

Exercises for Hands-on 3: Valgrind

Make sure to fill in questions in the space provided underneath the question text.

Question 1: Study helgrind's output and the ph.c program. Clearly something is wrong with put() on line 61.

- **a)** Identify a sequence of events that can lead to keys missing for 2 threads.
- **b)** Using that sequence, explain why there missing keys with 2 or more threads, but not with 1 thread. Your explanation should refer to the given code and make use of theory to justify your answer.

Write or type your answers to both parts of questions 1 below. Your answers should not extend onto a second page.

Fixing the error

To avoid this sequence of events, insert lock and unlock statements in put so that the number keys missing is always 0. The relevant pthread calls are (for more see the manual pages, man pthread):

```
pthread_mutex_t lock;    // declare a lock
pthread_mutex_init(&lock, NULL); // initialize the lock
pthread_mutex_lock(&lock); // acquire lock
pthread_mutex_unlock(&lock); // release lock
```

The function get() has an example use of locks, and main initializes it.

Modify ph.c to make it correct and recompile with gcc. Test your code first with 1 thread, then test it with 2 threads. Is it correct (i.e. have you eliminated missing keys)? Check the correctness using helgrind. (Note that valgrind slows down ph a lot, so you may want to modify NKEYS to be some small number.)

Question 2: Describe your changes and argue why these changes ensure correctness. (Do not include the entire ph.c code, but do tell us which lines you have added and feel free to include the put code in your explanation.)

Write or type your answers to question 2 below. Your answers should not extend onto a second page.

Question 3: Is the two-threaded version faster than the single-threaded version in the put phase? Report the running times for the two-threaded version and the single-threaded version. (Make sure to undo your `NKEYS` change.)

Question 4: Most likely (depending on your exact changes) ph won't achieve any speedup. Why is that?

Write or type your answers to questions 3 and 4 below. Your answers should not extend onto a second page.

Living Dangerously

Remove the locks from `get()` in `ph.c`, recompile, and rerun the program.

Question 5: You should observe speedup on several cores for the `get` phase of `ph`. Why is that?

Question 6: Why does `valgrind` report no errors for `get()`?

Question 7: How long did this assignment take you?

Write or type your answers to questions 5-7 below. Your answers should not extend onto a second page.