OS Project2

Rotation Lock

TEAM6

정석원
박상혁
강지원

INDEX

001 Rotation Lock

002 Function

003 Test program

- selector.c
- trial.c

004 Demo

001 Rotation Lock Structure

1. **rot_lock_t: struct that contains rotational lock information**

```c
struct rot_lock_t {
    int degree;
    int range;
    int type;
    pid_t pid;
    struct list_head loc;
};
```

2. **pend_head: list_head of pend list**

```c
static struct list_head pend_head = LIST_HEAD_INIT(pend_head);
```

- pend list is a list used to store rot_lock_t which is in sleep state

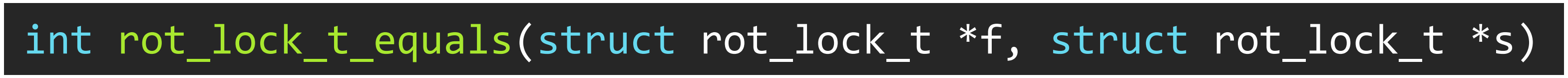
3. **acq_head: list_head of acq list**

```c
static struct list_head acq_head = LIST_HEAD_INIT(acq_head);
```

- acq list is a list used to store rot_lock_t which is running on threads

002 Function

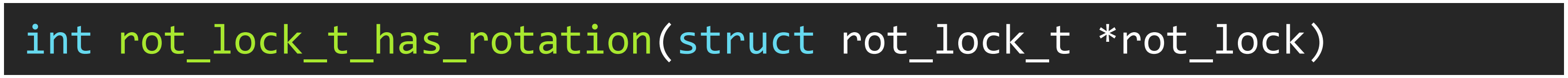
1. rot_lock_t_equals



```
int rot_lock_t_equals(struct rot_lock_t *f, struct rot_lock_t *s)
```

- Checking if two rot_lock_t are equal or not.

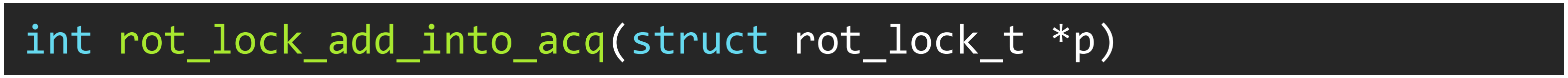
2. rot_lock_t_has_rotation



```
int rot_lock_t_has_rotation(struct rot_lock_t *rot_lock)
```

- Checking if rot_lock_t has rotation.
- Call this function AFTER grabbing the rot_spinlock.

3. rot_lock_add_into_acq



```
int rot_lock_add_into_acq(struct rot_lock_t *p)
```

- This function insert new rot_lock_t into acq_head.
- If success, return1. Otherwise return 0.

002 Function

4. lock_available

```
int lock_available(struct rot_lock_t *p)
```

- This function checks if new rot_lock_t can be inserted into acq_list.
- If possible, return 1. Otherwise, return 0.

5. rot_lock_t_remove

```
int rot_lock_t_remove(struct rot_lock_t *p)
```

- This function checks if the same rot_lock_t exists in qcq_list.
- If it exists, delete and return 1.

6. wake_up_candidate

```
int wake_up_candidate(void)
```

- This function will find proper lock and wake such process.
- Call this function AFTER grabbing the rot_spinlock.

002 Function

7. sys_set_rotation

```
int sys_set_rotation(int degree)
```

- This function checks validate input degree and change rotation degree.

8. sys_rotlock_read

```
int sys_rotlock_read(int degree, int range)
```

- Rotational read lock.

9. sys_rotlock_write

```
int sys_rotlock_write(int degree, int range)
```

- Rotational write lock.

002 Function

10. sys_rotunlock_read

```
int sys_rotunlock_read(int degree, int range)
```

- Rotational read unlock.

11. sys_rotunlock_write

```
int sys_rotunlock_write(int degree, int range)
```

- Rotational write unlock.

12. exit_rotlock

```
int exit_rotlock(void)
```

- Clean acq_list and pend_list.

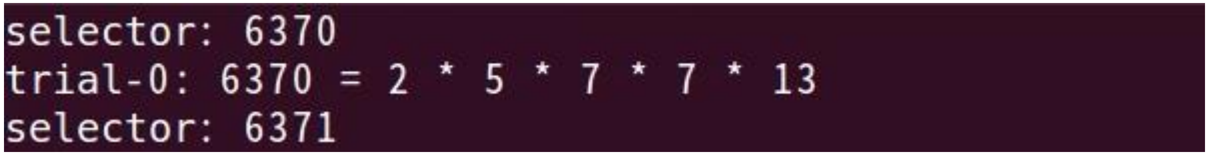
003 Test program

1. **selector**

- This program writes integer from the argument to a file called 'integer' when the device is positioned between degree 0 to 180.

2. **trial**

- This program reads the integer from a file called 'integer' and calculates the prime number factorization of the integer, and writes the result using standard output.



```
selector: 6370
trial-0: 6370 = 2 * 5 * 7 * 7 * 13
selector: 6371
```

004 Demo

https://youtu.be/mSoYTd3Wkdg