#include <unistd.h>

#include <fcntl.h>

#include <stdio.h>

#include <dirent.h>

#include <stdlib.h>

#include <string.h>

#include <sys/stat.h>

void swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void sort\_array(int numbers[], int n) {

int i, j;

for (i = 0; i < n-1; i++)

for (j = 0; j < n-i-1; j++)

if (numbers[j] > numbers[j+1])

swap(&numbers[j], &numbers[j+1]);

}

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

char \*dir = argv[1];

if (dir == NULL) {

fprintf(stderr, "No source folder provided, exiting the code");

return -1;

}

int source\_dir\_exists = access(dir, F\_OK);

if (source\_dir\_exists == -1) {

fprintf(stderr, "Could not locate the folder, exiting the code");

return -1;

}

int src\_dir\_permission = access(dir, R\_OK);

if (src\_dir\_permission == -1) {

fprintf(stderr, "Do not have read permission to folder, exiting the code");

return -1;

}

char \*dest\_folder = (char\*)malloc(strlen(dir) + 10);

sprintf(dest\_folder, "%s/sorted/", dir);

int destination\_dir\_exists = access(dest\_folder, F\_OK);

if (destination\_dir\_exists == -1) {

fprintf(stderr, "Destination directory %s does not exist, creating it\n", dest\_folder);

mkdir(dest\_folder, 0777);

}

int dest\_dir\_permission = access(dest\_folder, W\_OK);

if (dest\_dir\_permission == -1) {

fprintf(stderr, "Do not have write permission to destination folder, exiting the code");

return -1;

}

printf("Sorting the files from %s\n", dir);

// read from source and check if files are there, return error if no files

DIR \*dir\_open = opendir(dir);

struct dirent\* in\_file;

int file\_count = 0;

char \*extn;

while ((in\_file = readdir(dir\_open))) {

if (!strcmp (in\_file->d\_name, "."))

continue;

if (!strcmp (in\_file->d\_name, ".."))

continue;

if (!strcmp (in\_file->d\_name, "sorted"))

continue;

extn = strrchr(in\_file->d\_name, '.');

if (strcmp (extn, ".bin"))

continue;

file\_count += 1;

char \*file\_path = (char\*)malloc(strlen(dir) + 20);

sprintf(file\_path, "%s/%s", dir, in\_file->d\_name);

// check if we have access to read the unsorted files

int check\_file\_access = access(file\_path, R\_OK);

if (check\_file\_access == -1) {

fprintf(stderr, "Do not have read permission for the file %s, exiting", file\_path);

return -1;

}

struct stat sb;

if (stat(file\_path, &sb) == -1) {

fprintf(stderr, "Could not get file stat for %s, exiting", file\_path);

return -1;

}

// open file in read only mode

int file\_fd = open(file\_path, O\_RDONLY);

// find out the number of integers in the file

int file\_numbers = 0;

int\* file\_contents = malloc(sb.st\_size);

while (read(file\_fd, file\_contents, sizeof(int)) != 0) {

file\_numbers += 1;

}

if (file\_numbers == 0) {

file\_count -= 1;

continue;

}

// reset the fd to point to 0th index

lseek(file\_fd, 0, 0);

// read the numbers from file and put them into int array

int file\_numbers\_array[file\_numbers];

int counter = 0;

while (counter < file\_numbers) {

read(file\_fd, file\_contents, sizeof(int));

file\_numbers\_array[counter] = \*file\_contents;

counter += 1;

}

close(file\_fd);

// sort the file

sort\_array(file\_numbers\_array, file\_numbers);

// create the new sorted\_\*.bin file name

char sorted\_file\_name[strlen(in\_file->d\_name)-2];

strncpy(sorted\_file\_name, &in\_file->d\_name[2], strlen(in\_file->d\_name));

// create the destination file path

char \*dest\_file\_path = (char\*)malloc(strlen(dest\_folder) + 20);

sprintf(dest\_file\_path, "%s%s", dest\_folder, sorted\_file\_name);

// check write write permission of sorted\_<>.bin files, if dest folder was present already

if (destination\_dir\_exists != -1) {

int check\_write\_file\_access = access(dest\_file\_path, W\_OK);

if (check\_write\_file\_access == -1) {

fprintf(stderr, "Do not have write permission for the file %s, exiting", dest\_file\_path);

return -1;

}}

// open file descriptor for destination file

int write\_fd = open(dest\_file\_path, O\_RDWR|O\_CREAT, 0644);

// write the sorted numbers to file

for (int j=0; j<file\_numbers; j++) {

write(write\_fd, &file\_numbers\_array[j], sizeof(int));

}

close(write\_fd);

printf("Finished sorting files\n");

}

if (file\_count==0) {

fprintf(stderr, "No unsorted files present in the given directory, exiting the code");

return -1;

}

return 0;

}