re Running a Bank Doing 10,000,000 Transfers

We have a simulation that can create varying number of accounts

Each account has a unique ID, and a balance (starting with $1 000)

We generate transfers between random accounts for 10% of their current balance

They must call `securely_connect_to_bank` before starting the transfer

Coding Example

Done live! We'll have data races and deadlocks

Our goal is to be faster than 11 seconds

You can find the template in the `examples` repository

To compile it, run the following commands:

```
cd 23-parallelization-example # if not already there
meson setup build
meson compile -C build
```

Run the program using: `./banksim <num_accounts>`