#include <unistd.h>

#include <fcntl.h>

#include <stdio.h>

#include <dirent.h>

#include <stdlib.h>

#include <pthread.h>

unsigned int globalSeed;

volatile int currentFileNumber = 0;

pthread\_mutex\_t mutexA = PTHREAD\_MUTEX\_INITIALIZER;

struct thread\_args

{

char \*d;

int num\_files;

int num\_ints;

};

void \*createFiles(void \*ptr) {

struct thread\_args \*args = ptr;

int num\_files = args->num\_files;

char \*d = args->d;

int num\_ints = args->num\_ints;

for (int i=0; i < num\_files; i++) {

char \*filepath = (char\*)malloc(80 \* sizeof(char));

pthread\_mutex\_lock(&mutexA);

sprintf(filepath, "%s/unsorted\_%d.bin", d, currentFileNumber);

currentFileNumber += 1;

int fd = open(filepath, O\_RDWR|O\_CREAT, 0644);

pthread\_mutex\_unlock(&mutexA);

for (int j=0; j<num\_ints; j++) {

u\_int32\_t n = rand\_r(&globalSeed);

write(fd, &n, sizeof(n));

}

close(fd);

}

return 0;

}

int main(int argc, char \*\*argv) {

char \*d = argv[1];

char \*ptr;

if (d == NULL) {

perror("Location argument not passed, closing the program...");

return -1;

}

if (argc < 5) {

perror("Not all arguments passed, Follow the pattern\n./filecreator <path> <number files> <number int> <number threads>\nclosing the program...");

return -1;

}

long f = strtol(argv[2], &ptr, 10);

long r = strtol(argv[3], &ptr, 10);

long t = strtol(argv[4], &ptr, 10);

int result = access(d, W\_OK);

if (result == -1) {

perror("Could not locate directory, closing the program...");

return -1;

}

pthread\_t w1[t];

for (int i=0; i < t; i++) {

struct thread\_args \*args = malloc(sizeof \*args);

args->d = d;

args->num\_files = f/t;

if (i== t-1) {

args->num\_files = f/t + f%t;

}

args->num\_ints = r;

pthread\_create(&w1[i], NULL, &createFiles, args);

}

for (int i=0; i < t; i++){

pthread\_join(w1[i], NULL);

}

return 0;

}