

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №41

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

Доцент, к. т. н.		О. А. Кононов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

БАРЬЕРЫ

по дисциплине: 'СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ'

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	1742		Д. В. Коробков
	номер группы	подпись, дата	инициалы, фамилия

Санкт-Петербург 2020

1. Постановка задачи

Изучить барьеры на примере программы, приведенной в методическом пособии по лабораторной работе №5 и использующей барьеры.

2. Листинг программы

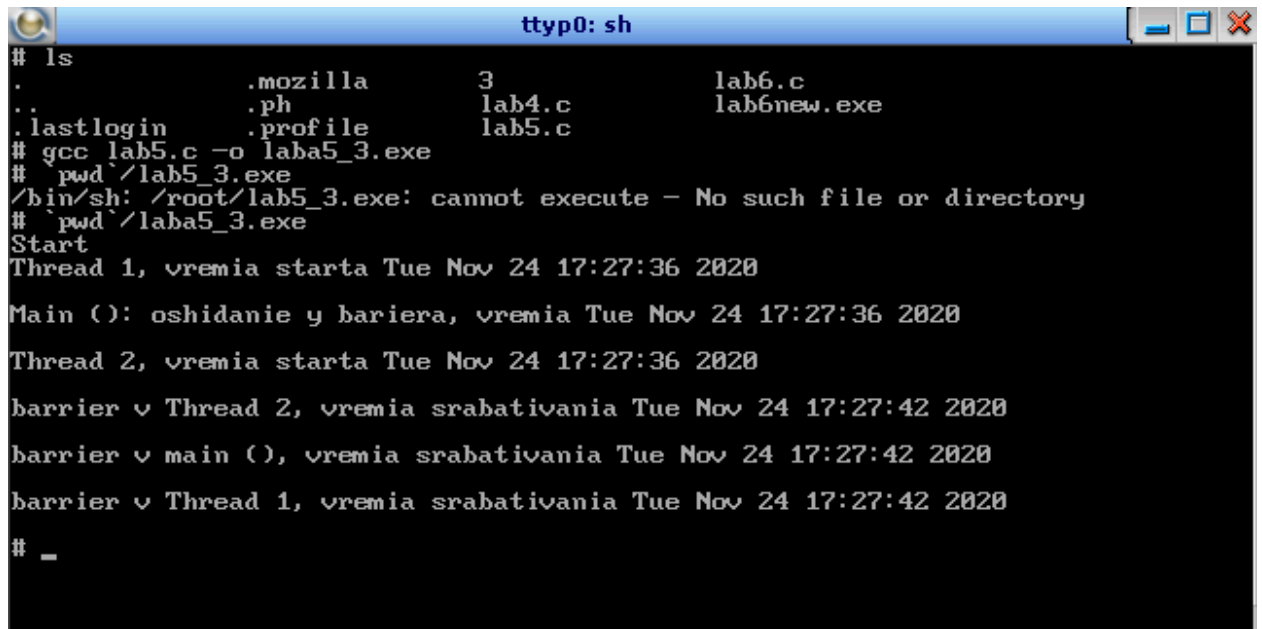
```
#include <sync.h>
#include <sys/neutrino.h>
barrier_t barrier;
void *thread1 (void *not_used)
{
    time_t now;
    char buf [27];
    time(&now);
    printf ("Thread 1, vremia starta %s \n", ctime_r (&now, buf));
    sleep (3);
    barrier_wait (&barrier);
    time(&now);
    printf ("barrier v Thread 1, vremia srabativania %s \n", ctime_r (&now, buf));
}
void *thread2 (void *not_used)
{
    time_t now;
    char buf [27];
    time(&now);
    printf ("Thread 2, vremia starta %s \n", ctime_r (&now, buf));
    sleep (6);
    barrier_wait (&barrier);
    time(&now);
    printf ("barrier v Thread 2, vremia srabativania %s \n", ctime_r (&now, buf));
}

main()
{
    time_t now;
    char buf[27];
    barrier_init(&barrier, NULL, 3);
    printf ("Start \n");
    pthread_create (NULL, NULL, thread1, NULL);
    pthread_create (NULL, NULL, thread2, NULL);
    time(&now);
    printf ("Main (): oshibanie y bariera, vremia %s \n", ctime_r(&now, buf));
    barrier_wait (&barrier);
    time (&now);
    printf ("barrier v main (), vremia srabativania %s \n", ctime_r (&now, buf));
    sleep(5);
}
```

Рисунок 1 – Текст программы

3. Результаты работы программы

В результате выполнения программы удалось понять, как работает метод синхронизации с применением барьера. Для удобного представления процесса работы программы ниже представлена временная диаграмма исполнения потоков.



```
# ls
.          .mozilla      3          lab6.c
..         .ph        lab4.c     lab6new.exe
.lastlogin .profile     lab5.c
# gcc lab5.c -o laba5_3.exe
# pwd`/lab5_3.exe
/bin/sh: /root/lab5_3.exe: cannot execute - No such file or directory
# `pwd`/laba5_3.exe
Start
Thread 1, vremia starta Tue Nov 24 17:27:36 2020
Main (): oshidanie y bariera, vremia Tue Nov 24 17:27:36 2020
Thread 2, vremia starta Tue Nov 24 17:27:36 2020
barrier v Thread 2, vremia srabativania Tue Nov 24 17:27:42 2020
barrier v main (), vremia srabativania Tue Nov 24 17:27:42 2020
barrier v Thread 1, vremia srabativania Tue Nov 24 17:27:42 2020
# _
```

