

## 1. Цель работы

Написать программу, реализующую работу с барьерами

## 2. Ход работы

Основной поток создает объект типа «барьер» при помощи функции `barrier_init0` и инициализирует его значением счетчика, равным числу потоков (включая себя), которые должны «встретиться» у барьера, прежде чем он «прорвется». В данной работе индекс равен 3 — один для потока `main()`, один для потока `thread1()` и один для потока `thread2()`. Далее потоки стартуют. Для осуществления синхронизации основной поток (`main()`) блокирует сам себя на барьере, так как барьер будет разблокирован только после того, как рабочие потоки аналогично присоединятся к нему. После того как нужное число потоков выполнит вызов функции `barrier_wait()`, все эти потоки будут разблокированы «одновременно».

## 3. Листинг программы

```
#include <stdio.h>
#include <time.h>
#include <sync.h>
#include <sys/neutrino.h>

barrier_t barrier;

//int data_ready = 0;
//int inf = 0;
//pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
//pthread_cond_t condvar = PTHREAD_COND_INITIALIZER;

void *thread1 (void *not_used)
{
    time_t now;
    char buf[27];
    time(&now);
```

```

    printf("Potok 1, vremia starta %s \n", ctime_r(&now,buf));
    sleep(3);
    barrier_wait(&barrier);
    time(&now);
    printf("barier v potoke 1, vremia srobativania %s\n", ctime_r(&now,buf));
}

void *thread2 (void * not_used)
{
    time_t now;
    char buf[27];
    time(&now);
    printf("Potok 2, vremia starta %s \n", ctime_r(&now,buf));
    sleep(6);
    barrier_wait(&barrier);
    time(&now);
    printf("barier v potoke 2, vremia srobativania %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(): oshidanie y bariera, vremia %s\n",
ctime_r(&now,buf));
    barrier_wait(&barrier);
    time(&now);
    printf("barier v main(), vremia srobativania %s\n",
ctime_r(&now,buf));
    sleep(5);
}

```

#### 4. Результат работы программы

```
ttyp2: sh
# cd Real_Time_Systems_7sem/LR5
# ls
.      ..      main.c
# gcc main.c
# pwd`/a.out
Start
Potok 1, vremia starta Thu Dec  2 00:03:58 2021
Potok 2, vremia starta Thu Dec  2 00:03:58 2021
Main(): oshidanie y bariera, vremia Thu Dec  2 00:03:58 2021
barier v potoke 2, vremia srbativania Thu Dec  2 00:04:04 2021
barier v potoke 1, vremia srbativania Thu Dec  2 00:04:04 2021
barier v main(), vremia srbativania Thu Dec  2 00:04:04 2021
# _
```

Рисунок 1 - Результаты работы программы

#### 5. Временная диаграмма

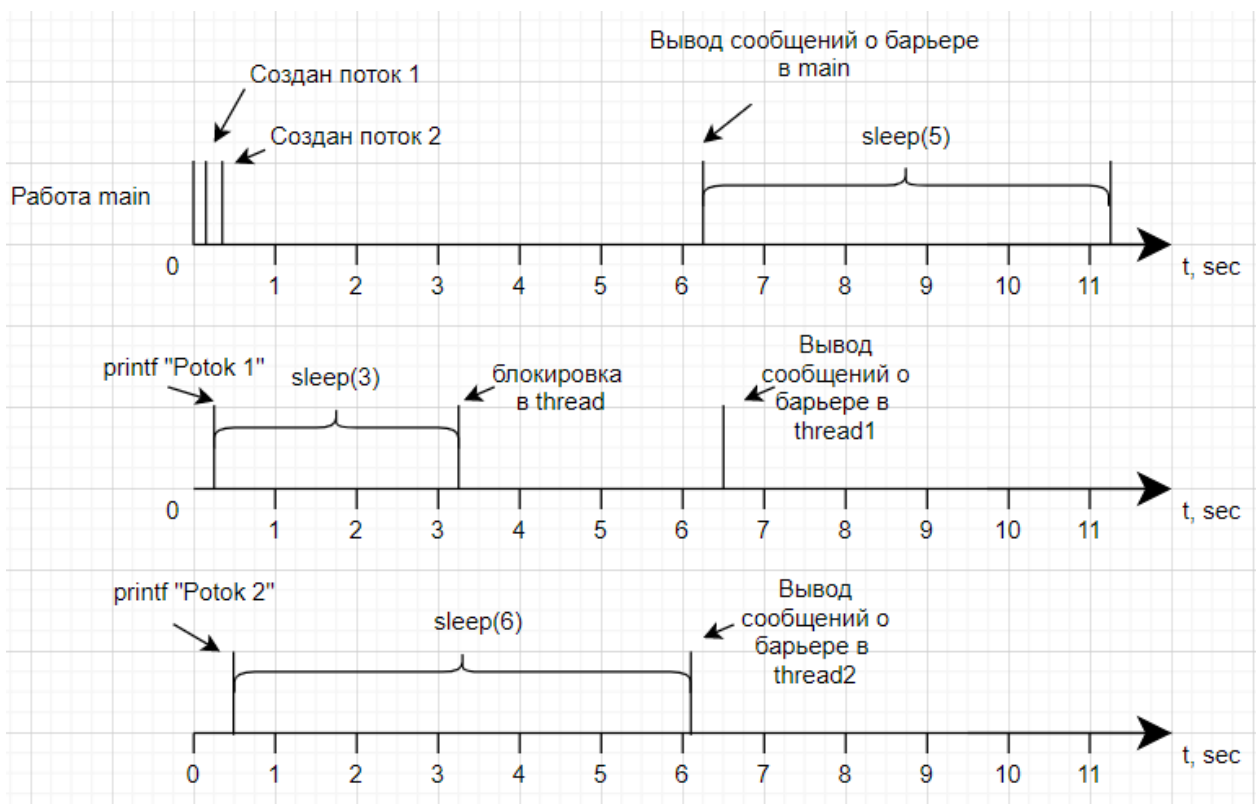


Рисунок 2 - Временная диаграмма

## 6. Вывод

В ходе выполнения лабораторной работы была реализована программа для освоения навыков работы с барьерами при помощи функции `barrier_wait()`.