



НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

по лабораторной работе № 3

Дисциплина: Операционные системы

Преподаватель	<hr/>	Н.Ю. Рязанова
	(Подпись, дата)	(И.О. Фамилия)

Москва, 2020

Задание 1

«Производство-потребление»

Листинг программы:

main_header.h

```
#ifndef MAIN_H
#define MAIN_H

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>
#include <sys/stat.h>
#include <sys/shm.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define FULL_SEMN 0
#define EMPT_SEMN 1
#define BIN_SEMN 2

#define QUEUE_SIZE 5
#define BUF_SIZE 2 + QUEUE_SIZE
#define PROC_N 3

#endif // MAIN_H
```

main.c

```
#include "main_header.h"

int rand_dt()
{
    return 1 + rand() % 3;
}

void prod_func(size_t b_size, int isem_descry, char* buf, int my_n)
{
    srand(time(NULL) + my_n*100);
    struct sembuf pre_sem[2] = { {EMPT_SEMN, -1, SEM_UNDO},
                                  {BIN_SEMN, -1, SEM_UNDO} };
    struct sembuf post_sem[2] = { {FULL_SEMN, 1, SEM_UNDO},
                                   {BIN_SEMN, 1, SEM_UNDO} };
    char cur_letter = 'A';
    while (1)
    {
        if (semop(isem_descry, pre_sem, 2) == -1)
        {
            perror("semop");
        }
    }
}
```

```

        exit(1);
    }

    buf[buf[1]] = cur_letter;
    if (++buf[1] >= b_size)
        buf[1] = 2;
    printf("Producer N°%d wrote:\t %c\n", my_n+1, cur_letter);

    if (semop(isem_descry, post_sem, 2) == -1)
    {
        perror("semop");
        exit(1);
    }

    if (++cur_letter > 'Z')
        cur_letter = 'A';
    sleep(rand_dt());
}

}

void cons_func(size_t b_size, int isem_descry, char* buf, int my_n)
{
    srand(time(NULL) + my_n*10);
    struct sembuf pre_sem[2] = { {FULL_SEMN, -1, SEM_UNDO},
                                  {BIN_SEMN, -1, SEM_UNDO} };
    struct sembuf post_sem[2] = { {EMPT_SEMN, 1, SEM_UNDO},
                                   {BIN_SEMN, 1, SEM_UNDO} };
    char cur_letter;
    while (1)
    {
        if (semop(isem_descry, pre_sem, 2) == -1)
        {
            perror("semop");
            exit(1);
        }

        cur_letter = buf[buf[0]];
        printf("Consumer N°%d read:\t\t %c\n", my_n+1, cur_letter);
        if (++buf[0] >= b_size)
            buf[0] = 2;

        if (semop(isem_descry, post_sem, 2) == -1)
        {
            perror("semop");
            exit(1);
        }

        sleep(rand_dt());
    }
}

int main(void)
{
    int perms = S_IRWXU | S_IRWXO | S_IRWXG;

```

```

int isem_descry = semget(IPC_PRIVATE, 3, IPC_CREAT | perms );
if (isem_descry == -1)
{
    perror("semget");
    return 1;
}
int full_ctl = semctl(isem_descry, FULL_SEMN, SETVAL, 0);
int empt_ctl = semctl(isem_descry, EMPT_SEMN, SETVAL, QUEUE_SIZE);
int bin_ctl = semctl(isem_descry, BIN_SEMN, SETVAL, 1);
if (full_ctl == -1 || empt_ctl == -1 || bin_ctl == -1)
{
    perror("semctl");
    return 1;
}

int mem_id = shmget(IPC_PRIVATE, (BUF_SIZE)*sizeof(char), IPC_CREAT | perms);
if (mem_id == -1)
{
    perror("shmget");
    return 1;
}
char* addr = shmat(mem_id, 0, 0);
if (addr == (char*)(-1))
{
    perror("shmat");
    return 1;
}

addr[0] = (char)2;
addr[1] = (char)2;

printf("> Start of simulation\n");
for (size_t i=0; i<PROC_N; i++)
{
    pid_t prod_pid = fork();
    switch (prod_pid)
    {
        case -1:
            perror("fork");
            return 1;
        case 0:
            prod_func(BUF_SIZE, isem_descry, addr, i);
            return 0;
        default:
            printf("> Producer created\n");
            break;
    }
}

pid_t cons_pid = fork();
switch (cons_pid)
{
    case -1:
        perror("fork");
        return 1;
    case 0:
        cons_func(BUF_SIZE, isem_descry, addr, i);

```

```
        return 0;
    default:
        printf("> Consumer created\n");
        break;
    }
}

int status, pid;
for (size_t i=0; i<PROC_N*2; i++)
{
    pid = wait(&status);
    if (pid == -1)
    {
        perror("wait");
        return 1;
    }
}

if (semctl(isem_descry, 0, IPC_RMID, 0) == -1)
{
    perror("semctl");
    return 1;
}
{
    perror("shmctl");
    return 1;
}
if (shmdt(addr) == -1)
{
    perror("shmdt");
    return 1;
}

return 0;
}
```

Пример работы:

```
vsevolod@vsevolod-HP-Pavilion14:~/work/OS_bmstu/lab3_1$ gcc -o main.o main.c
vsevolod@vsevolod-HP-Pavilion14:~/work/OS_bmstu/lab3_1$ ./main.o
> Start of simulation
> Producer created
> Consumer created
> Producer created
Producer №1 wrote:      A
Consumer №1 read:      A
> Consumer created
Producer №2 wrote:      A
> Producer created
> Consumer created
Producer №3 wrote:      A
Consumer №2 read:      A
Consumer №3 read:      A
Producer №1 wrote:      B
Consumer №1 read:      B
Producer №3 wrote:      B
Consumer №3 read:      B
Producer №1 wrote:      C
Producer №2 wrote:      B
Consumer №1 read:      C
Consumer №2 read:      B
Producer №3 wrote:      C
Producer №2 wrote:      C
Consumer №3 read:      C
Producer №3 wrote:      D
Producer №1 wrote:      D
Consumer №1 read:      C
Consumer №2 read:      D
Producer №3 wrote:      E
Producer №2 wrote:      D
Producer №1 wrote:      E
Consumer №1 read:      D
Consumer №3 read:      E
Producer №3 wrote:      F
Producer №2 wrote:      E
Consumer №2 read:      D
Producer №1 wrote:      F
Consumer №1 read:      E
Consumer №2 read:      F
Consumer №3 read:      E
```

Задание 2

«Читатели-писатели»

Листинг программы:

main_header.h

main.c

Пример работы: