SYSC 4001: Operating Systems
Programming Assignment 3:
Posix Threads
Due: March 6, noon

Assignment Description

This program sorts strings using "enzymes". An enzyme is a function that sorts two consecutive characters. We define one enzyme per pair of consecutive characters; these enzymes working together in parallel can sort the entire string. We use `pthreads` to instantiate and parallelize the enzymes. Unfortunately, this program doesn't seem to be working correctly. That's where you come in.

Before you edit the code, read through it, and consider these questions:

1) Why would this application be difficult to write using multiple processes instead of threads.
2) What is the significance of `'workperformed'`? How is it used?
3) Explain exactly what is the type of 'fp' in the following declaration:
   ```
     void *(*fp)(void *)
   ```

Now, to fix the program:

1) The function `run_enzyme()` needs to be created. Please see the notes inside `enzyme.c`.
2) The function `'make_enzyme_threads'` has a memory bug. Fix this by simply re-ordering the lines in this function. It is simple fix but may take a while for you to find it.
3) The function `'join_on_enzymes'` is incomplete. Read the relevant man pages and figure out how the function is supposed to work. Then insert the correct code snippets into `'whatgoeshere'`.
4) Your programming work can be considered complete when you have completed the above and all of the tests pass.

Testing

```
make test
./enzyme -test -f0 all
Running tests...
 1.make                ::pass
 2.sort                ::pass
 3.pleasequit1         ::pass
 4.pleasequit0         ::pass
 5.swap1               ::pass
 6.swap2               ::pass
 7.swap3               ::pass
 8.run_enzyme          ::pass
 9.join                ::pass
 10.cancel             ::pass
```

You may also want to experiment with the cancel function

```
./enzyme Cba
```

```
./enzyme CBA
```

Additional Questions to Ponder

1) Why do we not detach any of the enzyme threads? Would the program function if we detached the sleeper thread?
2) Why does the program use `sched_yield`? What happens if this is not used? Will the swap counts always be identical?
3) Threads are cancelled if the string contains a 'C' e.g. "Cherub". Why do we not include cancelled threads when adding up the total number of swaps?
4) What happens when a thread tries to join itself? (You may need to create a test program to try this). Does it deadlock? Or does it generate an error?
5) Briefly explain how the sleeper thread is implemented.
6) Briefly explain why PTHREAD_CANCEL_ASYNCHRONOUS is used in this assignment.
7) Briefly explain the bug in Part II, #2 above.

Submission Requirements

Submit your assignment using `cuLearn`, under `Assignment 3 Submission` for the course. You should submit only your source code solution to the assignment, i.e., `enzyme.c`.

Marks will be based on:

- Completeness of your submission
- Correct solution to the problem
- Adhering to the submission requirements
- Following good coding style
- Sufficient and high-quality in-line comments