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: Шаталов М.А.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 31.10.24

Москва, 2024

**Постановка задачи**

**Вариант 3.**

Родительский процесс создает два дочерних процесса. Первой строкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия File с таким именем на запись для child1. Аналогично для второй строки и процесса child2. Родительский и дочерний процесс должны быть представлены разными программами. parent child1 pipe1 In/out User In Out child2 In Out File Open with write mode pipe2 File Родительский процесс принимает от пользователя строки произвольной длины и пересылает их в дочерние процессы в зависимости от правила фильтрации. Процесс child1 и child2 производят работу над строками. Процессы пишут результаты своей работы в стандартный вывод.

Правило фильтрации: с вероятностью 80% строки отправляются в первый процесс, иначе во второй. Дочерние процессы удаляют все гласные из строк.

Взаимодействие между процессами осуществляется через системные сигналы/события и/или через отображаемые файлы (memory-mapped files).

**Общий метод и алгоритм решения**

Использованные системные вызовы:

* pid\_t fork(void); – создает дочерний процесс.
* int execl(const char \*path, const char \*arg, ...); - загружает и исполняет новый образ программы.
* close(int fd); - закрыть файл
* open(const char \*pathname, int flags, mode\_t mode); - открытие\создание файла
* exit(int status); - завершения выполнения процесса
* int shm\_open (const char \*name, int oflag, mode\_t mode); - создать или открыть разделяемую память
* int shm\_unlink (const char \*name);- удалить разделяемую память по имени
* int ftruncate (int fd, off\_t length); - изменить размер открытого файла
* void \*mmap (void \*addr, size\_t length, int prot, int flags, int fd, off\_t offset);-сопоставить область памяти с файлом
* int munmap (void \*addr, size\_t length); - отменить сопоставление области памяти
* sem\_t \*sem\_open (const char \*name, int oflag); - создать или открыть именованный семафор
* int sem\_post (sem\_t \*sem); - сигнализировать (разблокировать) семафор
* int sem\_wait (sem\_t \*sem); - ожидать (заблокироваться) на семафоре
* int sem\_unlink (const char \*name); - удалить именованный семафор

**Алгоритм работы программы**

Разделим алгоритм программы на этап инициализации и основной этап.

В блоке инициализации мы создаем и настраиваем отображаемые в память файлы(memory mapping), вводим имена файлов, создаем дочерние процессы, семафоры и так далее. В дочернем файле нужно будет повторить все действия настройки из родительского, за исключением того, что доченим процессам не требуется заново создавать файл отображения.

Дочерним процессам необходимо знать имена своих файлов, имя именного семафора и имя отображаемого файла для получения данных от родительского процесса. Все эти данные мы передадим из родительского процесса через флаги запуска программы child: int argc, char \*argv[].

Основная идея работы программы это два бесконечных цикла. Один в родительском процессе: он принимает строки и с вероятностью перенаправляет её либо в файл отображенный файл либо во второй. Цикл в дочернем процессе ждет пока выполнит отправку сигнала семафору и передаст введенную строку в файл, дальше процесс считывает строку из файла и выполняет действие со строкой: удаляет все гласные буквы и записывает строку в файл.

Условием выхода из циклов будет строка “!q”, обозначающая quit. После неё прерывается основной цикл программы, и в дочерние программы отправляется сигнал о завершении работы, отправляем сигнал с помощью записи в отображенный файл строки “!q”.

После завершения работы важно дождаться завершения работы детей и закрыть открытые дескрипторы.

Результат работы программы будет записан в файлы, чьи имена были переданы в начале работы родительского процесса.

Вероятность для отправки строк в разные файлы вычисляем с помощью функции rand().

Для синхронизации процессов и защиты программы от race condition будем использовать семафор. В момент записи родительского процесса в файл дочерний процесс ждет. После того как родительский процесс отправил сигнал семафору о том, что ресурс больше не используется

**Код программы**

**main.cpp**

#include <iostream>

#include <string>

#include <string.h>

#include <fcntl.h>

#include <sys/shm.h>

#include <sys/mman.h>

#include <semaphore.h>

#include <sys/wait.h>

// • O\_RDWR: Флаг для открытия файла для чтения и записи.  O\_RDONLY | O\_WRONLY == O\_RDWR

// • O\_CREAT: Флаг, который создает файл, если он не существует.

