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-216БВ-24

Студент: Настина М.О.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 20.11.25

Москва, 2024

# Постановка задачи

**Вариант 18.**

Реализовать родительский и дочерний процессы, взаимодействующие через именованную разделяемую память POSIX и именованные семафоры. Дочерний процесс принимает строку, удаляет из неё гласные символы, выводит результат и записывает его в свой файл.

# Общий метод и алгоритм решения

Использованные системные вызовы:

- shm_open(name, flags, mode) - создаёт именованный объект разделяемой памяти в RAM.
- ftruncate(fd, size) - задаёт размер объекта shared memory.
- mmap() - отображает объект shared memory в адресное пространство процесса, так что можно работать с ним как с обычной структурой в памяти.
- sem_open(name, flags, mode, value) - создаёт или открывает именованный семафор.
- sem_wait(sem) - блокирует процесс, пока семафор не станет > 0 и уменьшает его.
- sem_post(sem) - увеличивает значение семафора → разблокирует ожидающие процессы.
- sem_close - удаляет именованный семафор.
- sem_unlink - закрывает именованный семафор.
- shm_unlink(name) - удаляет именованный объект shared memory.

В рамках лабораторной работы необходимо было реализовать родительский процесс, задачей которого было создать два дочерних процесса, для них создать собственный сегмент разделяемой памяти и два именованных семафора, передавать строки дочерним процессам через shared memory, и дочерние процессы, каждый из которых должен был ожидать строку от родителя через семафор и удалять из строки все гласные буквы.

# Код программы

**Parent.c:**

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/mman.h>

#include <semaphore.h>

#include <sys/wait.h>

#include <ctype.h>

#include <time.h>
```

```
#define SHM_SIZE 4096

struct shm_block {
    size_t len;
    char buffer[SHM_SIZE];
};

void random_name(char *buf, const char *prefix) {
    sprintf(buf, "/%s_%d_%ld", prefix, getpid(), random());
}

int main() {
    srandom(time(NULL));

    char *file1 = NULL;
    char *file2 = NULL;
    size_t len = 0;

    printf("Enter first output filename: ");
    fflush(stdout);
    if (getline(&file1, &len, stdin) == -1) {
        perror("getline file1");
        return 1;
    }
    file1[strcspn(file1, "\n")] = 0;

    printf("Enter second output filename: ");
    fflush(stdout);
    if (getline(&file2, &len, stdin) == -1) {
```

```
    perror("getline file2");
    free(file1);
    return 1;
}
file2[strcspn(file2, "\n")] = 0;

char shm1_name[128], shm2_name[128];
char sem1_base[128], sem2_base[128];

random_name(shm1_name, "shm1");
random_name(shm2_name, "shm2");
random_name(sem1_base, "sem1");
random_name(sem2_base, "sem2");

int shm1_fd = shm_open(shm1_name, O_CREAT | O_RDWR, 0666);
ftruncate(shm1_fd, sizeof(struct shm_block));
struct shm_block *shm1 = mmap(NULL, sizeof(struct shm_block),
                        PROT_READ | PROT_WRITE,
                        MAP_SHARED, shm1_fd, 0);

char sem1_parent_name[256], sem1_child_name[256];
sprintf(sem1_parent_name, "/%s_parent", sem1_base);
sprintf(sem1_child_name,  "/%s_child",  sem1_base);

sem_t *sem1_parent = sem_open(sem1_parent_name, O_CREAT, 0666, 1);
sem_t *sem1_child  = sem_open(sem1_child_name,  O_CREAT, 0666, 0);

pid_t pid1 = fork();
if (pid1 == 0) {
    execl("./child", "child", file1, shm1_name, sem1_base, NULL);
```

```
    perror("execl child1");
    exit(1);
}

int shm2_fd = shm_open(shm2_name, O_CREAT | O_RDWR, 0666);
ftruncate(shm2_fd, sizeof(struct shm_block));
struct shm_block *shm2 = mmap(NULL, sizeof(struct shm_block),
                              PROT_READ | PROT_WRITE,
                              MAP_SHARED, shm2_fd, 0);

char sem2_parent_name[256], sem2_child_name[256];
sprintf(sem2_parent_name, "/%s_parent", sem2_base);
sprintf(sem2_child_name,  "/%s_child",  sem2_base);

sem_t *sem2_parent = sem_open(sem2_parent_name, O_CREAT, 0666, 1);
sem_t *sem2_child  = sem_open(sem2_child_name,  O_CREAT, 0666, 0);

