

# Synchronization

Questions answered in this lecture:

Why is synchronization necessary?

What are race conditions, critical sections, and atomic operations?

How to protect critical sections with atomic loads and stores?

# Cooperation requires Synchronization

**Example:**

Two threads share account balance in memory

Each runs common code, deposit()

```
void deposit (int amount) {  
    balance = balance + amount;  
}
```

**Compile to sequence of assembly instructions**

```
load    R1, balance  
add     R1, amount  
store  R1, balance
```

**Which variables are shared? Which private?**

# Concurrent Execution

What happens if 2 threads deposit concurrently?

Assume any interleaving of instructions is possible

Make no assumptions about scheduler

Initial balance: \$100

Thread 1:deposit(10)

Thread 2:deposit(20)

Load R1, balance

Load R1, balance

Add R1, amount

Add R1, amount

Store R1, balance

Store R1, balance

What is the final balance?