



## Lab

## 8:

## Basic

# multithreading

Due by midnight, Wednesday 11/21

Play with threads to search through an array faster!!!

## Getting started

[Right click and download this link.](#) This is the skeleton of your lab.

Don't forget to use `-lpthread` as a flag to gcc when compiling.

**Please be sure to put your name at the top.**

This program takes two arguments: the name of a binary data file containing an array of ints, and a value to search for. It will search for the value in the file, and say where it found it, or if the value doesn't exist.

I have given you a file you can access to test your program with. It's a big array (1,000,000 items) containing the values 0 through 999,999 in a random order.

Since it's a big file, don't bother copying it to your AFS space (but you can copy it to your own computer if you want). Instead, run the program like so:

```
./lab8 ~/jfb42/public/cs449/data.bin 10
```

to search the file for the value 10.

---

## What to do

This lab is just to get you a little more comfortable with using the pthread library, nothing too scary.

The idea is this: we have an unordered array and we want to search through it. Linear search is slow, but we can speed it up by using multiple threads (because thoth has kernel threading with hardware-accelerated threads). So, if we are using 8 threads, we should be able to do it in 1/8th the time.

*In reality, the overhead of making and switching between threads makes it a little slower than that, but hey, that's life.*

I already wrote the code to "slice up" the array and give each thread a piece of of it. So if the array is 80 items long, thread 0 will search items 0..9; thread 1 will search 10..19; etc.

**The functions you have to implement are at the bottom.** The comments describe what they should do.

## Notes

- Look at the SearchContext struct!! I put comments describing the fields there.
- Each function is only a few lines long. Some of them really are "that easy."
- Refer to the threading examples I gave in class on the examples page, particularly `21_thread_test.c`.
- Here are examples of how it should work once you've done it right:

```
$ ./lab8 ~jfb42/public/cs449/data.bin 0
thread 4 found 0 at index 527023
$ ./lab8 ~jfb42/public/cs449/data.bin 1
thread 0 found 1 at index 93
$ ./lab8 ~jfb42/public/cs449/data.bin 2
thread 4 found 2 at index 559093
$ ./lab8 ~jfb42/public/cs449/data.bin 3
thread 5 found 3 at index 702520
$ ./lab8 ~jfb42/public/cs449/data.bin -1
value -1 was not found.
$ ./lab8 ~jfb42/public/cs449/data.bin 1000000
value 1000000 was not found.
```

---

## Submission

Please remove all the comments I put in the file before you submit.

[Then submit as usual.](#)