pid_t pid2 = fork();
if (pid2 == 0) {
    execl("./child", "child", file2, shm2_name, sem2_base, NULL);
    perror("execl child2");
    exit(1);
}
printf("Enter lines to process (Ctrl+D to finish):\n");
fflush(stdout);
char *line = NULL;
size_t line_len = 0;
int line_num = 1;

while (getline(&line, &line_len, stdin) != -1) {
```

```
    line[strcspn(line, "\n")] = 0;

    if (line_num % 2 == 1) {
        sem_wait(sem1_parent);
        shm1->len = strlen(line);
        strncpy(shm1->buffer, line, SHM_SIZE - 1);
        shm1->buffer[SHM_SIZE - 1] = '\0';
        sem_post(sem1_child);
    } else {
        sem_wait(sem2_parent);
        shm2->len = strlen(line);
        strncpy(shm2->buffer, line, SHM_SIZE - 1);
        shm2->buffer[SHM_SIZE - 1] = '\0';
        sem_post(sem2_child);
    }

    line_num++;
}
free(line);
free(file1);
free(file2);
sem_wait(sem1_parent);
shm1->len = 0;
sem_post(sem1_child);
sem_wait(sem2_parent);
shm2->len = 0;
sem_post(sem2_child);
waitpid(pid1, NULL, 0);
waitpid(pid2, NULL, 0);
munmap(shm1, sizeof(struct shm_block));
```

```
    munmap(shm2, sizeof(struct shm_block));
    close(shm1_fd);
    close(shm2_fd);
    shm_unlink(shm1_name);
    shm_unlink(shm2_name);
    sem_close(sem1_parent);
    sem_close(sem1_child);
    sem_close(sem2_parent);
    sem_close(sem2_child);
    sem_unlink(sem1_parent_name);
    sem_unlink(sem1_child_name);
    sem_unlink(sem2_parent_name);
    sem_unlink(sem2_child_name);
    return 0;
}
```

**Child.c:**

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

#define SHM_SIZE 4096

struct shm_block {
    size_t len;
    char buffer[SHM_SIZE];
```

```
};
int is_vowel(char c) {
    c = tolower(c);
    return (c=='a'||c=='e'||c=='i'||c=='o'||c=='u');
}
char* remove_vowels(const char *str) {
    size_t len = strlen(str);
    char *res = malloc(len+1);
    char *p = res;
    while (*str) {
        if (!is_vowel(*str))
            *p++ = *str;
        str++;
    }
    *p = '\0';
    return res;
}
int main(int argc, char argv) {
    if (argc != 4) {
        fprintf(stderr, "Usage: child <filename> <shm_name> <sem_base>\n");
        return 1;
    }
    char *filename = argv[1];
    char *shm_name = argv[2];
    char *sem_base = argv[3];
    char sem_parent_name[256], sem_child_name[256];
    sprintf(sem_parent_name, "/%s_parent", sem_base);
    sprintf(sem_child_name,  "/%s_child",  sem_base);
    int shm_fd = shm_open(shm_name, O_RDWR, 0666);
    if (shm_fd < 0) {
```

```
    perror("shm_open");
    return 1;
}
struct shm_block *shm = mmap(NULL, sizeof(struct shm_block),
                        PROT_READ | PROT_WRITE,
                        MAP_SHARED, shm_fd, 0);

if (shm == MAP_FAILED) {
    perror("mmap");
    return 1;
}
sem_t *sem_parent = sem_open(sem_parent_name, 0);
sem_t *sem_child  = sem_open(sem_child_name, 0);

FILE *out = fopen(filename, "w");
if (!out) {
    perror("fopen");
    return 1;
}

while (1) {
    sem_wait(sem_child);

    if (shm->len == 0) {
        break;
    }
    char *processed = remove_vowels(shm->buffer);
    char output[SHM_SIZE * 2];
    snprintf(output, sizeof(output), "Child processing: %s\n", processed);
    fputs(output, stdout);
```

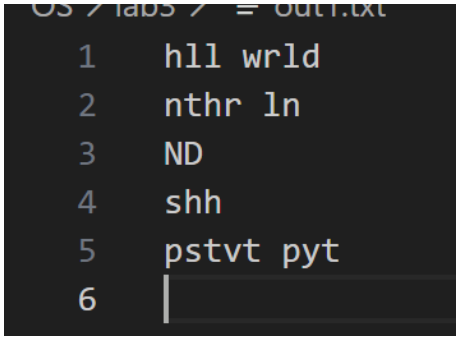
```
        fflush(stdout);
        fprintf(out, "%s\n", processed);
        fflush(out);
        free(processed);
        sem_post(sem_parent);
    }
    fclose(out);
    munmap(shm, sizeof(struct shm_block));
    sem_close(sem_parent);
    sem_close(sem_child);
    return 0;
}
```

# Протокол работы программы

**Тестирование:**

```
out2.txt
hello world
Child processing: hll wrld
this is test
Child processing: ths s tst
another line
Child processing: nthr ln
I LOVE LINUX
Child processing:  LV LNX
END
Child processing: ND
sky
Child processing: sky
shh
Child processing: shh
brrrrrrrrr
Child processing: brrrrrrrrr
postavte pyat
Child processing: pstvt pyt
```

```
OS > lab5 > = out1.txt
1   hll wrld
2   nthr ln
3   ND
4   shh
5   pstvt pyt
6
```




**Strace:**

execve("./parent", ["./parent"], 0x7fff9d0989a8 /* 37 vars */) = 0

brk(NULL) = 0x562e59654000

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f01a5d17000

access("/etc/ld.so.preload", R_OK) = -1 ENOENT (Нет такого файла или каталога)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=23303, ...}) = 0

mmap(NULL, 23303, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f01a5d11000

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\0\220\243\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

fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f01a5aff000

mmap(0x7f01a5b27000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f01a5b27000

mmap(0x7f01a5caf000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7f01a5caf000

mmap(0x7f01a5cfe000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7f01a5cfe000

mmap(0x7f01a5d04000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f01a5d04000

close(3) = 0

```
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f01a5afc000

arch_prctl(ARCH_SET_FS, 0x7f01a5afc740) = 0

set_tid_address(0x7f01a5afca10)         = 117488

set_robust_list(0x7f01a5afca20, 24)     = 0

rseq(0x7f01a5afd060, 0x20, 0, 0x53053053) = 0

mprotect(0x7f01a5cfe000, 16384, PROT_READ) = 0

mprotect(0x562e2945c000, 4096, PROT_READ) = 0

mprotect(0x7f01a5d4f000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

munmap(0x7f01a5d11000, 23303)           = 0

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x5), ...}) = 0

getrandom("\x76\xae\x99\x84\x2d\x7e\xbc\x9c", 8, GRND_NONBLOCK) = 8

brk(NULL)                               = 0x562e59654000

brk(0x562e59675000)                     = 0x562e59675000

write(1, "Enter first output filename: ", 29Enter first output filename: ) = 29

fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x5), ...}) = 0

read(0, 0x562e59654730, 1024)           = ? ERESTARTSYS (To be restarted if SA_RESTART is set)

--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---

read(0, 0x562e59654730, 1024)           = ? ERESTARTSYS (To be restarted if SA_RESTART is set)

--- SIGWINCH {si_signo=SIGWINCH, si_code=SI_KERNEL} ---

read(0, out1.txt

"out1.txt\n", 1024)             = 9

write(1, "Enter second output filename: ", 30Enter second output filename: ) = 30

read(0, out2.txt

"out2.txt\n", 1024)             = 9
```

getpid() = 117488

getpid() = 117488

getpid() = 117488

getpid() = 117488

**openat(AT_FDCWD, "/dev/shm/shm1_117488_1118530818", O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC, 0666) = 3**

**ftruncate(3, 4104) = 0**

**mmap(NULL, 4104, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7f01a5d15000**

openat(AT_FDCWD, "/dev/shm/sem.sem1_117488_1489926043_parent", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

getrandom("\xfc\x43\xce\xf3\x74\x95\xa2\x5f", 8, GRND_NONBLOCK) = 8

newfstatat(AT_FDCWD, "/dev/shm/sem.AtkCwH", 0x7ffc1faee8c0, AT_SYMLINK_NOFOLLOW) = -1 ENOENT (Нет такого файла или каталога)

**openat(AT_FDCWD, "/dev/shm/sem.AtkCwH", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC, 0666) = 4**

write(4, "\1\0\0\0\0\0\0\200\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32) = 32

**mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f01a5d14000**

**link("/dev/shm/sem.AtkCwH", "/dev/shm/sem.sem1_117488_1489926043_parent") = 0**

fstat(4, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0

**unlink("/dev/shm/sem.AtkCwH") = 0**

**sem_close(4) = 0**

openat(AT_FDCWD, "/dev/shm/sem.sem1_117488_1489926043_child", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = -1 ENOENT (Нет такого файла или каталога)

getrandom("\xfc\x1b\x89\x9a\x5a\x84\xb8\x51", 8, GRND_NONBLOCK) = 8

newfstatat(AT_FDCWD, "/dev/shm/sem.aybDNf", 0x7ffc1faee8c0, AT_SYMLINK_NOFOLLOW) = -1 ENOENT (Нет такого файла или каталога)

**openat(AT_FDCWD, "/dev/shm/sem.aybDNf", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC, 0666) = 4**

write(4, "\0\0\0\0\0\0\0\200\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32) = 32

**mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f01a5d13000**

**link("/dev/shm/sem.aybDNf", "/dev/shm/sem.sem1_117488_1489926043_child") = 0**

fstat(4, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0

**unlink("/dev/shm/sem.aybDNf")  = 0**

**sem_close(4)  = 0**

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace: Process 117793 attached

, child_tidptr=0x7f01a5afca10) = 117793

[pid 117793] set_robust_list(0x7f01a5afca20, 24 <unfinished ...>

[pid 117488] **openat(AT_FDCWD, "/dev/shm/shm2_117488_112854500", O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC <unfinished ...>**

[pid 117793] <... set_robust_list resumed>) = 0

[pid 117488] <... openat resumed>)  = 4

[pid 117793] execve("./child", ["child", "out1.txt", "/shm1_117488_1118530818", "/sem1_117488_1489926043"], 0x7ffc1faef408 /* 37 vars */ <unfinished ...>

[pid 117488] **ftruncate(4, 4104)  = 0**

[pid 117488] **mmap(NULL, 4104, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>**

[pid 117793] <... execve resumed>)  = 0

[pid 117488] <... mmap resumed>)  = 0x7f01a5d11000

[pid 117488] openat(AT_FDCWD, "/dev/shm/sem.sem2_117488_1706163573_parent", O_RDWR|O_NOFOLLOW|O_CLOEXEC <unfinished ...>

[pid 117793] brk(NULL <unfinished ...>

[pid 117488] <... openat resumed>)  = -1 ENOENT (Нет такого файла или каталога)

[pid 117793] <... brk resumed>)  = 0x56093f354000

[pid 117488] getrandom("\xfe\x9b\x07\xdb\xfa\x8a\xa5\x1c", 8, GRND_NONBLOCK) = 8

[pid 117793] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 117488] newfstatat(AT_FDCWD, "/dev/shm/sem.y3LmoU",  <unfinished ...>

[pid 117793] <... mmap resumed>)  = 0x7f99322a7000

[pid 117488] <... newfstatat resumed>0x7ffc1faee8c0, AT_SYMLINK_NOFOLLOW) = -1 ENOENT (Нет такого файла или каталога)

[pid 117793] access("/etc/ld.so.preload", R_OK <unfinished ...>

[pid 117488] **openat(AT_FDCWD, "/dev/shm/sem.y3LmoU", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC, 0666 <unfinished ...>**

[pid 117793] <... access resumed>) = -1 ENOENT (Нет такого файла или каталога)

[pid 117488] <... openat resumed>) = 5

[pid 117793] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 117488] write(5, "\1\0\0\0\0\0\0\200\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32 <unfinished ...>

[pid 117793] <... openat resumed>) = 3

[pid 117488] <... write resumed>) = 32

[pid 117793] fstat(3, <unfinished ...>

[pid 117488] **mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 5, 0 <unfinished ...>**

[pid 117793] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=23303, ...}) = 0

[pid 117488] <... mmap resumed>) = 0x7f01a5afb000

[pid 117488] **link("/dev/shm/sem.y3LmoU", "/dev/shm/sem.sem2_117488_1706163573_parent" <unfinished ...>**

[pid 117793] mmap(NULL, 23303, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>

[pid 117488] <... link resumed>) = 0

[pid 117793] <... mmap resumed>) = 0x7f99322a1000

[pid 117488] fstat(5, <unfinished ...>

[pid 117793] close(3 <unfinished ...>

[pid 117488] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=32, ...}) = 0

[pid 117793] <... close resumed>) = 0

[pid 117488] **unlink("/dev/shm/sem.y3LmoU" <unfinished ...>**

[pid 117793] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 117488] <... unlink resumed>) = 0

[pid 117793] <... openat resumed>) = 3

[pid 117488] **sem_close(5 <unfinished ...>**

[pid 117793] read(3,  <unfinished ...>

[pid 117488] <... close resumed>)       = 0

[pid 117488] openat(AT_FDCWD, "/dev/shm/sem.sem2_117488_1706163573_child", O_RDWR|O_NOFOLLOW|O_CLOEXEC <unfinished ...>

[pid 117793] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0"..., 832) = 832

[pid 117488] <... openat resumed>)      = -1 ENOENT (Нет такого файла или каталога)

[pid 117793] pread64(3,  <unfinished ...>

[pid 117488] getrandom( <unfinished ...>

[pid 117793] <... 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 117488] <... getrandom resumed>"\xe3\xe4\x1f\x5c\x44\xd8\x76\xa5", 8, GRND_NONBLOCK) = 8

[pid 117793] fstat(3,  <unfinished ...>

[pid 117488] newfstatat(AT_FDCWD, "/dev/shm/sem.p8Dlfv",  <unfinished ...>

[pid 117793] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

[pid 117488] <... newfstatat resumed>0x7ffc1faee8c0, AT_SYMLINK_NOFOLLOW) = -1 ENOENT (Нет такого файла или каталога)

[pid 117793] pread64(3,  <unfinished ...>

[pid 117488] **openat(AT_FDCWD, "/dev/shm/sem.p8Dlfv", O_RDWR|O_CREAT|O_EXCL|O_NOFOLLOW|O_CLOEXEC, 0666 <unfinished ...>**

[pid 117793] <... 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 117793] mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0 <unfinished ...>

[pid 117488] <... openat resumed>)      = 5

[pid 117793] <... mmap resumed>)        = 0x7f993208f000

[pid 117488] write(5, "\0\0\0\0\0\0\0\200\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0", 32 <unfinished ...>

[pid 117793] mmap(0x7f99320b7000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000 <unfinished ...>

[pid 117488] <... write resumed>) = 32

[pid 117793] <... mmap resumed>) = 0x7f99320b7000

[pid 117488] **mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 5, 0 <unfinished ...>**

[pid 117793] mmap(0x7f993223f000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000 <unfinished ...>

[pid 117488] <... mmap resumed>) = 0x7f01a5afa000

[pid 117793] <... mmap resumed>) = 0x7f993223f000

[pid 117488] **link("/dev/shm/sem.p8Dlfv", "/dev/shm/sem.sem2_117488_1706163573_child" <unfinished ...>**

[pid 117793] mmap(0x7f993228e000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000 <unfinished ...>

[pid 117488] <... link resumed>) = 0

[pid 117793] <... mmap resumed>) = 0x7f993228e000

[pid 117488] fstat(5, <unfinished ...>

[pid 117793] mmap(0x7f9932294000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 117488] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=32, ...}) = 0

[pid 117793] <... mmap resumed>) = 0x7f9932294000

[pid 117488] unlink("/dev/shm/sem.p8Dlfv" <unfinished ...>

[pid 117793] close(3 <unfinished ...>

[pid 117488] <... unlink resumed>) = 0

[pid 117793] <... close resumed>) = 0

[pid 117488] **sem_close(5) = 0**

[pid 117793] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 117488] clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD <unfinished ...>

[pid 117793] <... mmap resumed>) = 0x7f993208c000

[pid 117793] arch_prctl(ARCH_SET_FS, 0x7f993208c740 <unfinished ...>

[pid 117488] <... clone resumed>, child_tidptr=0x7f01a5afca10) = 117794

```
strace: Process 117794 attached
[pid 117793] <... arch_prctl resumed>)  = 0
[pid 117488] write(1, "Enter lines to process (Ctrl+D t"..., 43 <unfinished ...>
Enter lines to process (Ctrl+D to finish):
[pid 117794] set_robust_list(0x7f01a5afca20, 24 <unfinished ...>
[pid 117793] set_tid_address(0x7f993208ca10 <unfinished ...>
[pid 117488] <... write resumed>)       = 43
[pid 117794] <... set_robust_list resumed>) = 0
[pid 117488] read(0,  <unfinished ...>
[pid 117793] <... set_tid_address resumed>) = 117793
[pid 117794] execve("./child", ["child", "out2.txt", "/shm2_117488_112854500",
"/sem2_117488_1706163573"], 0x7ffc1faef408 /* 37 vars */ <unfinished ...>
[pid 117793] set_robust_list(0x7f993208ca20, 24) = 0
[pid 117793] rseq(0x7f993208d060, 0x20, 0, 0x53053053) = 0
[pid 117794] <... execve resumed>)      = 0
[pid 117793] mprotect(0x7f993228e000, 16384, PROT_READ <unfinished ...>
[pid 117794] brk(NULL <unfinished ...>
[pid 117793] <... mprotect resumed>)    = 0
[pid 117794] <... brk resumed>)         = 0x5571ed85a000
[pid 117793] mprotect(0x560926be5000, 4096, PROT_READ <unfinished ...>
[pid 117794] mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 117793] <... mprotect resumed>)    = 0
[pid 117794] <... mmap resumed>)        = 0x7f4653fdc000
[pid 117793] mprotect(0x7f99322df000, 8192, PROT_READ <unfinished ...>
[pid 117794] access("/etc/ld.so.preload", R_OK <unfinished ...>
[pid 117793] <... mprotect resumed>)    = 0
[pid 117794] <... access resumed>)      = -1 ENOENT (Нет такого файла или каталога)
```

[pid 117793] prlimit64(0, RLIMIT_STACK, NULL,  <unfinished ...>

[pid 117794] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 117793] <... prlimit64 resumed>{rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

[pid 117794] <... openat resumed>)      = 3

[pid 117793] munmap(0x7f99322a1000, 23303 <unfinished ...>

[pid 117794] fstat(3,  <unfinished ...>

[pid 117793] <... munmap resumed>)      = 0

[pid 117794] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=23303, ...}) = 0

[pid 117793] **openat(AT_FDCWD, "/dev/shm/shm1_117488_1118530818", O_RDWR|O_NOFOLLOW|O_CLOEXEC <unfinished ...>**

[pid 117794] mmap(NULL, 23303, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>

[pid 117793] <... openat resumed>)      = 3

[pid 117794] <... mmap resumed>)        = 0x7f4653fd6000

[pid 117793] **mmap(NULL, 4104, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0 <unfinished ...>**

[pid 117794] close(3 <unfinished ...>

[pid 117793] <... mmap resumed>)        = 0x7f99322a5000

[pid 117794] <... close resumed>)       = 0

[pid 117793] openat(AT_FDCWD, "/dev/shm/sem.sem1_117488_1489926043_parent", O_RDWR|O_NOFOLLOW|O_CLOEXEC <unfinished ...>

[pid 117794] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 117793] <... openat resumed>)      = 4

[pid 117794] <... openat resumed>)      = 3

[pid 117793] fstat(4,  <unfinished ...>

[pid 117794] read(3,  <unfinished ...>

[pid 117793] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=32, ...}) = 0

[pid 117794] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"..., 832) = 832

[pid 117793] getrandom( <unfinished ...>

[pid 117794] pread64(3,  <unfinished ...>

[pid 117793] <... getrandom resumed>"\xca\x9c\x8a\xc1\x5f\x2d\xa2\x70", 8, GRND_NONBLOCK) = 8

[pid 117794] <... 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 117793] brk(NULL <unfinished ...>

[pid 117794] fstat(3,  <unfinished ...>

[pid 117793] <... brk resumed>)         = 0x56093f354000

[pid 117794] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

[pid 117793] brk(0x56093f375000 <unfinished ...>

[pid 117794] pread64(3,  <unfinished ...>

[pid 117793] <... brk resumed>)         = 0x56093f375000

[pid 117794] <... 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 117793] **mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>**

[pid 117794] mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0 <unfinished ...>

[pid 117793] <... mmap resumed>)        = 0x7f99322a4000

[pid 117794] <... mmap resumed>)        = 0x7f4653dc4000

[pid 117793] **sem_close(4 <unfinished ...>**

[pid 117794] mmap(0x7f4653dec000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000 <unfinished ...>

[pid 117793] <... close resumed>)       = 0

[pid 117794] <... mmap resumed>)        = 0x7f4653dec000

[pid 117793] openat(AT_FDCWD, "/dev/shm/sem.sem1_117488_1489926043_child", O_RDWR|O_NOFOLLOW|O_CLOEXEC <unfinished ...>

[pid 117794] mmap(0x7f4653f74000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7f4653f74000

[pid 117793] <... openat resumed>)      = 4

[pid 117794] mmap(0x7f4653fc3000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000 <unfinished ...>

[pid 117793] fstat(4,  <unfinished ...>

[pid 117794] <... mmap resumed>)        = 0x7f4653fc3000

[pid 117793] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=32, ...}) = 0

[pid 117793] **mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0 <unfinished ...>**