// • O\_TRUNC: Флаг, который обрезает файл до нулевого размера, если он уже существует.

int main()

{

    std::string f1, f2;

    std::cout << "Enter file name for child1: ";

    std::cin >> f1;

    std::cout << "Enter file name for child2: ";

    std::cin >> f2;

    const int SIZE = 4096; // size in bytes

    // create first shared memory file

    const char \*name1 = "OS1"; // shared name

    int shm\_fd1;               // shared memory file descriptor

    void \*ptr1;                // pointer to shared memory object

    shm\_fd1 = shm\_open(name1, O\_CREAT | O\_RDWR, 0666);

    ftruncate(shm\_fd1, SIZE);

    ptr1 = mmap(0, SIZE, PROT\_WRITE, MAP\_SHARED, shm\_fd1, 0);

    // create second shared memory file

    const char \*name2 = "OS2"; // shared name

    int shm\_fd2;

    void \*ptr2;

    shm\_fd2 = shm\_open(name2, O\_CREAT | O\_RDWR, 0666);

    ftruncate(shm\_fd2, SIZE);

    ptr2 = mmap(0, SIZE, PROT\_WRITE, MAP\_SHARED, shm\_fd2, 0);

    // Создаем семафоры

    const char \*sem1\_name = "/sem1";

    const char \*sem2\_name = "/sem2";

    sem\_t \*sem1 = sem\_open(sem1\_name, O\_CREAT | O\_EXCL, 0666, 1);

    sem\_t \*sem2 = sem\_open(sem2\_name, O\_CREAT | O\_EXCL, 0666, 1);

    // создаем детей

    pid\_t ch1 = fork();

    switch (ch1)

    {

    case -1: // error

        std::cout << "Ошибка при выполнении fork 1";

        exit(1);

        break;

    case 0: // child1 code

        execl("./child", "child1", f1.c\_str(), sem1\_name, name1, NULL);

        std::cout << "Ошибка при вызове execl 1";

        return 1;

    default: // dad

        pid\_t ch2 = fork();

        switch (ch2)

        {

        case -1: // error

            std::cout << "Ошибка при выполнении fork 1";

            exit(1);

            break;

        case 0: // child2 code

            execl("./child", "child2", f2.c\_str(), sem2\_name, name2, NULL);

            std::cout << "Ошибка при вызове execl 2";

            return 1;

        default: // dady code:)

            std::string st;

            std::cout << "Enter linens. To finish entering lines, enter '!q'.\n";

            while (1)

            {

                std::cin >> st;

                if (st == "!q")

                    break;

                if (rand() % 100 <= 80)

                {

                    sem\_wait(sem1);

                    strcpy((char \*)ptr1, st.c\_str());

                    sem\_post(sem1); // Сигнализируем, что файл1 свободен

                }

                else

                {

                    sem\_wait(sem2);

                    strcpy((char \*)ptr2, st.c\_str());

                    sem\_post(sem2); // Сигнализируем, что файл2 свободен

                }

            }

            // Отправляем сигнал завершения детям

            st = "!q";

            // Первый ребенок

            sem\_wait(sem1);

            strcpy((char \*)ptr1, st.c\_str());

            sem\_post(sem1);

            // Второй ребенок

            sem\_wait(sem2);

            strcpy((char \*)ptr2, st.c\_str());

            sem\_post(sem2);

            int status;

            waitpid(ch1, &status, 0);

            waitpid(ch2, &status, 0);

            // Освобождение ресурсов

            munmap(ptr1, SIZE);

            close(shm\_fd1);

            shm\_unlink(name1); // Удаление разделяемой памяти после завершения работы

            munmap(ptr2, SIZE);

            close(shm\_fd2);

            shm\_unlink(name2);

            // Закрытие семафоров

            sem\_close(sem1);

            sem\_close(sem2);

            sem\_unlink("/sem1");

            sem\_unlink("/sem2");

            break;

        }

        break;

    }

    return 0;

}

**Child.cpp**

#include <iostream>

#include <string>

#include <string.h>

#include <fcntl.h>

#include <sys/shm.h>

#include <sys/mman.h>

#include <semaphore.h>

#include <sys/wait.h>

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

{

    const char \*file\_name(argv[1]);

    const char \*semaphore\_name(argv[2]);

    const char \*shared\_mem\_name(argv[3]);

    int fd = open(file\_name, O\_RDWR | O\_CREAT | O\_TRUNC, 0666);

    const int SIZE = 4096; // Размер разделяемой памяти

    // Открытие семафора //err-1

    sem\_t \*sem = sem\_open(semaphore\_name, O\_RDWR, 0666);

