

333 Section 10 SOLUTION - Concurrency and pthreads

Welcome back to section! :)

Exercise 1

How do threads and processes compare in these categories? Circle the correct answer.

	Multiple Threads	Multiple Processes
Memory / Address Space	Shared	Separate
Stack	Separate	Separate
Heap	Shared	Separate
Communication	Easy	Difficult
Synchronization	Difficult	Not Applicable
Context-switch "Weight"	Light	Heavy
Crash Tolerance (what happens to the others when one dies?)	Not Tolerant	Tolerant

Exercise 2

It's payday! It's time for UW to pay each of the 333 TAs their monthly salary. Each of the TA's bank account is inside the `bank_accounts[]` array and the person who is in charge of paying the TAs is a 333 student and decided to use pthreads to pay the TAs by adding 1000 into each bank account. Here is the program the student wrote:

```
// Assume all necessary libraries and header files are included
const int NUM_TAS = 10;

static int bank_accounts[NUM_TAS];
static pthread_mutex_t sum_lock;

void *thread_main(void *arg) {
    int *TA_index = static_cast<int*>(arg);

    pthread_mutex_lock(&sum_lock);
    bank_accounts[*TA_index] += 1000;
    pthread_mutex_unlock(&sum_lock);

    delete TA_index;
    return nullptr;
}

int main(int argc, char** argv) {
    pthread_t thds[NUM_TAS];
    pthread_mutex_init(&sum_lock, NULL);

    for (int i = 0; i < NUM_TAS; i++) {
        int *num = new int(i);
        if (pthread_create(&thds[i], nullptr, &thread_main, num) != 0) {
            /*report error*/
        }
    }

    for (int i = 0; i < NUM_TAS; i++) {
        cout << bank_accounts[i] << endl;
    }

    pthread_mutex_destroy(&sum_lock);
    return 0;
}
```

(see next page)

a) Does the program increase the TAs' bank accounts correctly? Why or why not?

No, it's not correct. It requires main to call `pthread_join` to wait for each thread to finish before exiting the main program.

`pthread_exit()` will let a child thread leave its parent, but it needs to be used in conjunction with `pthread_join` in order to check the results of the child thread.

b) Could we implement this program using processes instead of threads? Why would or why wouldn't we want to do this?

We could, but doing so would require some way for the processes to communicate with each other so that the data structure can be "shared" (remember that inter-process communication can be difficult and time consuming).

It is much easier to just use threads since each thread could directly access the data structure.

c) Assume that all the problems, if any, are now fixed. The student discovers that the program they wrote is kinda slow even though it's a multithreaded program. Why might it be the case? And how would you fix that?

Only one thread can increase the value of one account at a time and there is no difference from incrementing each account sequentially because we only have a single lock on this line for every single thread to share.

To fix this, we can have one lock per account so that multiple threads can increment the account at the same time. (An alternative solution is to just not use locks as well since the threads made will not conflict with each other, but we should aim for safe options for the bank accounts)

```
thread_mutex_lock(&sum_lock);  
bank_accounts[*TA_index] += 1000;  
pthread_mutex_unlock(&sum_lock);
```

Will turn to...

```
thread_mutex_lock(&acct_lcks[*TA_index]);  
bank_accounts[*TA_index] += 1000;  
pthread_mutex_unlock(&acct_lks[*TA_index]);
```

Exercise 3

Journaling Activity! Take about 5 minutes with yourself or your neighbor to journal about your experiences with homework and exercises this quarter.

Guiding Questions:

- What were a couple main themes or takeaways from each homework?
- What lessons (e.g., concepts, tools, habits, etc.) did you learn from each homework?
- If you could change something about a certain homework or exercise what would it be?
 - Let us know in the course evals :)

This is intended to be an open-ended, “experiential” question. What did you **personally** take away from the homework and exercises?

The point of the question is to give you a chance to take a step back and think about what you’ve learned through the homework and exercises.

Also for more practice answering experiential questions :)