

Operating System 2016-1 Project 2

# *Rotation Lock*

Team 1

2014-19768 Sungyun Hur

2014-17184 Yeonwoo Kim

2013-13494 Eunhyang Kim

# *Design*

## Data structure

lock\_list

```
struct lock_list{  
    struct rotation_range range;  
    struct list_head lst;  
    pid_t pid;  
    int rw;  
};
```

## *Design*

Global variables

Acquired list & Waiting list

```
struct list_head lock_waiting;  
struct list_head lock_acquired;
```

Current rotation degree

```
Struct dev_rotation curr_rot;
```



## *Design*

### Functions & System calls

#### set\_rotation

Change current degree.

Wake tasks in waiting list according to the Policy. Count the number meanwhile.

Return the number of tasks waked.

#### exit\_rotlock

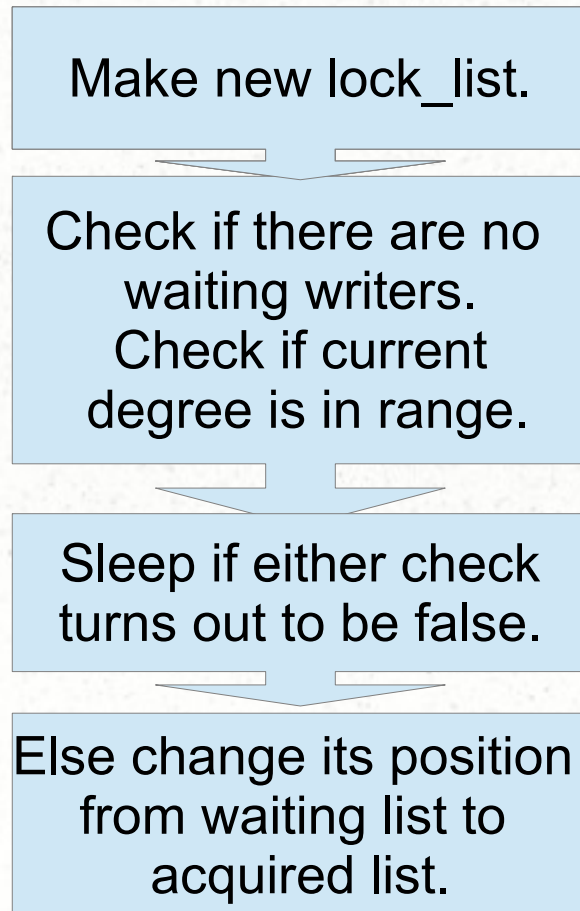
Find itself in waiting list and lock list.

Delete itself in the list it is currently in.

