Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №3 по курсу

«Операционные системы»

Группа: М8О-215Б-23

Студент: Кармишен Е.С.

Преподаватель: Миронов Е.С. (ПМИ)

Оценка: _____

Дата: 14.02.24

Москва, 2024

Постановка задачи

Вариант 12.

Составить и отладить программу на языке Си, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. В результате работы программа (основной процесс) должен создать для решение задачи один или несколько дочерних процессов. Взаимодействие между процессами осуществляется через системные сигналы/события и/или через отображаемые файлы (memory-mapped files).

Child1 переводит строки в верхний регистр. Child2 убирает все задвоенные пробелы.

Общий метод и алгоритм решения

Использованные системные вызовы:

Родительский процесс (parent.c):

- Системные вызовы:
  - fork(): Создает дочерние процессы.
  - waitpid(): Ожидает завершения дочерних процессов.
  - mmap(): Отображает файл shared memory в память.
  - munmap(): Освобождает отображенную память.
  - open(): Открывает файл shared memory.
  - close(): Закрывает файловый дескриптор.
  - ftruncate(): Устанавливает размер файла shared memory.
  - sem_init(): Инициализирует семафоры.
  - sem_destroy(): Уничтожает семафоры.
  - sem_post(): Увеличивает значение семафора.
  - sem_wait(): Уменьшает значение семафора.

Дочерний процесс 1 (child1.c):

- Системные вызовы:
  - mmap(): Отображает файл shared memory в память.
  - munmap(): Освобождает отображенную память.
    open(): Открывает файл shared memory.
  - close(): Закрывает файловый дескриптор.
  - sem_wait(): Ожидает, пока данные не будут готовы.
  - sem_post(): Сигнализирует о завершении обработки.
  - sem_trywait(): Проверяет, завершена ли работа.

Дочерний процесс 2 (child2.c):

- Системные вызовы:
  - mmap(): Отображает файл shared memory в память.
  - munmap(): Освобождает отображенную память.
  - open(): Открывает файл shared memory.
  - close(): Закрывает файловый дескриптор.
  - sem_wait(): Ожидает, пока данные не будут готовы.
  - sem_post(): Сигнализирует о завершении обработки.
  - sem_trywait(): Проверяет, завершена ли работа.

Работа программы: аналогична первой лабораторной, вместо pipe-ов используется MMF

Код программы

common.h

```c
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <ctype.h>

#include <sys/mman.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <semaphore.h>


#define MAX_LINE 1000


typedef struct {

char data[MAX_LINE];

sem_t sem_data_ready; // Семафор: данные готовы для обработки

sem_t sem_data_processed_by_1; // Семафор: данные обработаны

sem_t sem_data_processed_by_2; // Семафор: данные обработаны

} SharedData;
```

