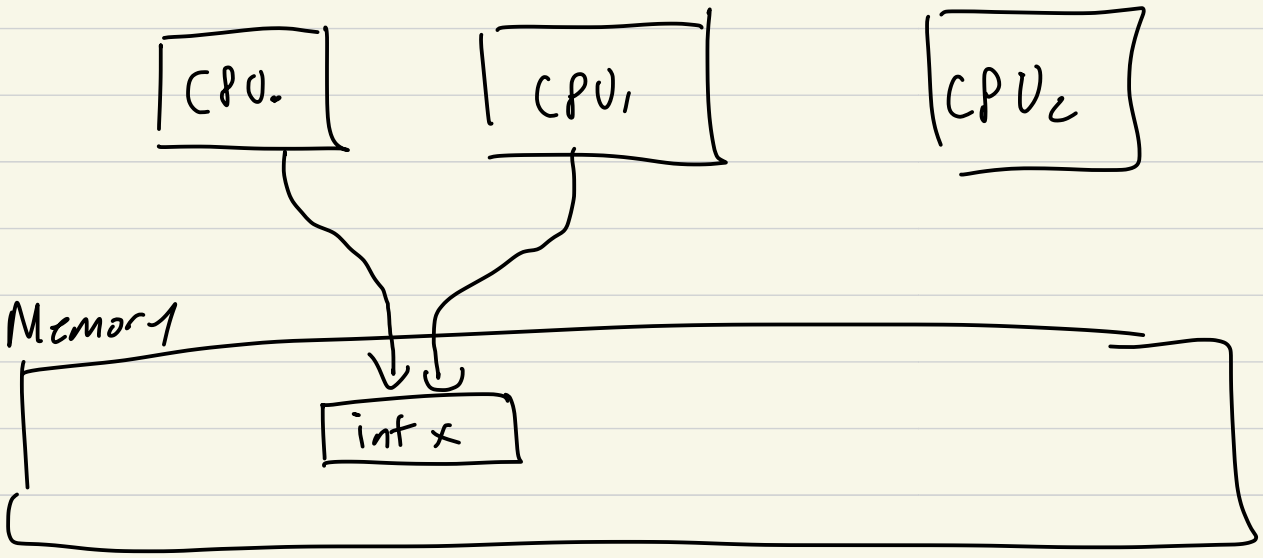
CS 326-01 Concurrency and Locks

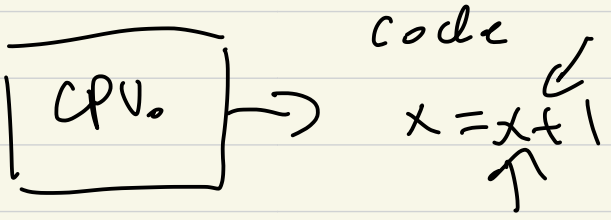
Project 03 ⇒ Queue/List based Scheduling

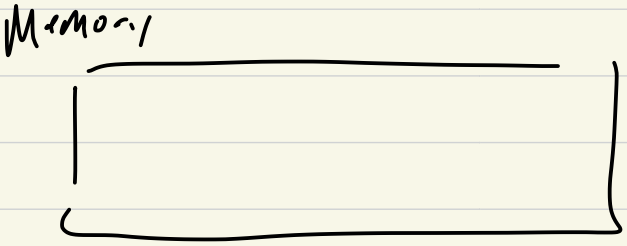
xv6 Book: Chapters 6 and 7

Concurrency

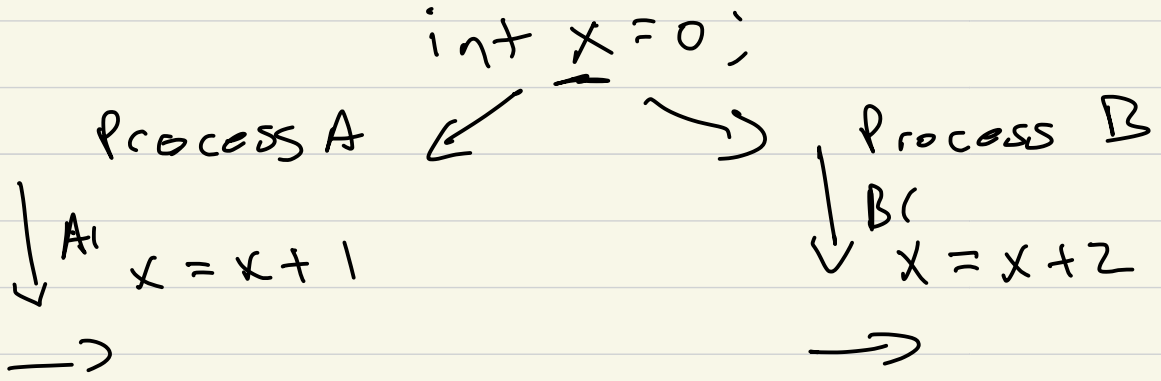


Shared Memory architecture





Race conditions



possible values of x?

