Network Programming

Race Condition

Lisa Frye, Instructor
frye@Kutztown.edu
Kutztown University

Race Condition

- Output depends on the sequence of events.
- The system’s or software’s behavior is dependent on the sequence or timing of events.
- A situation in concurrent programming where two concurrent threads or processes compete for a resource and the resulting final state depends on who gets the resource first.

Example

- Couple wants to deposit and withdraw money from a shared checking account during lunch.
- Unordered list of events
  - Wife: deposit $500
  - Husband: withdraw $400
  - Original balance: $200

Husband gets to ATM first

- Insufficient funds !!

Wife gets to ATM first

- Balance: $200
- Wife deposits $500
- Balance: $700
- Husband withdraws $400 successfully!

Programming Example

- read the value of x from a file or DB
  - Current value is 10
- x = x + 5
- print x

Processes run Sequentially

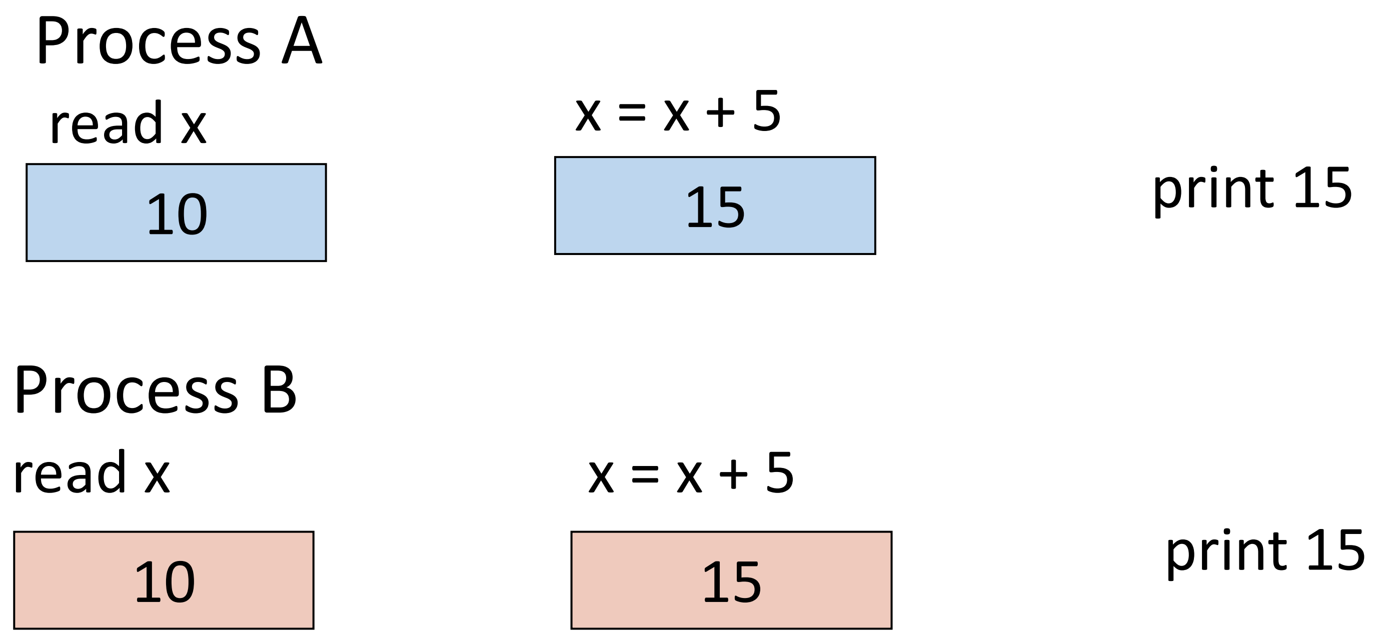
Process A

read x	x = x + 5	
10	15	print 15

Process B

read x	x = x + 5	
15	20	print 20

Processes run Concurrently



Solutions

- Locks – Mutual Exclusion
  - Read
  - Write
- Deadlock
- Atomic Operation
  - Other threads see it as happening simultaneously
  - No other thread will see the operation in a partially-completed state
  - No context switch in the middle

Programming Example

- Begin Atomic Operation
- read the value of x
- x = x + 5
- End Atomic Operation
- print x

Summary – Race Condition

- Definition
- Example
- Solutions