

# Locks and Barriers

Note Title

3/5/2009

efficient locks

- only one proc. to try to get lock on release

Ticket lock

two counters per lock

next-ticket  
now-serving





# Barriers

can build hardware barriers

IBM P1m Gene

sm usually uses locking mechanisms

## Centralized Barrier Alg

```
struct bar_type {  
    int counter;  
    struct lock_type lock;  
    int flag = 0;  
} bar_name;
```

```
BARRIER (bar_name, p) {  
    LOCK(bar_name.lock);  
    if (bar_name.counter == 0)  
        bar_name.flag = 0; /* reset flag if first to reach */  
    mycount = bar_name.counter++; /* mycount is private */  
    UNLOCK(bar_name.lock);  
    if (mycount == p) { /* last to arrive */  
        bar_name.counter = 0; /* reset for next barrier */  
        bar_name.flag = 1; /* release waiters */  
    }  
    else while (bar_name.flag == 0) ; /* busy wait for release */  
}
```

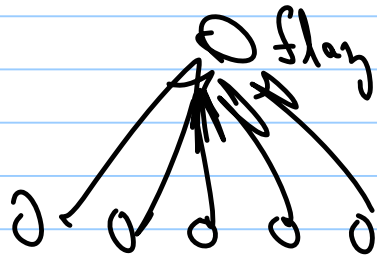
Watch  
term.  
cond.

# Working Centralized Barrier

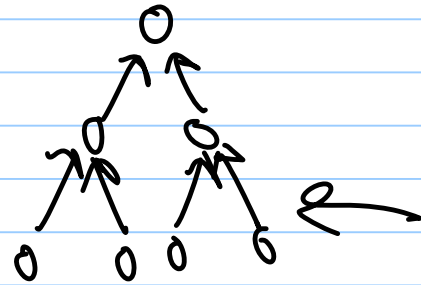
sense reversal

```
BARRIER (bar_name, p) {
    local_sense = !(local_sense);    /* toggle private sense variable */
    LOCK(bar_name.lock);
    mycount = bar_name.counter++;    /* mycount is private */
    if (bar_name.counter == p) {
        UNLOCK(bar_name.lock);
        bar_name.counter = 0;
        bar_name.flag = local_sense; /* release waiters*/
    }
    else {
        UNLOCK(bar_name.lock);
        while (bar_name.flag != local_sense) ;
    }
}
```

Improved Alg. is tree org.



$O(p)$



$O(\lg p)$

still direct time on bus