parent.c

```c
#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/wait.h>

#include <sys/mman.h>

#include <fcntl.h>

#include <sys/stat.h>
```

```c
#include <semaphore.h>

#include "common.h"



#define SHARED_FILE "shared_memory.bin"



int main() {

int fd;

SharedData *shared;

pid_t child1, child2;



// Создаем файл для shared memory

fd = open(SHARED_FILE, O_RDWR | O_CREAT | O_TRUNC, 0666);

if (fd == -1) {

perror("Ошибка открытия файла");

exit(1);

}



// Устанавливаем размер файла

if (ftruncate(fd, sizeof(SharedData)) == -1) {

perror("Ошибка ftruncate");

close(fd);

exit(1);

}



// Отображаем файл в память

shared = (SharedData *)mmap(NULL, sizeof(SharedData), PROT_READ | PROT_WRITE, MAP_SHARED,
fd, 0);

if (shared == MAP_FAILED) {

perror("Ошибка mmap");

close(fd);

exit(1);
```

```
}


close(fd); // Файловый дескриптор больше не нужен


// Инициализация семафоров
sem_init(&shared->sem_data_ready, 1, 0); // Изначально данные не готовы
sem_init(&shared->sem_data_processed_by_1, 1, 0); // Изначально данные не обработаны
sem_init(&shared->sem_data_processed_by_2, 1, 0); // Изначально данные не обработаны
// Создаем первый дочерний процесс
if ((child1 = fork()) == -1) {
perror("Ошибка создания первого дочернего процесса");
munmap(shared, sizeof(SharedData));
exit(1);
}


if (child1 == 0) {
// Код для первого дочернего процесса (child1)
execl("./child1", "child1", SHARED_FILE, NULL);
perror("Ошибка execl для child1");
exit(1);
}


// Создаем второй дочерний процесс
if ((child2 = fork()) == -1) {
perror("Ошибка создания второго дочернего процесса");
munmap(shared, sizeof(SharedData));
exit(1);
}


if (child2 == 0) {
// Код для второго дочернего процесса (child2)
```

```
execl("./child2", "child2", SHARED_FILE, NULL);
perror("Ошибка execl для child2");
exit(1);
}

// Родительский процесс
printf("Введите строки (Ctrl+D для завершения):\n");

while (fgets(shared->data, MAX_LINE, stdin) != NULL) {
// Проверяем что не отправляем пустую строку
if (shared->data[0] != '\0') {
// Сигнализируем, что данные готовы для обработки
sem_post(&shared->sem_data_ready);
// Ждем, пока данные будут обработаны
sem_wait(&shared->sem_data_processed_by_2);
}

// Выводим результат
printf("Результат: %s\n", shared->data);
}
// Сигнализируем дочерним процессам о завершении (отправляем пустую строку)
shared->data[0] = '\0';
sem_post(&shared->sem_data_ready);
sem_wait(&shared->sem_data_processed_by_2);

// Ожидание завершения дочерних процессов
int status;
waitpid(child1, &status, 0); // Ждем завершения child1
if (WIFEXITED(status) && WEXITSTATUS(status) != 0) {
printf("Дочерний процесс 1 завершился с ошибкой: %i", WEXITSTATUS(status));
}
```

```
    waitpid(child2, &status, 0); // Ждем завершения child2
    if (WIFEXITED(status) && WEXITSTATUS(status) != 0) {
    printf("Дочерний процесс 2 завершился с ошибкой: %i", WEXITSTATUS(status));
    }

    // Уничтожение семафоров
    sem_destroy(&shared->sem_data_ready);
    sem_destroy(&shared->sem_data_processed_by_1);
    sem_destroy(&shared->sem_data_processed_by_2);

    // Освобождаем shared memory
    munmap(shared, sizeof(SharedData));

    printf("\nВсе процессы завершены.\n");

    return 0;
    }
```

child1.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <ctype.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <semaphore.h>
#include "common.h"
```

```
int main(int argc, char *argv[]) {
if (argc < 2) {
fprintf(stderr, "Не указан файл shared memory\n");
exit(1);
}

const char *shared_file = argv[1];
int fd;
SharedData *shared;

// Открываем файл shared memory
fd = open(shared_file, O_RDWR);
if (fd == -1) {
perror("Ошибка открытия файла");
exit(1);
}

// Отображаем файл в память
shared = (SharedData *)mmap(NULL, sizeof(SharedData), PROT_READ | PROT_WRITE, MAP_SHARED,
fd, 0);
if (shared == MAP_FAILED) {
perror("Ошибка mmap");
close(fd);
exit(1);
}

close(fd); // Файловый дескриптор больше не нужен

while (1) {
// Ждем, пока данные будут готовы для обработки
```

```
        sem_wait(&shared->sem_data_ready);


        // Проверяем, завершена ли работа
        if (shared->data[0] == '\0') {
        sem_post(&shared->sem_data_processed_by_1);
        break;
        }


        // Преобразуем строку в верхний регистр
        for (int i = 0; shared->data[i]; i++) {
        shared->data[i] = toupper(shared->data[i]);
        }


        // Сигнализируем, что данные обработаны
        sem_post(&shared->sem_data_processed_by_1);
        }


