

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

Лабораторная работа №7
за I семестр
по дисциплине: "Операционные системы и системное программирование"
Тема: "Семафоры"

ЛР.190333.ПО4.07 81 00

Листов 6

Выполнил:
студент 2-ого курса
IV-ого семестра
факультета ЭИС
группы ПО-4(1)
Галанин П. И.

Проверил:
ст. преподаватель
Давидюк Ю. И.

Лабораторная работа №7

Тема: «Семафоры»

Цель работы:

Ход работы:

Вариант V

Условие: Певый 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 соответствующему числу количество любых СИМВОЛОВ.

Решение:

Листинг: Makefile

```
1 c1:                                     #$ make c1 # компиляция программы 1
2     gcc one.c -lrt -pthread -o one.out
3 r1:                                     #$ make r1 # запуск программы 1
4     ./one.out
5 c2:                                     #$ make c2 # компиляция программы 2
6     gcc two.c -lrt -pthread -o two.out
7 r2:                                     #$ make r2 # запуск программы 2
8     ./two.out
```

Листинг: one.c

```
1 #include <unistd.h>
2 #include <stdio.h>      //printf()
3 #include <stdlib.h>     //rand()
4 #include <sys/types.h>
5 #include <sys/ipc.h>
6 #include <sys/shm.h>
7 #include <fcntl.h>
8 #include <semaphore.h>
9 #include <pthread.h>
10 #include <time.h>      //time(), localtime
11
12 int get_random(int a, int b);
13 void print_real_time();
14
15 int main()
16 {
17     sem_t* get = sem_open(
18         "sem_one",
19         O_CREAT,
20         0777,
```

| | | | | | | | | | | | |
|-----------|---------|----------|-------|------|------------------------------------|--|--|--|-------|------|--------|
| | | | | | ЛР.190333.ПО4.07 81 00 | | | | | | |
| Изм | Лист | № докум. | Подп. | Дата | | | | | | | |
| Разраб. | Галанин | | | | Лабораторная работа №7 Семафоры | | | | Лит. | Лист | Листов |
| Пров. | Давидюк | | | | | | | | Л | 2 | 6 |
| Н. контр. | | | | | | | | | БрГТУ | | |
| Утв. | | | | | | | | | | | |

```

21         0 //unsigned int value
22     );
23     sem_t* put = sem_open(
24         "sem_two", //const char* name
25         O_CREAT, //int oflag
26         0777, //mode_t mode
27         0 //unsigned int value
28     );
29     pthread_mutex_t mutex;
30     pthread_mutex_init(&mutex, NULL);
31
32     sem_post(put);
33     (void)umask(0);
34
35     while (1)
36     {
37         sem_wait(put);
38         print_real_time(); //печать реального времени
39         printf("Enter the symbol: ");
40         pthread_mutex_lock(&mutex);
41         int fd = open( //открываем файл
42             "file.bin", //*.bin - бинарный файл
43             O_CREAT | O_WRONLY,
44             0777 // - rwx rwx rwx (0 111 111 111)
45         );
46
47         //Если вводить по одному символу,
48         //то сообщение может отправляться несколько раз,
49         //при вводе нескольких символов (а не одного).
50         //Эту проблему решает строка.
51         char str[1024]; //инициализировали строку
52         scanf("%s", str); //вводим строку
53         char character = str[0]; //взяли первый символ со строки
54
55         int rand_number = get_random(1, 35); //рандомное число [1; 35]
56         print_real_time(); //печать реального времени
57         printf( //печать сообщения об отправке
58             "Process %d send random_number: %d\n\n",
59             getpid(), //PID - номер процесса
60             rand_number //рандомное число
61         );
62
63         char msg[1024]; //инициализация строки для сообщения
64         sprintf(msg, "%d", rand_number); //в сообщение записали рандомное число
65         write(fd, msg, sizeof(msg)); //записали сообщение в файл
66
67         close(fd); //закрываем файл
68         pthread_mutex_unlock(&mutex);
69         sem_post(get);
70     }
71
72     pthread_mutex_destroy(&mutex);
73     sem_close(put);
74     sem_close(get);
75
76     return 0;
77 }
78
79 int get_random(int x1, int x2)
80 {

```

```

81     int a = x1 < x2 ? x1 : x2;           //a = min(a, b)
82     int b = x1 > x2 ? x1 : x2;           //b = max(a, b)
83     return rand() % (b - a + 1) + a;      //random [a; b]
84 }
85
86 void print_real_time()
87 {
88     time_t now = time(0);
89     struct tm *ltm = localtime(&now);
90     printf(
91         "%02d:%02d:%02d : ",
92         1 + ltm->tm_hour,           // часы
93         1 + ltm->tm_min,           // минуты
94         1 + ltm->tm_sec            // секунды
95     );
96 }

