Assignment 01: filecreator

Work alone or with a partner. If you work with a partner, then indicate their name below AND make sure that your partner also submits this form so that they get credit.

The name, username and photo associated with your Google account will be recorded when you upload files and submit this form. Not **hermann@pdx.edu**? Switch account

* Required

Your Name *

Hermann Yepdjio

Partner's Name (if you worked with a partner).

N/A

filecreator -- create a C language program that creates unsorted binary files

filecreator.c should accept four command line parameters (d = location of directory, f = number of files to create, r = number of integers to insert into each output file and t = number of threads).
it should assume that d exists and give an error if d does not exist or cannot be written to.
then it should create f files in the directory d. each output file should be named "unsorted_<id>.bin" where <id> is a unique integer from 0 to (m-1).
in each output file the program should write r random 32-bit integers. each integer should be chosen randomly by the rand_r() function. note that the output files must be binary. not ascii.
finally, make this program multi-threaded such that each of t threads creates approximately f/t of the files.

You're editing your response. Sharing this URL allows others to also edit your response.

OPEN BLANK FORM

Now create your own filecreator.c according to the specifications described above.

Your source code may only use these C language functions and syscalls: write(), open(), and close(), sprintf(), fprintf() (for error messages only), printf() (for debugging messages only), exit(), atoi(), rand_r() (do not use rand() or any other method for generating random numbers, use rand_r() only), strerror(). It may also use any of the pthread functions.

If you feel you need use something else, then contact me about it.

Run your program with these inputs:

```
./filecreator $(pwd)/unsorted 10 1 1
./filecreator $(pwd)/unsorted 100 2 1
./filecreator $(pwd)/unsorted 1000 10 1

./filecreator $(pwd)/unsorted 10 1 2
./filecreator $(pwd)/unsorted 100 2 2
./filecreator $(pwd)/unsorted 1000 10 2
```

(and try many more inputs to convince yourself that it is working properly)

Upload your finished filecreator.c source code *

filecreator - Her...

Testing

I will take your source code and test it in the following ways:

Can it create 1000 files?
Are the created files all of the correct size?
Does the program correctly detect when the directory does not exist or is not writeable (it should exit with an error in these cases)?
Plus possibly some more tests to be determined.

File Tools

Even though Linux does not technically support typed files, it does provide a utility (called 'file') that allows us to determine whether a file is binary, ascii text or a legal executable.

Also, binary files are not easy to read and manipulate. We can't easily edit them in a text editor, print them to the terminal or search for strings using 'grep'. No worries, the linux utility 'hexdump' can help.

Use the 'file' command with filecreator.c, filecreator (or a.out), and one of your unsorted_xxx.bin files as inputs. Match each file with the file type information provided by the 'file' command. *

	data	executable	text
filecreator.c	○	○	●
filecreator (or a.out)	○	●	○
unsorted binary data file	●	○	○

Next, use the unix utility 'hexdump' to study the contents of one of your unsorted_xxx.bin files. read the hexdump man page or use google to learn how to print 32 bit integers from a binary data file. Copy and paste your precise hexdump command and the last five lines of output from your hexdump command here. *

last five lines of output:

```
1050163100
1741705195
385181049
654666415
1811548362
```

Exact command: `hexdump  -e '1/4 "%08d" "\n"'  unsorted_0.bin`

Submit

Never submit passwords through Google Forms.

You're editing your response. Sharing this URL allows others to also edit your response.

OPEN BLANK FORM