        // Освобождаем shared memory
        munmap(shared, sizeof(SharedData));


        return 0;
        }
```

child2.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <semaphore.h>
```

```c
#include "common.h"



void remove_extra_spaces(char *str) {

int i = 0, j = 0;

int space_found = 0;



while (str[i]) {

if (str[i] == ' ') {

if (!space_found) {

str[j++] = str[i];

space_found = 1;

}

} else {

str[j++] = str[i];

space_found = 0;

}

i++;

}

str[j] = '\0';

}



int main(int argc, char *argv[]) {

if (argc < 2) {

fprintf(stderr, "Не указан файл shared memory\n");

exit(1);

}



const char *shared_file = argv[1];

int fd;

SharedData *shared;
```

```
// Открываем файл shared memory
fd = open(shared_file, O_RDWR);
if (fd == -1) {
perror("Ошибка открытия файла");
exit(1);
}

// Отображаем файл в память
shared = (SharedData *)mmap(NULL, sizeof(SharedData), PROT_READ | PROT_WRITE, MAP_SHARED, fd,
0);
if (shared == MAP_FAILED) {
perror("Ошибка mmap");
close(fd);
exit(1);
}

close(fd); // Файловый дескриптор больше не нужен

while (1) {
// Ждем, пока данные будут готовы для обработки
sem_wait(&shared->sem_data_processed_by_1);

// Проверяем, завершена ли работа
if (shared->data[0] == '\0') {
sem_post(&shared->sem_data_processed_by_2);
break;
}

// Удаляем лишние пробелы
remove_extra_spaces(shared->data);
```

```
// Сигнализируем, что данные обработаны
sem_post(&shared->sem_data_processed_by_2);
}

// Освобождаем shared memory
munmap(shared, sizeof(SharedData));

return 0;
}
```

Протокол работы программы

Тестирование:

apple@MacBook-Pro-apple src % ./Parent
Введите строки (Ctrl+D для завершения):
apple space sweet dogs
Результат: APPLE SPACE SWEET DOGS
hot Jack cat
Результат: HOT JACK CAT
^D
Все процессы завершены.