    // Открытие shared memory //err-1

    int shm\_fd = shm\_open(shared\_mem\_name, O\_RDWR, 0666);

    // Сопоставление shared memory с памятью процесса/err-1

    void \*ptr = mmap(0, SIZE, PROT\_READ | PROT\_WRITE, MAP\_SHARED, shm\_fd, 0);

    while (1)

    {

        sem\_wait(sem);

        std::string st\_in((char \*)ptr);

        strcpy((char \*)ptr, "");

        sem\_post(sem);

        if (st\_in == "")

            continue;

        if (st\_in == "!q")

            break;

        std::string st\_out = "";

        for (int i = 0; i < st\_in.size(); i++)

        {

            if (!(st\_in[i] == 'a' || st\_in[i] == 'e' || st\_in[i] == 'i' || st\_in[i] == 'o' || st\_in[i] == 'u' || st\_in[i] == 'y' ||

                  st\_in[i] == 'A' || st\_in[i] == 'E' || st\_in[i] == 'I' || st\_in[i] == 'O' || st\_in[i] == 'U' || st\_in[i] == 'Y'))

            {

                st\_out += st\_in[i];

            }

        }

        st\_out += "\n";

        write(fd, st\_out.c\_str(), st\_out.size());

    }

    munmap(ptr, SIZE);

    close(shm\_fd);

    sem\_close(sem);

    sem\_unlink(semaphore\_name);

    close(fd);

    return 0;

}

**Протокол работы программы**

**Тестирование:**

root@3d4dec048ee8:/workspaces/MAI\_OS\_Labs/lab2/src# ./main

Enter file name for child1: f\_1

Enter file name for child2: f\_2

Enter linens. To finish entering lines, enter '!q'.

door

apple

tomato

table

answer

elefant

poor

rich

omega

q

root@3d4dec048ee8:/workspaces/MAI\_OS\_Labs/lab2/src# cat f\_1

tmt

tbl

lfnt

mg

root@3d4dec048ee8:/workspaces/MAI\_OS\_Labs/lab2/src# cat f\_2

dr

ppl

nswr

pr

rch

**Strace:**

execve("./main", ["./main"], 0x7ffde4631ef8 /\* 25 vars \*/) = 0

brk(NULL) = 0x1c47000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f380a4c6000

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=26022, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 26022, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f380a4bf000

close(3) = 0

openat(AT\_FDCWD, "/usr/local/lib64/libstdc++.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\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2530008, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 2543808, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f380a251000

mmap(0x7f380a2f6000, 1216512, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa5000) = 0x7f380a2f6000

mmap(0x7f380a41f000, 581632, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ce000) = 0x7f380a41f000

mmap(0x7f380a4ad000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25c000) = 0x7f380a4ad000

mmap(0x7f380a4bb000, 12480, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f380a4bb000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.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\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=907784, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 909560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f380a172000

mmap(0x7f380a182000, 471040, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x10000) = 0x7f380a182000

mmap(0x7f380a1f5000, 368640, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x83000) = 0x7f380a1f5000

mmap(0x7f380a24f000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xdc000) = 0x7f380a24f000

close(3) = 0

openat(AT\_FDCWD, "/usr/local/lib64/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=906528, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 181160, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f380a145000

mmap(0x7f380a149000, 143360, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7f380a149000

mmap(0x7f380a16c000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x27000) = 0x7f380a16c000

mmap(0x7f380a170000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2b000) = 0x7f380a170000

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\20t\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

