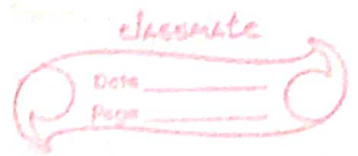


Shubhan Singh

2022300118

Comps B



Exp 4 report : Multithreading in C

Learnings: In this experiment, we learnt about threads in C. C programs can be parallelised by using threads, made using functions in the `<pthread.h>` library (short for POSIX threads). Pthreads can be created using the `pthread_create()` function, which starts a new thread using an object of the type `pthread_t`. Threads can run functions of return type `void*`, ~~with~~ having at most one argument of type `void*`.

Functions like `pthread_join()` and `pthread_exit()` assist in control of the program. If multiple pthreads are ~~not~~ operating on the same ^{section of} memory, ~~the~~^a `pthread_mutex_t` can be used to lock the critical section so that it is only accessed by one ~~program~~ thread at a time, ensuring consistency in the output of the program. The ~~pt~~ mutex (mutual extension) comes with function like `pthread_mutex_lock()` and `pthread_mutex_unlock()`. We used mutexes in the second part of the experiment.

Errors encountered: ~~We~~ ~~Unpredictable~~ Unpredictable behaviour was observed at first when the same object was accessed by multiple threads without locking. All elements ^{pointers} needed to be type casted to pass them into pthreads, this also led to some error.