```

Листинг: two.c

```

1  #include <unistd.h>
2  #include <stdio.h>           //printf()
3  #include <stdlib.h>          //rand()
4  #include <sys/types.h>
5  #include <sys/ipc.h>
6  #include <sys/shm.h>
7  #include <fcntl.h>
8  #include <semaphore.h>
9  #include <pthread.h>
10 #include <time.h>            //time(), localtime()
11
12 int get_random(int x1, int x2);
13 void print_real_time();
14
15 int main() {
16     char symbols[] = "AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtVvUuWwXxYyZz"; //символы
17     sem_t* get = sem_open(
18         "sem_one",           //const char* name
19         O_CREAT,             //int oflag
20         0777,               //mode_t mode
21         0                    //unsigned int value
22     );
23     sem_t* put = sem_open(
24         "sem_two",           //const char* name
25         O_CREAT,             //int oflag
26         0777,               //mode_t mode
27         0                    //unsigned int value
28     );
29     pthread_mutex_t mutex;
30     pthread_mutex_init(&mutex, NULL);
31
32     (void)umask(0);
33     while (1)                //бесконечный цикл
34     {
35         sem_wait(get);
36         pthread_mutex_lock(&mutex);
37         int fd = open(        //открываем файл
38             "file.bin",       //*.bin - бинарный файл
39             O_RDWR,
40             0777              //- rwx rwx rwx (0 111 111 111)
41         );

```

```

42
43     char str[1024]; //строку, которую возьмем из файла
44     read(fd, &str, sizeof(str)); //записали данные в строку
45
46     print_real_time();
47     printf(
48         "Process %d get message: %s\n", //форматная строка
49         getpid(), //PID - номер процесса
50         str //строка
51     );
52
53     print_real_time();
54     int length = atoi(str); //длина строки (длины цифры)
55     for(int i = 0; i < length; i++) //выводим (length - 1) случайную букву
56     {
57         int random_number = get_random(0, sizeof(symbols)); //случайное число из массива букв
58         char random_character = symbols[random_number]; //получаем символ
59         printf("%c", random_character); //печатаем символ
60     }
61     printf("\n\n");
62
63     close(fd); //закрываем файл
64     pthread_mutex_unlock(&mutex);
65     sem_post(put);
66 }
67
68 pthread_mutex_destroy(&mutex);
69 sem_close(put);
70 sem_close(get);
71
72 return 0;
73 }
74
75 int get_random(int x1, int x2)
76 {
77     int a = x1 < x2 ? x1 : x2; //a = min(a, b)
78     int b = x1 > x2 ? x1 : x2; //b = max(a, b)
79     return rand() % (b - a + 1) + a; //random [a; b]
80 }
81
82 void print_real_time()
83 {
84     time_t now = time(0);
85     struct tm *ltm = localtime(&now);
86     printf(
87         "%02d:%02d:%02d : ",
88         1 + ltm->tm_hour, // часы
89         1 + ltm->tm_min, // минуты
90         1 + ltm->tm_sec // секунды
91     );
92 }

```

Листинг: Tetminal 1

1 make c1

Листинг: Tetminal 2

1 make c2

Листинг: Tetminal 1

3 make r1

Листинг: Tetminal 2

3 make r2

Листинг: Tetminal 1

5 ./one.out
6 08:19:18 : Enter the symbol: q
7 08:19:18 : Process 8497 send random_number: 9
8
9 08:19:23 : Enter the symbol: 2
10 08:19:25 : Process 8497 send random_number: 12
11
12 08:19:25 : Enter the symbol: 3
13 08:19:29 : Process 8497 send random_number: 3
14
15 08:19:29 : Enter the symbol: 4
16 08:19:29 : Process 8497 send random_number: 6
17
18 08:19:29 : Enter the symbol: 5
19 08:19:30 : Process 8497 send random_number: 9
20
21 08:19:30 : Enter the symbol: q
22 08:19:32 : Process 8497 send random_number: 11
23
24 08:19:32 : Enter the symbol: w
25 08:19:32 : Process 8497 send random_number: 32
26
27 08:19:32 : Enter the symbol: e
28 08:19:33 : Process 8497 send random_number: 3
29
30 08:19:33 : Enter the symbol: r
31 08:19:33 : Process 8497 send random_number: 30
32
33 08:19:33 : Enter the symbol: t
34 08:19:33 : Process 8497 send random_number: 32
35
36 08:19:33 : Enter the symbol: y
37 08:19:35 : Process 8497 send random_number: 3
38
39 08:19:35 : Enter the symbol:
40

Листинг: Tetminal 2

5 ./two.out
6 08:19:23 : Process 8514 get message: 9
7 08:19:23 : jIwmvmXGq
8
9 08:19:25 : Process 8514 get message: 12
10 08:19:25 : sZpQdcCDAFr
11
12 08:19:29 : Process 8514 get message: 3
13 08:19:29 : Nzq
14
15 08:19:29 : Process 8514 get message: 6
16 08:19:29 : WENyss
17
18 08:19:30 : Process 8514 get message: 9
19 08:19:30 : cJAGcbasp
20
21 08:19:32 : Process 8514 get message: 11
22 08:19:32 : IzneoZpZBxY
23
24 08:19:32 : Process 8514 get message: 32
25 08:19:32 : oOsNEZZLyqnHqBrtyoJeJnThKZqHdM
26
27 08:19:33 : Process 8514 get message: 3
28 08:19:33 : UWZ
29
30 08:19:33 : Process 8514 get message: 30
31 08:19:33 : GqxEaUUWJsXISYfnOKXiklTqJBuBE
32
33 08:19:33 : Process 8514 get message: 32
34 08:19:33 : ahvFlChnyqbfegtkDcCocTFbBkfA
35
36 08:19:35 : Process 8514 get message: 3
37 08:19:35 : Sf
38
39
40