

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №7
По дисциплине: «Операционные системы и системное программирование»
Тема: «Семафоры»

Подготовил:
Студент 2 курса
Группы ПО-3(2)
Огиевич Е.А.
Проверила:
Давидюк Ю.И.

Брест, 2020

Лабораторная работа №7
Вариант 20

Цель: изучить средства межпроцессного взаимодействия, такие как семафоры.

Задание:

Процессы строят числа Фибоначчи, поочередно вычисляя следующее число, выводя его на экран и передавая его другому процессу, чтобы тот вычислил следующее.

Текст программы 1:

```
#include <unistd.h>
#include <sys/types.h>
#include <semaphore.h>
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#define sem1 "sem1"
#define sem2 "sem2"

/*****/

/*****ПРИНУДИТЕЛЬНО НЕ ЗАВЕРШАТЬ ДО ЗАВЕРШЕНИЯ ПРОГРАММЫ 1*****/

/*****/

int main()
{
    // Создаём 2 семафора, первый сразу включаем для начальной записи
    sem_t *s1 = sem_open(sem1, O_CREAT, 0777, 1);
    sem_t *s2 = sem_open(sem2, O_CREAT, 0777, 0);
    // Открываем файл, в него будут обращаться оба канала и брать данные
    int file_descr = open("file", O_RDWR | O_CREAT);
    if(file_descr < 0)
    {
        printf("Не удалось открыть файл.");
        exit(0);
    }
    // Т.к. числа Фибоначчи, нам нужно 2 стартовых значения, первое будет сразу в
    // файле
    // Чтобы файл при каждом запуске был готов к работе сразу, всегда пишем туда
    // единицу
    int one = 1;
    int write_status;

    write_status = write(file_descr, &one, sizeof(int));
    if(write_status < 0)
    {
```

```

printf("Не удалось сделать запись в файл.");
exit(0);
}
close(file_descr);

//int n;
//printf("До какого числа необходимо дойти: "); // имеется в виду порядковый номер в
последовательности
//scanf("%d", &n);

int read_status;
int current; int result; int prev = 0;
for (int i = 0; i < 5; i++) // 3 числа у нас и так есть
{
sem_wait(s1); printf("Остановлен семофор 1\n"); // первый семофор уже отработал,
приостанавливаем
file_descr = open("file", O_RDWR); // открываем файл
if(file_descr < 0)
{
printf("Не удалось открыть файл.");
exit(0);
}

// Из файла извлекаем текущее число
read_status = read(file_descr, &current, sizeof(int));
if(read_status < 0)
{
printf("Не удалось прочитать файл.");
exit(0);
}
close(file_descr);
printf("Программа 1 приняла число: %d\n", current);
// По определению чисел Фибоначчи высчитываем следующее(prev на 1 проход
равен 1)
result = current + prev;
prev = result; // и когда вычислили след.число сразу запоминаем его как предыдущее
для след.итерации

// Открываем файл и записываем в него текущий элемент
file_descr = open("file", O_RDWR);
if(file_descr < 0)
{
printf("Не удалось открыть файл.");
exit(0);
}

// тут запись
write_status = write(file_descr, &result, sizeof(int));
if(write_status < 0)
{
printf("Не удалось записать файл.");

```

```

}

printf("Программа 1 отправила число: %d\n", result);
close(file_descr);
sleep(1);
// запускаем второй семофор
sem_post(s2); printf("Запущен семофор 2\n");
}
sem_unlink(sem1);
sem_unlink(sem2);
return 0;
}

```

Текст программы 2:

```

#include <unistd.h>
#include <sys/types.h>
#include <semaphore.h>
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#define sem1 "sem1"
#define sem2 "sem2"

int main()
{
sem_t *s1 = sem_open(sem1, O_CREAT, 0777, 0);
sem_t *s2 = sem_open(sem2, O_CREAT, 0777, 0);
int read_status; int write_status; int file_descr;
int current; int prev = 1; int result;

while(1)
{
sem_wait(s2); printf("Остановлен семофор 2\n");
file_descr = open("file", O_RDWR);
if(file_descr < 0)
{
printf("Не удалось открыть файл.");
exit(0);
}

/*read_status = */read(file_descr, &current, sizeof(int));
/*if(read_status < 0)
{
printf("Не удалось прочитать файл.");
exit(0);
}

```

```

}*/
close(file_descr);

printf("Программа 2 приняла число: %d\n", current);

result = current + prev;
prev = result;

file_descr = open("file", O_RDWR);
if(file_descr < 0)
{
printf("Не удалось открыть файл.");
exit(0);
}

/*write_status = */write(file_descr, &result, sizeof(int));
/*if(write_status < 0)
{
printf("Не удалось записать файл.");
}*/

printf("Программа 2 отправила число: %d\n", result);
close(file_descr);
sleep(1);
sem_post(s1); printf("Запущен семафор 1\n");

}
sem_unlink(sem1);
sem_unlink(sem2);
return 0;
}

```

Результат выполнения:

```

natefoust@natefoust-X550LC:~/05/lab7$ gcc lab7.c -o 1.out -lnatefoust@natefoust-X550LC:~/05/lab7$ gcc lab72.c -lrt -lpthread
natefoust@natefoust-X550LC:~/05/lab7$ ./1.outnatefoust@natefoust-X550LC:~/05/lab7$ ./a.out
Остановлен семафор 1Остановлен семафор 2
Программа 1 приняла число: 1Программа 2 приняла число: 1
Программа 1 отправила число: 1Программа 2 отправила число: 2
Запущен семафор 2Запущен семафор 1
Остановлен семафор 1Остановлен семафор 2
Программа 1 приняла число: 2Программа 2 приняла число: 3
Программа 1 отправила число: 3Программа 2 отправила число: 5
Запущен семафор 2Запущен семафор 1
Остановлен семафор 1Остановлен семафор 2
Программа 1 приняла число: 5Программа 2 приняла число: 8
Программа 1 отправила число: 8Программа 2 отправила число: 13
Запущен семафор 2Запущен семафор 1
Остановлен семафор 1Остановлен семафор 2
Программа 1 приняла число: 13Программа 2 приняла число: 21
Программа 1 отправила число: 21Программа 2 отправила число: 34
Запущен семафор 2Запущен семафор 1
Остановлен семафор 1Остановлен семафор 2
Программа 1 приняла число: 34Программа 2 приняла число: 55
Программа 1 отправила число: 55Программа 2 отправила число: 89
Запущен семафор 2Запущен семафор 1
natefoust@natefoust-X550LC:~/05/lab7$ █

```

Вывод: изучил работу семафоров.