[pid 117794] mmap(0x7f4653fc9000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f4653fc9000

[pid 117793] <... mmap resumed>)        = 0x7f99322a3000

[pid 117794] close(3 <unfinished ...>

[pid 117793] **sem_close(4 <unfinished ...>**

[pid 117794] <... close resumed>)       = 0

[pid 117793] <... close resumed>)       = 0

[pid 117794] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 117793] openat(AT_FDCWD, "out1.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666 <unfinished ...>

[pid 117794] <... mmap resumed>)        = 0x7f4653dc1000

[pid 117794] arch_prctl(ARCH_SET_FS, 0x7f4653dc1740) = 0

[pid 117794] set_tid_address(0x7f4653dc1a10) = 117794

[pid 117794] set_robust_list(0x7f4653dc1a20, 24) = 0

[pid 117794] rseq(0x7f4653dc2060, 0x20, 0, 0x53053053) = 0

[pid 117794] mprotect(0x7f4653fc3000, 16384, PROT_READ) = 0

[pid 117794] mprotect(0x5571d2031000, 4096, PROT_READ) = 0

[pid 117794] mprotect(0x7f4654014000, 8192, PROT_READ) = 0

[pid 117794] prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

[pid 117794] munmap(0x7f4653fd6000, 23303) = 0

[pid 117794] **openat(AT_FDCWD, "/dev/shm/shm2_117488_112854500", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = 3**

[pid 117794] **mmap(NULL, 4104, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7f4653fda000**

[pid 117794] openat(AT_FDCWD, "/dev/shm/sem.sem2_117488_1706163573_parent", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = 4

[pid 117794] fstat(4, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0

[pid 117794] getrandom("\x9d\x42\x40\x60\xc0\x2e\x23\x8b", 8, GRND_NONBLOCK) = 8

[pid 117794] brk(NULL)                  = 0x5571ed85a000

[pid 117794] brk(0x5571ed87b000)        = 0x5571ed87b000

[pid 117794] **mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f4653fd9000**

[pid 117794] **sem_close(4)                = 0**

[pid 117794] openat(AT_FDCWD, "/dev/shm/sem.sem2_117488_1706163573_child", O_RDWR|O_NOFOLLOW|O_CLOEXEC) = 4

[pid 117794] fstat(4, {st_mode=S_IFREG|0644, st_size=32, ...}) = 0

[pid 117794] **mmap(NULL, 32, PROT_READ|PROT_WRITE, MAP_SHARED, 4, 0) = 0x7f4653fd8000**

[pid 117794] **sem_close(4)                = 0**

[pid 117794] openat(AT_FDCWD, "out2.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 4

[pid 117794] **sem_wait(0x7f4653fd8000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>**

[pid 117793] <... openat resumed>)      = 4

[pid 117793] **sem_wait(0x7f99322a3000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANYHello world**

**<unfinished ...>**

[pid 117488] <... read resumed>"Hello world\n", 1024) = 12

[pid 117488] **sem_post(0x7f01a5d13000, FUTEX_WAKE, 1) = 1**

[pid 117793] <... futex resumed>)       = 0

[pid 117488] read(0,  <unfinished ...>

[pid 117793] fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x5), ...}) = 0

[pid 117793] write(1, "Child processing: Hll wrld\n", 27Child processing: Hll wrld

) = 27

[pid 117793] fstat(4, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0

[pid 117793] write(4, "Hll wrld\n", 9)  = 9

[pid 117793] **sem_wait(0x7f99322a3000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANYThis is a test**

**<unfinished ...>**

[pid 117488] <... read resumed>"This is a test\n", 1024) = 15

[pid 117488] **sem_post(0x7f01a5afa000, FUTEX_WAKE, 1) = 1**

[pid 117794] <... futex resumed>)       = 0

[pid 117488] read(0,  <unfinished ...>

[pid 117794] fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x5), ...}) = 0

[pid 117794] write(1, "Child processing: Ths s  tst\n", 29Child processing: Ths s  tst

) = 29

[pid 117794] fstat(4, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0

[pid 117794] write(4, "Ths s  tst\n", 11) = 11

[pid 117794] **sem_wait(0x7f4653fd8000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANYAnother line**

**<unfinished ...>**

[pid 117488] <... read resumed>"Another line\n", 1024) = 13

[pid 117488] **sem_post(0x7f01a5d13000, FUTEX_WAKE, 1) = 1**

[pid 117793] <... futex resumed>)       = 0

[pid 117488] read(0,  <unfinished ...>

[pid 117793] write(1, "Child processing: nthr ln\n", 26Child processing: nthr ln

) = 26

[pid 117793] write(4, "nthr ln\n", 8)   = 8

[pid 117793] **sem_wait(0x7f99322a3000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANYLast line**

