OS LAB: ASSIGNMENT 4
REPORT

Group number: 30

Group members

Prakhar Singh	20CS10045
Akash Das	20CS10006
Rohit Kumar Prajapati	20CS30041
Saras Umakant Pantulwar	20CS30046

Data Structures:

We have used two datastructures one is Action which is used for different actions, it has following data members:
int user_id,
int action_id,
int action_type,
time_t action_time

Other structure used is adj_list it has following data members:
int id,
int *neighbours (it contains the list of all neighbours (node id) of the current node)
int num_neighbours (it contains number of neighbours of the current number)
int count[3] (it is used to recognize type of action i.e {0: post, 1: comment, 2: like})
queue<Action> wall_queue (contains actions of this nodes)
queue<Action> feed_queue (contains actions of the neighbours of this node)
int reading_order (defines reading order i.e for priority 0 and for chronological 1)
pthread_mutex_t feed_lock (pthread_mutex_lock allows threads to acquire a mutual exclusion lock, which is used to protect shared resources from concurrent access.)
pthread_cond_t feed_cond (pthread_cond_t to allow threads to wait for a condition to be true before proceeding with their execution.)

Design decision for queue sizes:

For feed queue and wall queue we have used queue container from STL library. Queue is a container adapter that operates in a first-in, first-out (FIFO) order which helps in maintaining chronological order.

Design of how locks are used to ensure no race condition and concurrency:

Whenever there is a possibility that we are writing or reading to a data structure that is shared, we have used mutex locks there. Hence, we have used eight locks. A detailed Overview is as follows:

- UserSimulator:
    1. Before writing to the shared queue
- ReadPost:
    1. Before reading from the feed queue set
    2. Before reading from the feed queue for new actions
- PushUpdate:
    1. Reading from shared Queue
    2. Before writing to the feed queue of a node
    3. Before writing to the shared feed queue set
- Before writing to the terminal or the log file
- Before writing to the feed queue