Рисунок 2 – Результат работы программы

4. Временная диаграмма исполнения потоков

Временная диаграмма показывает, работу трех потоков (главного и двух созданных). На ней видно, что созданные потоки имели возможность доработать до условленного места в программе. Каждый поток объявил как о своем старте, так и о своем завершении. Также под временной диаграммой находится пояснение «хода» выполнения программы по потокам. От их старта, до их завершения.

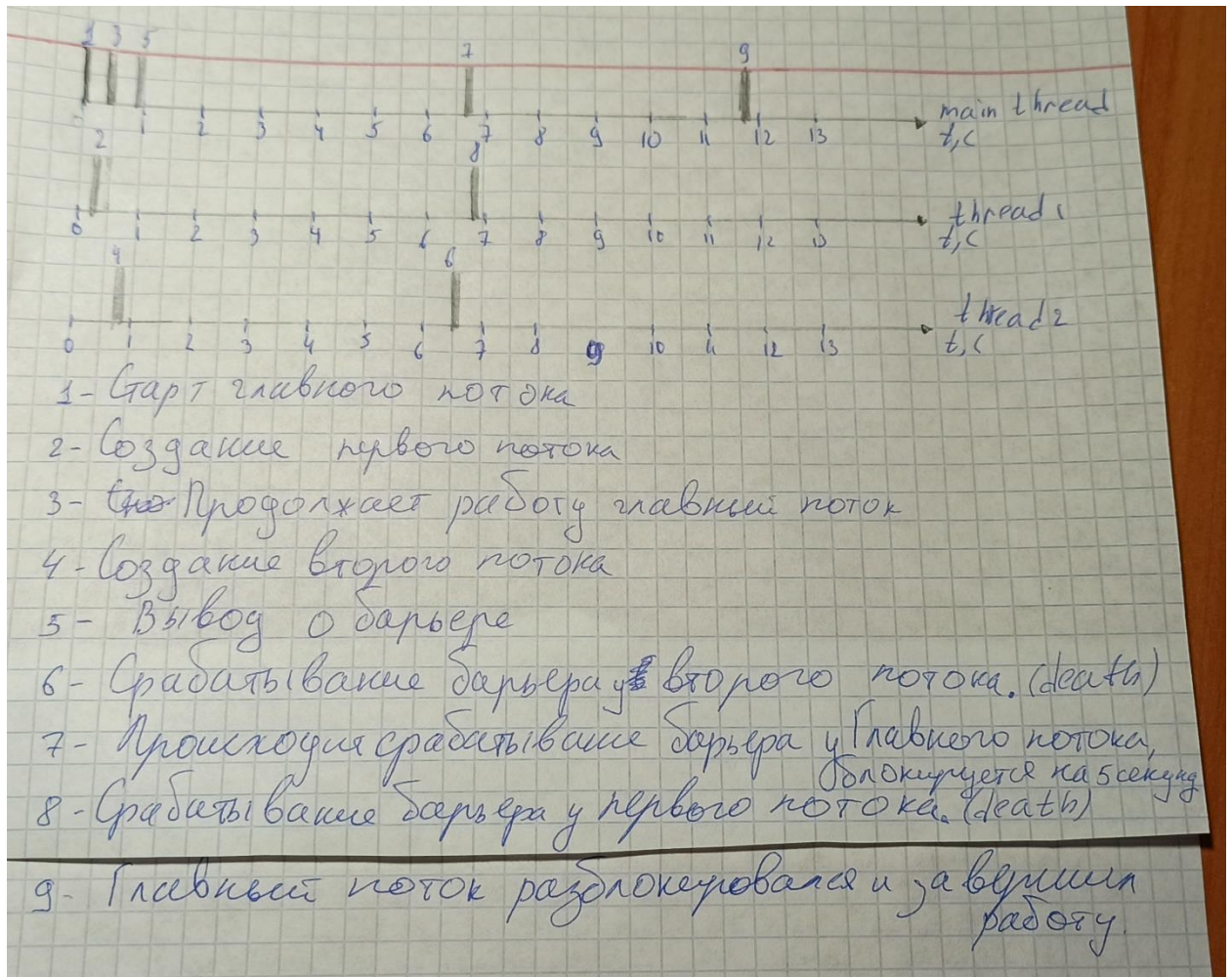


Рисунок 3 – Временная диаграмма исполнения потоков

5. Вывод

В результате выполнения лабораторной работы были изучены барьеры на примере программы из методического пособия. С помощью временной диаграммы было выполнен анализ хода работы программы. Программа работает корректно, что подтверждается снимком результата работы программы.