Lecture 32 (Locking and Stuff)

1 xchg Instruction

1. Read memory, modify memory, write register - RMW instruction
2. Get exclusive access with write permission for memory address
3. Perform the RMW operation
4. But do not respond to any other requests from local or other caches or directory while this operation is in progress
5. Respond after execution

2 Spinlock

```
.lock:
    mov r1, 1
    xchg r1, 0[r2]
    cmp r1, 0
    bne .lock
    ret

.unlock:
    mov r1, 0
    xchg r1, 0[r2]
    ret
```

3 Test and Exchange Lock

```
.lock:
    mov r1, 1

.test:
    ld r2, 0[r0]
    cmp r1, 0
    bne .test
```

```
xchg r1, 0[r0]
cmp r1, 0
bne .test
ret
```

4 Atomic Operations

1. Test and set - `tas r1, 0[r0]`
2. Fetch and increment - `fai r1, 0[r0]`
3. Fetch and add - `faa r1, r2, 0[r0]`
4. Compare and Set - `cas r1, r2, r3, 0[r0]`
5. Load linked, store conditional - `ll r1, 0[r0]` and `sc r3, r2, 0[r0]` (store only if value not modified since `ll`)

5 Eliminating Starvation

1. Request T finds another request R that is waiting for a long time
2. T decides to help R
3. This is an altruistic algorithm

6 Consensus Number

Maximum number of threads that can solve a problem using a wait-free algorithm