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=1922136, ...}, 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, 1970000, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3809f64000

mmap(0x7f3809f8a000, 1396736, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x26000) = 0x7f3809f8a000

mmap(0x7f380a0df000, 339968, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17b000) = 0x7f380a0df000

mmap(0x7f380a132000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ce000) = 0x7f380a132000

mmap(0x7f380a138000, 53072, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f380a138000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3809f62000

arch\_prctl(ARCH\_SET\_FS, 0x7f3809f63480) = 0

set\_tid\_address(0x7f3809f63750) = 44944

set\_robust\_list(0x7f3809f63760, 24) = 0

rseq(0x7f3809f63da0, 0x20, 0, 0x53053053) = 0

mprotect(0x7f380a132000, 16384, PROT\_READ) = 0

mprotect(0x7f380a170000, 4096, PROT\_READ) = 0

mprotect(0x7f380a24f000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3809f60000

mprotect(0x7f380a4ad000, 45056, PROT\_READ) = 0

mprotect(0x403000, 4096, PROT\_READ) = 0

mprotect(0x7f380a4f8000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x7f380a4bf000, 26022) = 0

futex(0x7f380a4bb73c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

getrandom("\x9a\xad\xf9\x23\x01\x84\xba\xbd", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x1c47000

brk(0x1c68000) = 0x1c68000

newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

write(1, "Enter file name for child1: ", 28Enter file name for child1: ) = 28

newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0

read(0, pop1

"pop1\n", 1024) = 5

write(1, "Enter file name for child2: ", 28Enter file name for child2: ) = 28

read(0, pop2

"pop2\n", 1024) = 5

write(1, "All good\n", 9All good

) = 9

**openat(AT\_FDCWD, "/dev/shm/OS1", O\_RDWR|O\_CREAT|O\_NOFOLLOW|O\_CLOEXEC, 0666) = 3**

**ftruncate(3, 4096) = 0**

**mmap(NULL, 4096, PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f380a4c5000**

**openat(AT\_FDCWD, "/dev/shm/OS2", O\_RDWR|O\_CREAT|O\_NOFOLLOW|O\_CLOEXEC, 0666) = 4**

**ftruncate(4, 4096) = 0**

**mmap(NULL, 4096, PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7f380a4c4000**

getrandom("\x65\xf7\x67\x87\x0b\x2e\x40\x33", 8, GRND\_NONBLOCK) = 8

newfstatat(AT\_FDCWD, "/dev/shm/sem.bUgla0", 0x7ffd36d2c3e0, AT\_SYMLINK\_NOFOLLOW) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/dev/shm/sem.bUgla0", O\_RDWR|O\_CREAT|O\_EXCL, 0666) = 5

write(5, "\0\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", 32) = 32

mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 5, 0) = 0x7f380a4c3000

link("/dev/shm/sem.bUgla0", "/dev/shm/sem.sem1") = 0

newfstatat(5, "", {st\_mode=S\_IFREG|0644, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

unlink("/dev/shm/sem.bUgla0") = 0

close(5) = 0

getrandom("\xae\x54\x5a\x07\x83\x17\x67\xcc", 8, GRND\_NONBLOCK) = 8

newfstatat(AT\_FDCWD, "/dev/shm/sem.Iowi94", 0x7ffd36d2c3e0, AT\_SYMLINK\_NOFOLLOW) = -1 ENOENT (No such file or directory)

openat(AT\_FDCWD, "/dev/shm/sem.Iowi94", O\_RDWR|O\_CREAT|O\_EXCL, 0666) = 5

write(5, "\0\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", 32) = 32

mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 5, 0) = 0x7f380a4c2000

link("/dev/shm/sem.Iowi94", "/dev/shm/sem.sem2") = 0

newfstatat(5, "", {st\_mode=S\_IFREG|0644, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

unlink("/dev/shm/sem.Iowi94") = 0

close(5) = 0

**clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLDstrace: Process 45042 attached**

**, child\_tidptr=0x7f3809f63750) = 45042**

[pid 45042] set\_robust\_list(0x7f3809f63760, 24 <unfinished ...>

**[pid 44944] clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD <unfinished ...>**

[pid 45042] <... set\_robust\_list resumed>) = 0

strace: Process 45043 attached

**[pid 45042] execve("./child", ["child1", "pop1", "/sem1", "OS1"], 0x7ffd36d2c958 /\* 25 vars \*/ <unfinished ...>**

**[pid 44944] <... clone resumed>, child\_tidptr=0x7f3809f63750) = 45043**

[pid 45043] set\_robust\_list(0x7f3809f63760, 24 <unfinished ...>

[pid 44944] (1, "Enter linens. To finish entering"..., 52 <unfinished ...>

[pid 45043] <... set\_robust\_list resumed>) = 0

Enter linens. To finish entering lines, enter '!q'.

[pid 44944] <... write resumed>) = 52

**[pid 45043] execve("./child", ["child2", "pop2", "/sem2", "OS2"], 0x7ffd36d2c958 /\* 25 vars \*/ <unfinished ...>**

[pid 44944] read(0, <unfinished ...>

[pid 45042] <... execve resumed>) = 0

[pid 45043] <... execve resumed>) = 0

[pid 45042] brk(NULL <unfinished ...>

[pid 45043] brk(NULL) = 0x1d06000

[pid 45042] <... brk resumed>) = 0x9b6000

[pid 45042] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 45043] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 45042] <... mmap resumed>) = 0x7f17f3bc1000

[pid 45043] <... mmap resumed>) = 0x7fb4683be000

[pid 45042] access("/etc/ld.so.preload", R\_OK <unfinished ...>

[pid 45043] access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

[pid 45042] <... access resumed>) = -1 ENOENT (No such file or directory)

[pid 45042] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 45043] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 45042] <... openat resumed>) = 3

[pid 45043] <... openat resumed>) = 3

[pid 45042] newfstatat(3, "", <unfinished ...>

[pid 45043] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=26022, ...}, AT\_EMPTY\_PATH) = 0

[pid 45042] <... newfstatat resumed>{st\_mode=S\_IFREG|0644, st\_size=26022, ...}, AT\_EMPTY\_PATH) = 0

[pid 45043] mmap(NULL, 26022, PROT\_READ, MAP\_PRIVATE, 3, 0 <unfinished ...>

[pid 45042] mmap(NULL, 26022, PROT\_READ, MAP\_PRIVATE, 3, 0 <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb4683b7000

[pid 45042] <... mmap resumed>) = 0x7f17f3bba000

[pid 45043] close(3) = 0

[pid 45042] close(3 <unfinished ...>

[pid 45043] openat(AT\_FDCWD, "/usr/local/lib64/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3

[pid 45042] <... close resumed>) = 0

[pid 45043] read(3, <unfinished ...>

[pid 45042] openat(AT\_FDCWD, "/usr/local/lib64/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 45043] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 45042] <... openat resumed>) = 3

[pid 45043] newfstatat(3, "", <unfinished ...>

[pid 45042] read(3, <unfinished ...>

[pid 45043] <... newfstatat resumed>{st\_mode=S\_IFREG|0755, st\_size=2530008, ...}, AT\_EMPTY\_PATH) = 0

[pid 45042] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 45043] mmap(NULL, 2543808, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0 <unfinished ...>

[pid 45042] newfstatat(3, "", <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb468149000

[pid 45042] <... newfstatat resumed>{st\_mode=S\_IFREG|0755, st\_size=2530008, ...}, AT\_EMPTY\_PATH) = 0

[pid 45043] mmap(0x7fb4681ee000, 1216512, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa5000 <unfinished ...>

[pid 45042] mmap(NULL, 2543808, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0 <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb4681ee000

[pid 45043] mmap(0x7fb468317000, 581632, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ce000 <unfinished ...>

[pid 45042] <... mmap resumed>) = 0x7f17f394c000

[pid 45042] mmap(0x7f17f39f1000, 1216512, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa5000 <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb468317000

[pid 45042] <... mmap resumed>) = 0x7f17f39f1000

[pid 45043] mmap(0x7fb4683a5000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25c000 <unfinished ...>

[pid 45042] mmap(0x7f17f3b1a000, 581632, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ce000 <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb4683a5000

[pid 45042] <... mmap resumed>) = 0x7f17f3b1a000

[pid 45043] mmap(0x7fb4683b3000, 12480, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 45042] mmap(0x7f17f3ba8000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25c000 <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb4683b3000

[pid 45042] <... mmap resumed>) = 0x7f17f3ba8000

[pid 45043] close(3 <unfinished ...>

[pid 45042] mmap(0x7f17f3bb6000, 12480, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 45043] <... close resumed>) = 0

[pid 45042] <... mmap resumed>) = 0x7f17f3bb6000

[pid 45043] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 45042] close(3 <unfinished ...>

[pid 45043] <... openat resumed>) = 3

[pid 45042] <... close resumed>) = 0

[pid 45043] read(3, <unfinished ...>

[pid 45042] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 45043] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 45042] <... openat resumed>) = 3

