

Some Practice Problems for Week 4

You may try to answer the following questions only after studying all lecture-contents of Week 4. Some of these practice problems will be discussed during the online session.

Q1. Why are modern computers multi-core?

Q2. What is the cache coherence problem? Describe a protocol to resolve this problem.

Q3. Give an example where we must have a concurrent program instead of a sequential program.

Q4. Describe how to add two matrices using concurrent threads.

Q5. Describe how to transpose a matrix using concurrent threads.

Q6. Can you perform the above matrix transpose operation 'in-place', i.e., without creating or allocating any noticeable extra memory?

Q7. Describe how to compute matrix multiplication using concurrent threads.

Q8. Is it possible to compute (and speedup) *factorial* computation using parallel threads?

Q9. Give an example of a computational problem where it is not possible to perform parallel computation.

Q10. Explain why does this program compute unreliably?

```

void *functionC();
int counter =0;
main() {
    int rc1,rc2;
    pthread_t thread1,thread2;// Two threads execute functionC()
    pthread_create(&thread1,NULL,&functionC,NULL);
    pthread_create(&thread2,NULL,&functionC,NULL);
    pthread_join(thread1,NULL);
    pthread_join(thread2,NULL);
    return0;
}
void *functionC(){
    counter++;
    printf("Counter value: %d\n", counter);
}

```

Q11. Give an example where sharing a pointer to a stack variable by several concurrent threads leads to unreliable program execution.