Strace:

```
execve("./parent", ["./parent"], 0x7ffd98dc6968 /* 34 vars */) = 0
brk(NULL)                               = 0x559a8e7b9000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc9fdb4020) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f83c63b5000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=22187, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 22187, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f83c63af000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"..., 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f83c6186000
```

```
mprotect(0x7f83c61ae000, 2023424, PROT_NONE) = 0
mmap(0x7f83c61ae000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
 0x7f83c61ae000
mmap(0x7f83c6343000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f83c6343000
mmap(0x7f83c639c000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) =
0x7f83c639c000
mmap(0x7f83c63a2000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7f83c63a2000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f83c6183000
arch_prctl(ARCH_SET_FS, 0x7f83c6183740) = 0
set_tid_address(0x7f83c6183a10)         = 139005
set_robust_list(0x7f83c6183a20, 24)     = 0
rseq(0x7f83c61840e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f83c639c000, 16384, PROT_READ) = 0
mprotect(0x559a68f06000, 4096, PROT_READ) = 0
mprotect(0x7f83c63ef000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f83c63af000, 22187)           = 0
openat(AT_FDCWD, "shared_memory.bin", O_RDWR|O_CREAT|O_TRUNC, 0666) = 3 - открытие файла
ftruncate(3, 1096)                      = 0
mmap(NULL, 1096, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7f83c63ee000
close(3)                                = 0
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace: Process 139006 attached
, child_tidptr=0x7f83c6183a10) = 139006
[pid 139006] set_robust_list(0x7f83c6183a20, 24 <unfinished ...>
[pid 139005] clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD <unfinished ...>
[pid 139006] <... set_robust_list resumed>) = 0
strace: Process 139007 attached
[pid 139006] execve("./child1", ["child1", "shared_memory.bin"], 0x7ffc9fdb41f8 /* 34 vars */ <unfinished
...>
[pid 139005] <... clone resumed>, child_tidptr=0x7f83c6183a10) = 139007
[pid 139007] set_robust_list(0x7f83c6183a20, 24 <unfinished ...>
[pid 139005] newfstatat(1, "",  <unfinished ...>
[pid 139007] <... set_robust_list resumed>) = 0
[pid 139005] <... newfstatat resumed>{st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0xb), ...}, AT_EMPTY_PATH)
= 0
[pid 139007] execve("./child2", ["child2", "shared_memory.bin"], 0x7ffc9fdb41f8 /* 34 vars */ <unfinished
...>
[pid 139005] getrandom("\x36\x60\xb8\x86\x9c\x7e\xd9\xfe", 8, GRND_NONBLOCK) = 8
[pid 139005] brk(NULL)                  = 0x559a8e7b9000
[pid 139005] brk(0x559a8e7da000)        = 0x559a8e7da000
[pid 139006] <... execve resumed>)      = 0
[pid 139005] write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\320\272\320\270 (Ctr"..., 66 <unfinished ...>
[pid 139006] brk(NULL <unfinished ...>
[pid 139005] <... write resumed>)       = 66
[pid 139005] newfstatat(0, "",  <unfinished ...>
[pid 139006] <... brk resumed>)         = 0x560153a59000
[pid 139005] <... newfstatat resumed>{st_mode=S_IFIFO|0600, st_size=0, ...}, AT_EMPTY_PATH) = 0
[pid 139007] <... execve resumed>)      = 0
[pid 139005] read(0,  <unfinished ...>
[pid 139006] arch_prctl(0x3001 /* ARCH_??? */, 0x7fff443314e0 <unfinished ...>
```

```
[pid 139005] <... read resumed>"abo1ba    ba\nqwer3ty     ty\nagag"..., 4096) = 49
[pid 139007] brk(NULL <unfinished ...>
[pid 139005] futex(0x7f83c63ee428, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
<unfinished ...>
[pid 139006] <... arch_prctl resumed>)  = -1 EINVAL (Invalid argument)
[pid 139007] <... brk resumed>)         = 0x55d15b9ec000
[pid 139006] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 139007] arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd05b91d80 <unfinished ...>
[pid 139006] <... mmap resumed>)        = 0x7f3ed7017000
[pid 139007] <... arch_prctl resumed>)  = -1 EINVAL (Invalid argument)
[pid 139006] access("/etc/ld.so.preload", R_OK <unfinished ...>
[pid 139007] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 139006] <... access resumed>)      = -1 ENOENT (No such file or directory)
[pid 139007] <... mmap resumed>)        = 0x7fbfeee21000
[pid 139006] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>
[pid 139007] access("/etc/ld.so.preload", R_OK <unfinished ...>
[pid 139006] <... openat resumed>)      = 3
[pid 139007] <... access resumed>)      = -1 ENOENT (No such file or directory)
[pid 139006] newfstatat(3, "",  <unfinished ...>
[pid 139007] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>
[pid 139006] <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=22187, ...}, AT_EMPTY_PATH) = 0
[pid 139007] <... openat resumed>)      = 3
[pid 139006] mmap(NULL, 22187, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>
[pid 139007] newfstatat(3, "",  <unfinished ...>
[pid 139006] <... mmap resumed>)        = 0x7f3ed7011000
[pid 139007] <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=22187, ...}, AT_EMPTY_PATH) = 0
[pid 139006] close(3 <unfinished ...>
[pid 139007] mmap(NULL, 22187, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>
[pid 139006] <... close resumed>)       = 0
[pid 139007] <... mmap resumed>)        = 0x7fbfeee1b000
[pid 139006] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC <unfinished ...>
[pid 139007] close(3)                   = 0
[pid 139006] <... openat resumed>)      = 3
[pid 139007] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC <unfinished ...>
[pid 139006] read(3,  <unfinished ...>
[pid 139007] <... openat resumed>)      = 3
[pid 139006] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"..., 832) =
832
[pid 139007] read(3,  <unfinished ...>
[pid 139006] pread64(3,  <unfinished ...>
[pid 139007] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"..., 832) =
832
[pid 139006] <... pread64 resumed>"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784
[pid 139007] pread64(3,  <unfinished ...>
[pid 139006] pread64(3,  <unfinished ...>
[pid 139007] <... pread64 resumed>"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784
[pid 139006] <... pread64 resumed>"\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 48,
848) = 48
[pid 139007] pread64(3,  <unfinished ...>
[pid 139006] pread64(3,  <unfinished ...>
[pid 139007] <... pread64 resumed>"\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 48,
```

```
848) = 48
[pid 139006] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"..., 68, 896) = 68
[pid 139007] pread64(3,  <unfinished ...>
[pid 139006] newfstatat(3, "",  <unfinished ...>
[pid 139007] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"..., 68, 896) = 68
[pid 139006] <... newfstatat resumed>{st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 0
[pid 139007] newfstatat(3, "",  <unfinished ...>
[pid 139006] pread64(3,  <unfinished ...>
[pid 139007] <... newfstatat resumed>{st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 0
[pid 139006] <... pread64 resumed>"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784
[pid 139007] pread64(3,  <unfinished ...>
[pid 139006] mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0 <unfinished ...>
[pid 139007] <... pread64 resumed>"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784,
64) = 784
[pid 139006] <... mmap resumed>)        = 0x7f3ed6de8000
[pid 139007] mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0 <unfinished ...>
[pid 139006] mprotect(0x7f3ed6e10000, 2023424, PROT_NONE <unfinished ...>
[pid 139007] <... mmap resumed>)        = 0x7fbfeebf2000
[pid 139006] <... mprotect resumed>)    = 0
[pid 139007] mprotect(0x7fbfeec1a000, 2023424, PROT_NONE <unfinished ...>
[pid 139006] mmap(0x7f3ed6e10000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x28000 <unfinished ...>
[pid 139007] <... mprotect resumed>)    = 0
[pid 139006] <... mmap resumed>)        = 0x7f3ed6e10000
[pid 139007] mmap(0x7fbfeec1a000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
 0x28000 <unfinished ...>
[pid 139006] mmap(0x7f3ed6fa5000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000
<unfinished ...>
[pid 139007] <... mmap resumed>)        = 0x7fbfeec1a000
[pid 139006] <... mmap resumed>)        = 0x7f3ed6fa5000
[pid 139007] mmap(0x7fbfeedaf000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000
<unfinished ...>
[pid 139006] mmap(0x7f3ed6ffe000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x215000 <unfinished ...>
[pid 139007] <... mmap resumed>)        = 0x7fbfeedaf000
[pid 139006] <... mmap resumed>)        = 0x7f3ed6ffe000
[pid 139007] mmap(0x7fbfeee08000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x215000 <unfinished ...>
[pid 139006] mmap(0x7f3ed7004000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0
<unfinished ...>
[pid 139007] <... mmap resumed>)        = 0x7fbfeee08000
[pid 139006] <... mmap resumed>)        = 0x7f3ed7004000
[pid 139007] mmap(0x7fbfeee0e000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0
<unfinished ...>
[pid 139006] close(3 <unfinished ...>
[pid 139007] <... mmap resumed>)        = 0x7fbfeee0e000
[pid 139006] <... close resumed>)       = 0
[pid 139007] close(3 <unfinished ...>
[pid 139006] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 139007] <... close resumed>)       = 0
```

```
[pid 139006] <... mmap resumed>)        = 0x7f3ed6de5000
[pid 139007] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 139006] arch_prctl(ARCH_SET_FS, 0x7f3ed6de5740 <unfinished ...>
[pid 139007] <... mmap resumed>)        = 0x7fbfeebef000
[pid 139006] <... arch_prctl resumed>)  = 0
[pid 139007] arch_prctl(ARCH_SET_FS, 0x7fbfeebef740 <unfinished ...>
[pid 139006] set_tid_address(0x7f3ed6de5a10 <unfinished ...>
[pid 139007] <... arch_prctl resumed>)  = 0
[pid 139006] <... set_tid_address resumed>) = 139006
[pid 139007] set_tid_address(0x7fbfeebefa10 <unfinished ...>
[pid 139006] set_robust_list(0x7f3ed6de5a20, 24 <unfinished ...>
[pid 139007] <... set_tid_address resumed>) = 139007
[pid 139006] <... set_robust_list resumed>) = 0
[pid 139007] set_robust_list(0x7fbfeebefa20, 24 <unfinished ...>
[pid 139006] rseq(0x7f3ed6de60e0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 139007] <... set_robust_list resumed>) = 0
[pid 139006] <... rseq resumed>)        = 0
[pid 139007] rseq(0x7fbfeebf00e0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 139006] mprotect(0x7f3ed6ffe000, 16384, PROT_READ <unfinished ...>
[pid 139007] <... rseq resumed>)        = 0
[pid 139006] <... mprotect resumed>)    = 0
[pid 139007] mprotect(0x7fbfeee08000, 16384, PROT_READ <unfinished ...>
[pid 139006] mprotect(0x56014680a000, 4096, PROT_READ <unfinished ...>
[pid 139007] <... mprotect resumed>)    = 0
[pid 139006] <... mprotect resumed>)    = 0
[pid 139007] mprotect(0x55d120064000, 4096, PROT_READ <unfinished ...>
[pid 139006] mprotect(0x7f3ed7051000, 8192, PROT_READ <unfinished ...>
[pid 139007] <... mprotect resumed>)    = 0
[pid 139006] <... mprotect resumed>)    = 0
[pid 139007] mprotect(0x7fbfeee5b000, 8192, PROT_READ <unfinished ...>
[pid 139006] prlimit64(0, RLIMIT_STACK, NULL,  <unfinished ...>
[pid 139007] <... mprotect resumed>)    = 0
[pid 139006] <... prlimit64 resumed>{rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
[pid 139007] prlimit64(0, RLIMIT_STACK, NULL,  <unfinished ...>
[pid 139006] munmap(0x7f3ed7011000, 22187 <unfinished ...>
[pid 139007] <... prlimit64 resumed>{rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
[pid 139006] <... munmap resumed>)      = 0
[pid 139007] munmap(0x7fbfeee1b000, 22187 <unfinished ...>
[pid 139006] openat(AT_FDCWD, "shared_memory.bin", O_RDWR <unfinished ...>
[pid 139007] <... munmap resumed>)      = 0
[pid 139007] openat(AT_FDCWD, "shared_memory.bin", O_RDWR <unfinished ...>
[pid 139006] <... openat resumed>)      = 3
[pid 139007] <... openat resumed>)      = 3
[pid 139006] mmap(NULL, 1096, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0 <unfinished ...>
[pid 139007] mmap(NULL, 1096, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0 <unfinished ...>
[pid 139006] <... mmap resumed>)        = 0x7f3ed7050000
[pid 139007] <... mmap resumed>)        = 0x7fbfeee5a000
[pid 139006] close(3 <unfinished ...>
[pid 139007] close(3 <unfinished ...>
[pid 139006] <... close resumed>)       = 0
[pid 139007] <... close resumed>)       = 0
[pid 139006] futex(0x7f3ed70503e8, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
<unfinished ...>
```

```