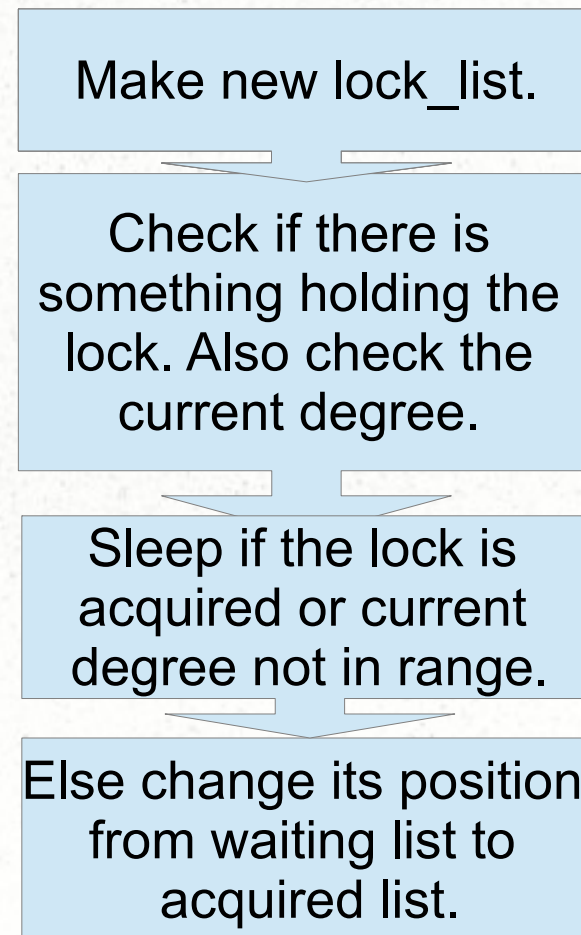
## *Design*

### Functions & System calls

#### **rotlock\_read**



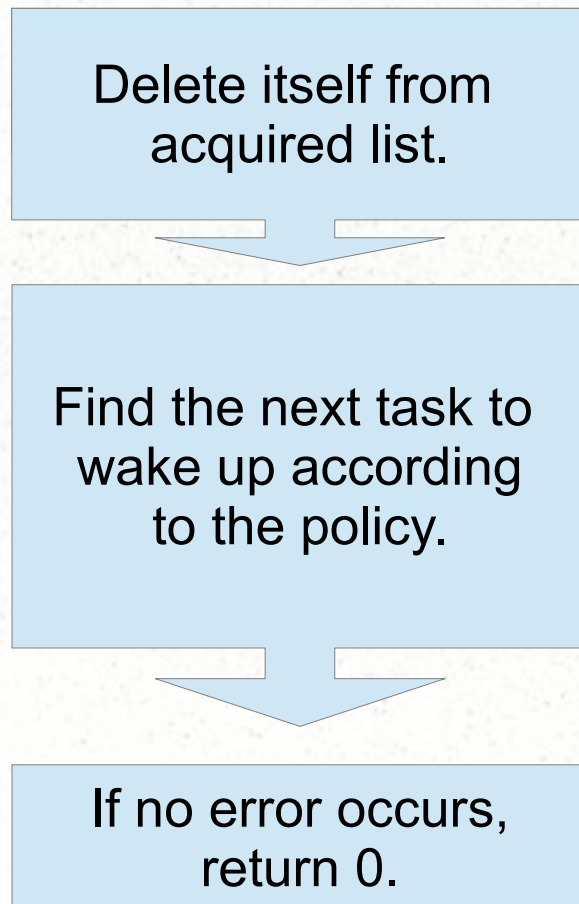
#### **rotlock\_write**



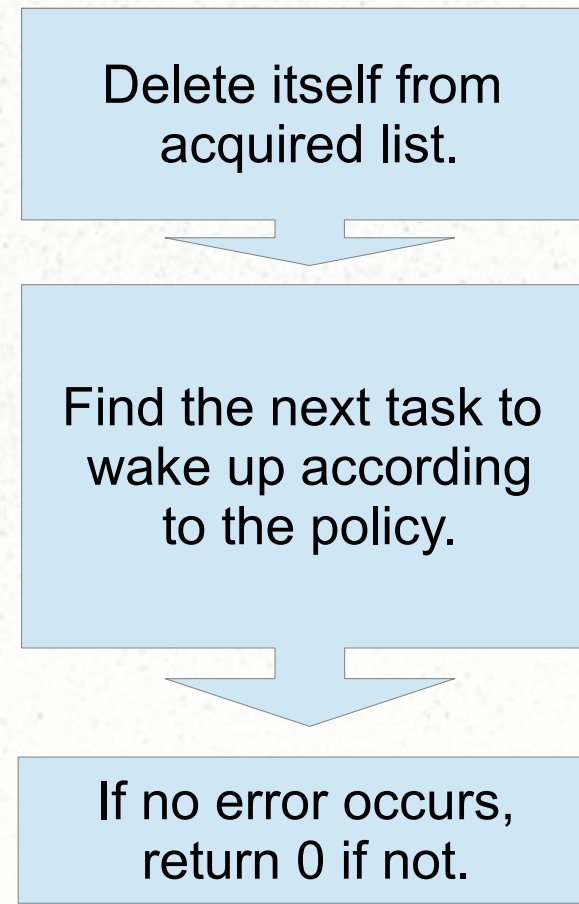
## *Design*

Functions & System calls

**rotunlock\_read**

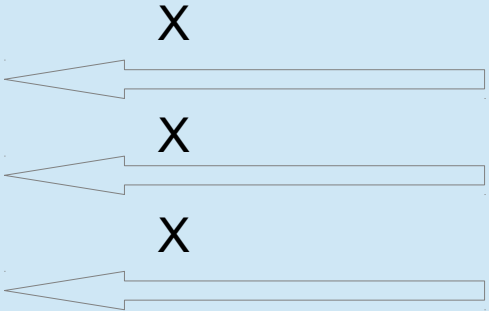


**rotunlock\_write**





## *Policy*

Acquired List		Waiting List
1. Writer		1-1 Readers 1-2 Readers & Writers 1-3 Writers
2. Readers		2-1 Readers 2-2 Readers & Writers 2-3 Writers
3. (None)		3-1 Readers 3-2 Readers & Writers 3-3 Writers

# *Demo*

<https://youtu.be/iQC9VpY1Ba8>



## *We learned ..*

- \* how locks work
- \* how to think concurrently
- \* to think about all possible situations before coding.