Sri Hari Malla - CS19BTECH11039
Prof : Dr. Sathya Peri
Parallel and Concurrent Programming
Indian Institute of Technology Hyderabad

THEORY ASSIGNMENT - I

Sri Hari Malla - CS19BTECH11039

January 23, 2022

Filter Lock :

We know that filter locks are the adaption of peterson lock in which the support is added for n threads where `n >= 3`. While the peterson's lock is starvation free which implies deadlock free, this algorithm also follows the same.

In this analogy we have two important properties

- At least one thread trying to enter level `L` succeeds.
- If more than one thread is trying to enter level `L`, then at least one is blocked.

However the correctness of the filter lock algorithm is discussed in the class.

RTP : For j between 1 and n - 1 there are at most n - j + 1 threads at level j

Let's take the induction on j.

- **Base Case :** j = 1, the case is trivial. n threads will be passing at the 1st level.
- At this level there are n threads, but according to the above mentioned properties, atleast one is blocked from entering further levels.
- **Induction Hypothesis :** Let's assume it works for j+1 without loss of generality.
- It follows the same properties, so at jth level, n-j+1 threads enter and n-j threads enter the next level.
- By applying the formula (n-j-1+1) n-j threads will hit the j+1th level. Hence it is true for j+1 while its true for j.
- **Check :** Consider j = n-1, which shows 2 threads will be at the n-1 th level.

- By following the properties mentioned, while at least one thread succeeds and at least one thread gets blocked. Resulting only one thread to enter in the ultimate level.

Now it is proven that for j between 1 to n-1, there are at most n-j+1 threads at that level considering the property that at least one thread succeeds and at least one thread gets blocked. If we are able to prove that at least one thread gets blocked in entering the next level, then the lemma will be proved sufficiently.

Assumption : Let's assume that there are at least n-j+2 threads at some level j(one of the level didn't block any thread)

Let T be the last thread at level j to update the `victim[j]`. Since T is the last thread to update the victim, for any other thread R at level j, (T being the last thread, R being any remaining thread excluding T)

```
code block to look at:
    level[me] = i; victim[i] = me; // me is threadID
    while ((E k != me) (level[k] >= i && victim[i] == me)) {};
```

$$\text{order : write}_R(\text{victim[j]}) \rightarrow \text{write}_T(\text{victim[j]})$$

clearly from the code section, one can see that R writes level[R] before it writes to victim[j], so

$$\text{order : write}_R(\text{level[R]=j}) \rightarrow \text{write}_R(\text{victim[j]} \rightarrow \text{write}_T(\text{victim[j]})$$

From the code section, one can see that T reads level[R] after it writes to victim[j], so

$$\text{order : write}_R(\text{level[R]=j}) \rightarrow \text{write}_R(\text{victim[j]} \rightarrow \text{write}_T(\text{victim[j]} \rightarrow \text{read}_T(\text{level[R]}))$$

Because R is at level j, every time T reads level[R], it observes a value greater than or equal to j implying that T couldn't have completed its waiting loop. So, at least one thread should be getting blocked at each level suggesting that n-j+1 threads will be present at the jth level. Hence proved using contradiction for the statement, at least one thread gets blocked at each level.

By both initial induction and contradiction, we have enough evidence that the statement given is True.

Therefore, for j between 1 to n-1, there are at most n-j+1 threads at level j.

LaTeX generated document

THE END