Operating Systems 1 - Laboratory Task 3 105 minutes
Threads and mutexes

Protected Character Table

Write a multithreaded program **pct** which is designed to manage in-memory table of characters. Program is called with 3 arguments n, m, both in range [1, 100] and k in range [3, 100].

```
./pct 4 5 10 # n is 3, m is 5, k is 10
```

The program creates 2-dimensional array filled with random lowercase characters (a-z) with n rows and m columns. Each row is protected by a separate mutex.

The program creates a k worker threads waiting for signals.

The main thread reads commands from the standard input. There are three commands supported: `print`, `replace` and `exit`.

When user enters print, the main thread prints the table contents like in the following output:

```
./pct 4 5 3
> print
cgzbf
abuue
yrvas
```

When user enters `replace n` (i.e. `replace 3`), the main thread sends SIGUSR1 signal to the process itself n times. The main thread blocks this signal. When worker thread receives SIGUSR1 it shall pick a random character from the whole array and replace it with a new random character printing an informative message like:

```
> replace 1
[TID 12345] Received SIGUSR1: Replacing ‘u’ to ‘x’ at index [2,3]
> print
cgzbf
abxue
yrvas
```

When user enters `exit`, the main thread sends SIGINT signal to all worker threads separately. In response worker threads end, main thread joins them and the program exits cleanly.

```
> exit
[TID 12345] Received SIGINT: Exiting
[TID 12346] Received SIGINT: Exiting
[TID 12347] Received SIGINT: Exiting
```

Every access to the row data has to lock corresponding mutex. Each thread shall safely generate a different pseudo-random sequence.

STAGES (TOTAL OF 14 POINTS)

Stage	Points	Requirements
1	4	Program correctly validates all arguments and creates k worker threads. All worker threads print message containing its TID at start and end, sleeping 1 second in-between. The main thread joins all worker thread.
2	4	Program creates a table and fills it with random characters. The main thread waits for commands, print command is working as described. Mutexes are not yet required.
3	4	Replace command is working as described. Random numbers are correctly generated. Access to the table is protected by mutexes.
4	2	Exit command is working as described.

UPLOAD

Please upload your solution to: **/home2/samba/sobotkap/unix/**

You have to upload each stage immediately after finishing it. You upload to a network share via ssh.mini.pw.edu.pl server:

`scp user.etapX.tar.bz2 user@ssh.mini.pw.edu.pl:/home2/samba/sobotkap/unix/`

Please name your stages files according to the schema: `LOGIN.etapN.tar.bz2(.gz)` Solutions not following the naming convention may not be checked.

THE STATEMENT

By decree 27/2020 of University Rector you must add the following statement to the uploads:

```
--------------------------------------------------------------------------
I declare that this piece of work which is the basis for recognition of
achieving learning outcomes in the OPS1 course was completed on my own.
[First and last name] [Student record book number (Student ID number)]
--------------------------------------------------------------------------
```

Please add it as comment at the beginning of each source file you upload. Replace square brackets with your data.