[pid 139007] futex(0x7fbfeee5a428, FUTEX_WAKE, 1 <unfinished ...>
[pid 139005] <... futex resumed>)       = 0
[pid 139007] <... futex resumed>)       = 1
[pid 139005] write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202: ABO1BA
BA\n", 30 <unfinished ...>
[pid 139007] futex(0x7fbfeee5a408, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
<unfinished ...>
[pid 139005] <... write resumed>)       = 30
[pid 139005] write(1, "\n", 1)          = 1
[pid 139005] futex(0x7f83c63ee3e8, FUTEX_WAKE, 1) = 1
[pid 139006] <... futex resumed>)       = 0
[pid 139005] futex(0x7f83c63ee428, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
<unfinished ...>
[pid 139006] futex(0x7f3ed7050408, FUTEX_WAKE, 1 <unfinished ...>
[pid 139007] <... futex resumed>)       = 0
[pid 139006] <... futex resumed>)       = 1
[pid 139007] futex(0x7fbfeee5a428, FUTEX_WAKE, 1 <unfinished ...>
[pid 139006] futex(0x7f3ed70503e8, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
 <unfinished ...>
[pid 139005] <... futex resumed>)       = 0
[pid 139007] <... futex resumed>)       = 1
[pid 139005] write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202: QWER3TY
TY\n", 31 <unfinished ...>
[pid 139007] futex(0x7fbfeee5a408, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
<unfinished ...>
[pid 139005] <... write resumed>)       = 31
[pid 139005] write(1, "\n", 1)          = 1
[pid 139005] futex(0x7f83c63ee3e8, FUTEX_WAKE, 1) = 1
[pid 139006] <... futex resumed>)       = 0
[pid 139005] futex(0x7f83c63ee428, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
 <unfinished ...>
[pid 139006] futex(0x7f3ed7050408, FUTEX_WAKE, 1 <unfinished ...>
[pid 139007] <... futex resumed>)       = 0
[pid 139006] <... futex resumed>)       = 1
[pid 139007] futex(0x7fbfeee5a428, FUTEX_WAKE, 1 <unfinished ...>
[pid 139006] futex(0x7f3ed70503e8, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
<unfinished ...>
[pid 139005] <... futex resumed>)       = 0
[pid 139007] <... futex resumed>)       = 1
[pid 139005] write(1, "\320\240\320\265\320\267\321\203\320\273\321\214\321\202\320\260\321\202: AGAG48AGAG
A"..., 39 <unfinished ...>
[pid 139007] futex(0x7fbfeee5a408, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
<unfinished ...>
[pid 139005] <... write resumed>)       = 39
[pid 139005] write(1, "\n", 1)          = 1
[pid 139005] read(0, "", 4096)          = 0
[pid 139005] futex(0x7f83c63ee3e8, FUTEX_WAKE, 1) = 1
[pid 139006] <... futex resumed>)       = 0
[pid 139005] futex(0x7f83c63ee428, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY
<unfinished ...>
[pid 139006] futex(0x7f3ed7050408, FUTEX_WAKE, 1 <unfinished ...>
[pid 139007] <... futex resumed>)       = 0
[pid 139006] <... futex resumed>)       = 1
```

```
[pid 139007] futex(0x7fbfeee5a428, FUTEX_WAKE, 1 <unfinished ...>
[pid 139006] munmap(0x7f3ed7050000, 1096 <unfinished ...>
[pid 139005] <... futex resumed>)       = 0
[pid 139007] <... futex resumed>)       = 1
[pid 139005] wait4(139006,  <unfinished ...>
[pid 139006] <... munmap resumed>)      = 0
[pid 139007] munmap(0x7fbfeee5a000, 1096 <unfinished ...>
[pid 139006] exit_group(0 <unfinished ...>
[pid 139007] <... munmap resumed>)      = 0
[pid 139006] <... exit_group resumed>)  = ?
[pid 139007] exit_group(0)              = ?
[pid 139006] +++ exited with 0 +++
[pid 139005] <... wait4 resumed>[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 139006
[pid 139007] +++ exited with 0 +++
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=139006, si_uid=1000, si_status=0, si_utime=0,
si_stime=0} ---
wait4(139007, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 139007
munmap(0x7f83c63ee000, 1096)            = 0
write(1, "\n", 1)                       = 1
write(1, "\320\222\321\201\320\265 \320\277\321\200\320\276\321\206\320\265\321\201\321\201\321\213
\320\267\320\260\320\262\320\265"..., 44) = 44
exit_group(0)                           = ?
+++ exited with 0 +++
```

Вывод

Лабораторная работа демонстрирует использование разделяемой памяти (shared memory) и семафоров для взаимодействия между процессами в Linux. Программа корректно обрабатывает ввод пользователя, синхронизирует работу процессов и завершается без утечек ресурсов.