#1

A1 x = 1	B1 x = 2
B1 x = 3	A1 x = 3

int x = 0

Process A

```
A1  ld r0, [&x]
A2  addi r0, r0, #1
A3  sd r0, [&x]
```

Process B

```
B1  ld r0, [&x]
B2  addi r0, r0, #2
B3  sd r0, [&x]
```

Possible interleavings

A1 $r_0 \leftarrow 0$	
A2 $r_0 \leftarrow r_0 + 1$	
A3 $x \leftarrow r_0$ $x = 1$	
	B1
	B2
	B3
B1 $r_0 \leftarrow 1$	
B2 $r_0 \leftarrow r_0 + 2$	
B3 $x \leftarrow r_0$ $x = 3$	
	A1
	A2
	A3

$x = 3$

A1 $r_0 \leftarrow 0$	
B1 $r_0 \leftarrow 0$	
A2 $r_0 \leftarrow r_0 + 1$ $r_0 = 1$	
A3 $x \leftarrow r_0$ $x = 1$	
B2 $r_0 \leftarrow r_0 + 1$ $r_0 = 2$	
B3 $x \leftarrow r_0$ $x = 2$	

Race Condition

A1 $r_0 \leftarrow 0$	
B1 $r_0 \leftarrow 0$	
	B2 $r_0 \leftarrow r_0 + 2$ $r_0 = 2$
	B3 $x \leftarrow r_0$ $x = 2$
A2 $r_0 \leftarrow r_0 + 1$ $r_0 = 1$	
A3 $x \leftarrow r_0$ $x = 1$	

Locks

```
int x = 0;
lock x_lock = 0;
```

Process A

```
acquire(&x_lock)
[x = x + 1;]
release(&x_lock)
```

Process B

```
acquire(&x_lock)
x = x + 2;
release(&x_lock)
```

atomically

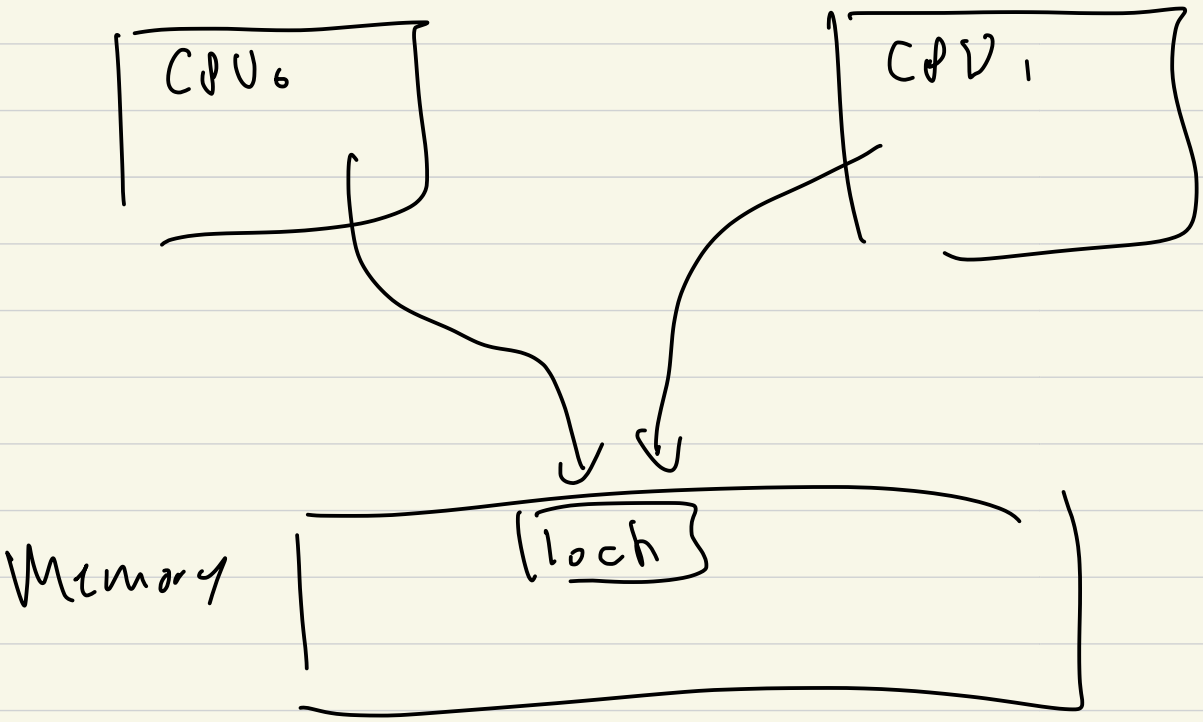
critical section

---

spinlocks

```
acquire(int *lock) {
    [while (*lock == 1);
     *lock = 1]
}
```

X does not work



Atomic swap

address value

```
while ( atomic_swap (lock, 1) == 1);
```