

Lecture 13 home work

Read the "Fast mutual exclusion for uniprocessors" paper.

Assignment

Assume you are giving the following code for an implementation of a list:

```
struct List {
    int data;
    struct List *next;
};

List *list = 0;

insert(int data) {
    List *l = new List;
    l->data = data;
    l->next = list;
    list = l;
}

fn() {
    insert(100);
}

main() {
    thread_create(..., fn);
    thread_create(..., fn);
    thread_schedule();
}
```

In this code multiple threads insert into a list concurrently. Will this code work correctly?

Give an implementation of *insert* using restartable atomic sequences? (Please hand in your answer at the beginning of lecture.)