[pid 45043] newfstatat(3, "", <unfinished ...>

[pid 45042] read(3, <unfinished ...>

[pid 45043] <... newfstatat resumed>{st\_mode=S\_IFREG|0644, st\_size=907784, ...}, AT\_EMPTY\_PATH) = 0

[pid 45042] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 45043] mmap(NULL, 909560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0 <unfinished ...>

[pid 45042] newfstatat(3, "", <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb46806a000

[pid 45042] <... newfstatat resumed>{st\_mode=S\_IFREG|0644, st\_size=907784, ...}, AT\_EMPTY\_PATH) = 0

[pid 45043] mmap(0x7fb46807a000, 471040, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x10000 <unfinished ...>

[pid 45042] mmap(NULL, 909560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0 <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb46807a000

[pid 45042] <... mmap resumed>) = 0x7f17f386d000

[pid 45043] mmap(0x7fb4680ed000, 368640, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x83000 <unfinished ...>

[pid 45042] mmap(0x7f17f387d000, 471040, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x10000 <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb4680ed000

[pid 45042] <... mmap resumed>) = 0x7f17f387d000

[pid 45043] mmap(0x7fb468147000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xdc000 <unfinished ...>

[pid 45042] mmap(0x7f17f38f0000, 368640, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x83000 <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb468147000

[pid 45042] <... mmap resumed>) = 0x7f17f38f0000

[pid 45043] close(3 <unfinished ...>

[pid 45042] mmap(0x7f17f394a000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xdc000 <unfinished ...>

[pid 45043] <... close resumed>) = 0

[pid 45042] <... mmap resumed>) = 0x7f17f394a000

[pid 45043] openat(AT\_FDCWD, "/usr/local/lib64/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 45042] close(3 <unfinished ...>

[pid 45043] <... openat resumed>) = 3

[pid 45042] <... close resumed>) = 0

[pid 45043] read(3, <unfinished ...>

[pid 45042] openat(AT\_FDCWD, "/usr/local/lib64/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 45043] <... read resumed>"\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 45042] <... openat resumed>) = 3

[pid 45043] newfstatat(3, "", <unfinished ...>

[pid 45042] read(3, <unfinished ...>

[pid 45043] <... newfstatat resumed>{st\_mode=S\_IFREG|0644, st\_size=906528, ...}, AT\_EMPTY\_PATH) = 0

[pid 45042] <... read resumed>"\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 45043] mmap(NULL, 181160, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fb46803d000

[pid 45042] newfstatat(3, "", <unfinished ...>

[pid 45043] mmap(0x7fb468041000, 143360, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7fb468041000

[pid 45042] <... newfstatat resumed>{st\_mode=S\_IFREG|0644, st\_size=906528, ...}, AT\_EMPTY\_PATH) = 0

[pid 45043] mmap(0x7fb468064000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x27000) = 0x7fb468064000

[pid 45042] mmap(NULL, 181160, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0 <unfinished ...>

[pid 45043] mmap(0x7fb468068000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2b000) = 0x7fb468068000

[pid 45042] <... mmap resumed>) = 0x7f17f3840000

[pid 45043] close(3 <unfinished ...>

[pid 45042] mmap(0x7f17f3844000, 143360, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000 <unfinished ...>

[pid 45043] <... close resumed>) = 0

[pid 45042] <... mmap resumed>) = 0x7f17f3844000

[pid 45043] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 45042] mmap(0x7f17f3867000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x27000 <unfinished ...>

[pid 45043] <... openat resumed>) = 3

[pid 45043] read(3, <unfinished ...>

[pid 45042] <... mmap resumed>) = 0x7f17f3867000

[pid 45043] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20t\2\0\0\0\0\0"..., 832) = 832

[pid 45042] mmap(0x7f17f386b000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2b000 <unfinished ...>

[pid 45043] pread64(3, <unfinished ...>

[pid 45042] <... mmap resumed>) = 0x7f17f386b000

[pid 45043] <... 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 45042] close(3 <unfinished ...>

[pid 45043] newfstatat(3, "", <unfinished ...>

[pid 45042] <... close resumed>) = 0

[pid 45043] <... newfstatat resumed>{st\_mode=S\_IFREG|0755, st\_size=1922136, ...}, AT\_EMPTY\_PATH) = 0

[pid 45042] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 45043] pread64(3, <unfinished ...>

