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

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

Make new lock_list.

Check if there are no waiting writers.
Check if current degree is in range.

Sleep if either check turns out to be false.

Else change its position from waiting list to acquired list.

rotlock_write

Make new lock list.

Check if there is something holding the lock. Also check the current degree.

Sleep if the lock is acquired or current degree not in range.

Else change its position from waiting list to acquired list.

Design

Functions & System calls rotunlock_read

rotunlock_write

Delete itself from acquired list.

Delete itself from acquired list.

Find the next task to wake up according to the policy.

Find the next task to wake up according to the policy.

If no error occurs, return 0.

If no error occurs, return 0 if not.

Policy

Acquired List		Waiting List
	X	1-1 Readers
1. Writer	X	1-2 Readers & Writers
	X	1-3 Writers
2. Readers	All Readers	2-1 Readers
	X	2-2 Readers & Writers
	X	2-3 Writers
3. (None)	All Readers	3-1 Readers
	One writer	3-2 Readers & Writers
	One writer	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.