Concurrent Programming

- Race Condition
  
  • A fundamental problems that can occurs while executing different threads asynchronously
  
  • A race condition occurs when the meaning of a program depends upon the order in which each thread accesses a shred variable.

```plaintext
i++
1. fetch i value, v, from memory
2. calculate new value v’ based on v
3. store v’ back to the memory allocated to i
```

- Deadlock

  • Dinning Philosophers

```plaintext
while (true) {
  grab left chopstick;
  grab right chopstick;
  eat;
  release left chopstick;
  release right chopstick;
  sleep;
}
```

• A deadlock occurs when a ring of threads comes to a point in the program where each needs a resource from the next thread in the ring in order to continue. No thread can provide the resource because each thread is waiting for another thread.

• Solutions:
  
  - number philosophers, even numbered philosophers pick left first, odd numbered pick right fist
  
  - order chopsticks, force philosophers use chopsticks in a ascending/descending order