МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИИТ

ОТЧЁТ

по лабораторной работе №7

«Семафоры»

Выполнил:

Студент 2 курса
группы ПО-9
Харитонович Захар Сергеевич
210672

Проверила:

Давидюк Ю. И.

Брест 2022

Вариант 5

Написать две (или более) программы, которые, работая параллельно зацикленно, обмениваются информацией согласно варианту. Передачу и получение информации каждым из процессов сопровождать выводом на экран информации типа "процесс такой-то передал/получил такую-то информацию". Синхронизацию работы процессов реализовать с помощью семафоров. Учтите, что при организации совместного доступа к разделяемому ресурсу (например, файлу) вам понадобится применять мьютексы.

Первый процесс в цикле ожидает ввода символа с потока stdin, после чего пишет в файл случайное число, каждый раз открывая и закрывая за собой файл. Второй процесс забирает из файла числа и выводит на экран соответствующее числу количество любых символов.

main.cpp

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <fcntl.h>
#include <unistd.h>
#include <semaphore.h>

int main() {
    pthread_mutex_t mutex;
    pthread_mutex_init(&mutex, NULL);
    sem_unlink("/sem1");
    sem_unlink("/sem2");
    sem_t *sem1 = sem_open("/sem1", O_CREAT | O_WRONLY, 0777, 0);
    sem_t *sem2 = sem_open("/sem2", O_CREAT | O_RDONLY, 0777, 1);
    while (true) {
        sem_wait(sem2);
        char c;
        read(0, &c, 1);
        char str[8];
        int strSize = sprintf(str, "%d", rand()%100);

        pthread_mutex_lock(&mutex);

        int file = open("file", O_CREAT | O_WRONLY | O_TRUNC, 0777);
        write(file, str, strSize);
        close(file);

        pthread_mutex_unlock(&mutex);
        sem_post(sem1);
    }
    return 0;
}
```

child.cpp

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <fcntl.h>
#include <unistd.h>
#include <semaphore.h>

int main() {
    pthread_mutex_t mutex;
    pthread_mutex_init(&mutex, NULL);
```

```
    sem_t *sem1 = sem_open("/sem1", O_RDONLY);
    sem_t *sem2 = sem_open("/sem2", O_WRONLY);
    while(true) {
        sem_wait(sem1);
        char str[8];
        int num;

        pthread_mutex_lock(&mutex);

        int file = open("file", O_RDONLY, 0777);
        read(file, str, 8);
        close(file);

        pthread_mutex_unlock(&mutex);

        sscanf(str, "%d", &num);
        for(int i = 0; i < num; i++) {
            printf("%c", (char)(rand()%26+65));
        }
        printf("\n\n\n");
        sem_post(sem2);
    }

}
```

```
$ g++ main.cpp -o main -lrt -pthread && g++ child.cpp -o child -
lrt -pthread
$ ./main
a
abc

$ ./child
NWLRBBMQBHCDARZOWKKYHIDDQSCDXRJMOWFRXSJYBLDBEFSARCBYNECDYGGXXPKL
ORELLNMPAPQFWKHOPKM


COQHNWNKUEWHSQMGBBUQCLJJIVSWMDKQTBXIXMVTRRBLJPTNSNFWZQFJMAFADRRW
SOFSBCNUVQHFFBSAQXWPQC


ACEHCHZVFRKMLNOZJKPQPXRJXKITZYXACBHHKICQCOENDTOMFGDWDWFCGPXIQVKU
YTDLCGDEWHTAC


IOHORDTQKVWCSGS


PQOQMSBOAGUWNNYQXNZLGDGWPBTRWBLNSADEUGUUMOQCDRUBETOKYXHOACHWDVMX
XRDRYXLMNDQTUKWAGMLEJUUKWCIBX


UBUMENMEYATDRMYDIAJXLOGHIQFMZHLVIHJ
```

Примечание: \n считывается как отдельный символ.