CS 491/521 Parallel Programming
Homework 2
Pthreads Programming
Due: 10/25, Tuesday, 11:59 pm

Read the research paper discussed in the class (also included in the reference at the end), and implement a simplified version of the worker program and the checker program using Pthreads. The basic steps and requirements are as follows:

1. (5 points)
   write a simple script to create a device file using the following *dd* command:
   dd if=/dev/zero of=./device-file bs=1M count=128

2. (80 points)
   write a multi-threaded worker program to write records to random addresses within the device file; your worker program should have the following features:
   (1) take a full file name (including the path) as input from the command line argument (tips: the file is the device file you created in the previous step);
   (2) allow the user to specify the number of worker threads (tips: the number of threads can be read from either a configuration file or the command line argument);
   (3) support at least two worker threads and at most 100 worker threads;
   (4) each work thread should keep doing the following work:
      a. use a random number generator to generate a random address (within the range of the device file);
      b. generate a record; the exact format of the record is up to you, but it should include four pieces of information at least: the worker thread's ID, the address (generated in step *a*) for the record, the record ID (i.e., the number of records generated by the worker thread so far), and the checksum of the whole record; the size of the record should be 8KB;
      c. use *pwrite()* to write the record (generated in step *b*) to the random address (generated in *a*); each write should be synchronous write;
      d. go back to *a*;

3. (5 points)
   write a simple script to run the worker program for a specified amount of time and then kill the worker program; the running time (in seconds) should be specified as a command line argument to the script;
4. (10 points)
   write a single-threaded checker program to read records from the device file; your checker program should have the following features:
   (1) take a file name (including the path) as a command line argument (this is the location of the device file);
   (2) read through the whole device file, print the following information for each record on the file:

worker thread's ID,
the address of the record,
the record ID,
the checksum of the record;

(3) after reading the whole device file, print the following information:
total number of worker threads,
total number of records written by each worker thread,
whether the last record of each thread is intact or not.

Note that the above features are the basic requirements. You are encouraged to include more features (you need to explain your design choices and justify why the features are desired).

Write a report about how you design the worker and the checker (e.g., your record format, your random number generator, your checksum algorithm, etc). Your report should also include instructions for the TA about how to compile and execute your code on a CS departmental machine.

Submit your report as well as your code to Canvas.

Reference:

Understanding the Robustness of SSDs under Power Fault
Mai Zheng, Joseph Tucek, Feng Qin, and Mark Lillibridge
In Proceedings of the 11th USENIX Conference on File and Storage Technologies (FAST'13), 2013
http://www.cs.nmsu.edu/~mzheng/publications/2013_FAST_PowerFaultSSD.pdf