Synchronization

Materials adapted from: The Little Book of Semaphores by Allen B. Downey

What does synchronization mean?

- Generally
  - Make two things happen at the same time. Examples?
- Computing
  - A relationship among events concerning the time of occurence
    - Any number of events

Synchronization Constraints

- Requirements related to the order of events/threads
  - Serialization
    - Event A must happen before Event B
  - Mutual Exclusion
    - Events A and B must not happen at the same time
- In “real” life, we might use a clock to enforce synchronization constraints
- In computing, clocks are not sufficient
- Software techniques to enforce synchronization constraints

Execution Model

- The first step in understanding software synchronization is a model of execution
- In simplest model,
  - computers execute one instruction after another
  - synchronization is trivial
  - If statement A comes before statement B, A will be executed first.
- Complications
  - Parallel computing (multi-processors)
  - Multi-threading (single processor)

More complex execution models

- Parallel computing (multiple processors)
  - Is a statement on one processor executed before a statement on another?
- Multi-threading (single processor)
  - A programmer does not control when each thread runs
    - This is the job of the OS scheduler
  - The programmer cannot tell when statements in different threads execute

- For synchronization purposes
  - We know the order of execution within one processor or thread
  - Between processors (or threads) it is not possible to tell the order

Synchronization example

- You and Bob live in different cities.
- How do you find out who ate lunch first on a given day?

Synchronization with message passing

- Message passing is a solution to many synchronization problems
- How to guarantee that Bob will not each lunch until you do

Thread A (You)

1. Breakfast
2. Work
3. Eat lunch
4. Call Bob

Thread B (Bob)

1. Breakfast
2. Wait for a call
3. Eat lunch

- Within a thread we know the order things happen: a1 < a2 < a3 < a4
- a1 < b1?
- a3 < b3?

Concurrent Events

- We sometimes say concurrent events happen at the same time.
- Two events are concurrent if we cannot tell by looking at the program which will happen first.
- In previous example,
  - You and Bob ate lunch sequentially (we know the order of events)
  - You and Bob ate breakfast concurrently (we cannot tell the order of events)

What is the output?

Thread A

1. Print “yes”

Thread B

1. Print “no”