CS 361 – Lab 12

Races

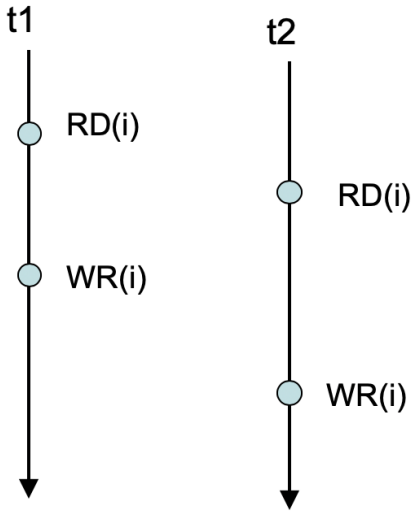
MONDAY, APRIL 29TH 2019.

Data Race

- A data race occurs when 2 instructions from different threads access the same memory location, at least one of these accesses is a write and there is no synchronization that is mandating *any* particular order among these accesses.
- A *data race* happens when there are two memory accesses in a program where both:
  - target the same location
  - are performed concurrently by two threads
  - are not reads (atleast one of them is write operation)
  - are not synchronization operations

Race Condition

- A race condition is a semantic error. It is a flaw that occurs in the timing or the ordering of events that leads to erroneous program behavior
- A race condition is an undesirable situation that occurs when a device or system attempts to perform two or more operations at the same time, but because of the nature of the device or system, the operations must be done in the proper sequence to be done correctly.



Solution

- Lock the variable first. This will ensure that no other thread is using that same variable
- Perform operations on it
- Unlock it so that other threads can use it.

```
Thread 1      Thread 2

 lock(l)       lock(l)
 x=1           x=2
 unlock(l)     unlock(l)
```

Lets end simple

- Given a program(lab12.c).
  - Understand where the race condition is.
    - Run the program multiple times, it should give you different results.
  - Avoid it using suitable locking mechanisms

There is no real ending. It's just the place where you stop the story.

Bid Adieu