**<unfinished ...>**

[pid 117488] <... read resumed>"Last line\n", 1024) = 10

[pid 117488] **sem_post(0x7f01a5afa000, FUTEX_WAKE, 1) = 1**

[pid 117794] <... futex resumed>) = 0

[pid 117488] read(0, <unfinished ...>

[pid 117794] write(1, "Child processing: Lst ln\n", 25Child processing: Lst ln

) = 25

[pid 117794] write(4, "Lst ln\n", 7) = 7

[pid 117794] **sem_wait(0x7f4653fd8000, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 0, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>**

[pid 117488] <... read resumed>"", 1024) = 0

[pid 117488] **sem_post(0x7f01a5d13000, FUTEX_WAKE, 1) = 1**

[pid 117793] <... futex resumed>) = 0

[pid 117488] **sem_post(0x7f01a5afa000, FUTEX_WAKE, 1 <unfinished ...>**

[pid 117793] **sem_close(4 <unfinished ...>**

[pid 117488] <... futex resumed>) = 1

[pid 117794] <... futex resumed>) = 0

[pid 117793] <... close resumed>) = 0

[pid 117793] **mmap(0x7f99322a5000, 4104 <unfinished ...>**

[pid 117488] wait4(117793, <unfinished ...>

[pid 117794] **sem_close(4 <unfinished ...>**

[pid 117793] <... munmap resumed>) = 0

[pid 117794] <... close resumed>) = 0

[pid 117793] **mmap(0x7f99322a4000, 32 <unfinished ...>**

[pid 117794] **mmap(0x7f4653fda000, 4104 <unfinished ...>**

[pid 117793] <... munmap resumed>) = 0

[pid 117794] <... munmap resumed>) = 0

[pid 117794] **mmap(0x7f4653fd9000, 32 <unfinished ...>**

[pid 117793] **mmap(0x7f99322a3000, 32 <unfinished ...>**

[pid 117794] <... munmap resumed>) = 0

[pid 117793] <... munmap resumed>) = 0

[pid 117794] **mmap(0x7f4653fd8000, 32 <unfinished ...>**

[pid 117793] exit_group(0 <unfinished ...>

[pid 117794] <... munmap resumed>) = 0

[pid 117793] <... exit_group resumed>) = ?

[pid 117794] exit_group(0) = ?

[pid 117794] +++ exited with 0 +++

[pid 117793] +++ exited with 0 +++

<... wait4 resumed>NULL, 0, NULL) = 117793

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=117794, si_uid=1000, si_status=0, si_utime=0, si_stime=2 /* 0.02 s */} ---

wait4(117794, NULL, 0, NULL) = 117794

**mmap(0x7f01a5d15000, 4104) = 0**

**mmap(0x7f01a5d11000, 4104) = 0**

**sem_close(3) = 0**

**sem_close(4) = 0**

**shm_unlink("/dev/shm/shm1_117488_1118530818") = 0**

**shm_unlink("/dev/shm/shm2_117488_112854500") = 0**

**mmap(0x7f01a5d14000, 32) = 0**

**mmap(0x7f01a5d13000, 32) = 0**

**mmap(0x7f01a5afb000, 32) = 0**

**mmap(0x7f01a5afa000, 32) = 0**

**sem_unlink("/dev/shm/sem.sem1_117488_1489926043_parent") = 0**

**sem_unlink("/dev/shm/sem.sem1_117488_1489926043_child") = 0**

**sem_unlink("/dev/shm/sem.sem2_117488_1706163573_parent") = 0**

**sem_unlink("/dev/shm/sem.sem2_117488_1706163573_child") = 0**

exit_group(0) = ?

+++ exited with 0 +++

# Вывод

В лабораторной работе были реализованы взаимодействие процессов через разделяемую память (shared memory) и именованные семафоры POSIX. Родительский процесс создаёт два дочерних и каждому выделяет собственный сегмент shared memory и набор семафоров для синхронизации записи и чтения. Родитель читает строки от пользователя и передаёт их детям по очереди через shared memory, а дети обрабатывают строки, удаляют гласные и записывают результат в свои файлы. Таким образом реализовано межпроцессное взаимодействие без использования каналов, только через совместную память и семафоры.