[pid 45042] <... openat resumed>) = 3

[pid 45043] <... 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 45042] read(3, <unfinished ...>

[pid 45043] mmap(NULL, 1970000, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0 <unfinished ...>

[pid 45042] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20t\2\0\0\0\0\0"..., 832) = 832

[pid 45043] <... mmap resumed>) = 0x7fb467e5c000

[pid 45042] pread64(3, <unfinished ...>

[pid 45043] mmap(0x7fb467e82000, 1396736, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x26000 <unfinished ...>

[pid 45042] <... 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 45043] <... mmap resumed>) = 0x7fb467e82000

[pid 45042] newfstatat(3, "", <unfinished ...>

[pid 45043] mmap(0x7fb467fd7000, 339968, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17b000 <unfinished ...>

[pid 45042] <... newfstatat resumed>{st\_mode=S\_IFREG|0755, st\_size=1922136, ...}, AT\_EMPTY\_PATH) = 0

[pid 45043] <... mmap resumed>) = 0x7fb467fd7000

[pid 45042] pread64(3, <unfinished ...>

[pid 45043] mmap(0x7fb46802a000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ce000 <unfinished ...>

[pid 45042] <... 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 45043] <... mmap resumed>) = 0x7fb46802a000

[pid 45043] mmap(0x7fb468030000, 53072, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 45042] mmap(NULL, 1970000, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0 <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb468030000

[pid 45043] close(3 <unfinished ...>

[pid 45042] <... mmap resumed>) = 0x7f17f365f000

[pid 45043] <... close resumed>) = 0

[pid 45042] mmap(0x7f17f3685000, 1396736, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x26000 <unfinished ...>

[pid 45043] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 45042] <... mmap resumed>) = 0x7f17f3685000

[pid 45043] <... mmap resumed>) = 0x7fb467e5a000

[pid 45042] mmap(0x7f17f37da000, 339968, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17b000 <unfinished ...>

[pid 45043] arch\_prctl(ARCH\_SET\_FS, 0x7fb467e5b480 <unfinished ...>

[pid 45042] <... mmap resumed>) = 0x7f17f37da000

[pid 45043] <... arch\_prctl resumed>) = 0

[pid 45042] mmap(0x7f17f382d000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ce000 <unfinished ...>

[pid 45043] set\_tid\_address(0x7fb467e5b750 <unfinished ...>

[pid 45042] <... mmap resumed>) = 0x7f17f382d000

[pid 45043] <... set\_tid\_address resumed>) = 45043

[pid 45042] mmap(0x7f17f3833000, 53072, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f17f3833000

[pid 45043] set\_robust\_list(0x7fb467e5b760, 24 <unfinished ...>

[pid 45042] close(3) = 0

[pid 45043] <... set\_robust\_list resumed>) = 0

[pid 45042] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 45043] rseq(0x7fb467e5bda0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 45042] <... mmap resumed>) = 0x7f17f365d000

[pid 45043] <... rseq resumed>) = 0

[pid 45042] arch\_prctl(ARCH\_SET\_FS, 0x7f17f365e480) = 0

[pid 45042] set\_tid\_address(0x7f17f365e750) = 45042

[pid 45043] mprotect(0x7fb46802a000, 16384, PROT\_READ <unfinished ...>

[pid 45042] set\_robust\_list(0x7f17f365e760, 24 <unfinished ...>

[pid 45043] <... mprotect resumed>) = 0

[pid 45043] mprotect(0x7fb468068000, 4096, PROT\_READ <unfinished ...>

[pid 45042] <... set\_robust\_list resumed>) = 0

[pid 45043] <... mprotect resumed>) = 0

[pid 45043] mprotect(0x7fb468147000, 4096, PROT\_READ <unfinished ...>

[pid 45042] rseq(0x7f17f365eda0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 45043] <... mprotect resumed>) = 0

[pid 45042] <... rseq resumed>) = 0

[pid 45042] mprotect(0x7f17f382d000, 16384, PROT\_READ) = 0

[pid 45043] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 45042] mprotect(0x7f17f386b000, 4096, PROT\_READ <unfinished ...>

[pid 45043] <... mmap resumed>) = 0x7fb467e58000

[pid 45042] <... mprotect resumed>) = 0

[pid 45042] mprotect(0x7f17f394a000, 4096, PROT\_READ <unfinished ...>

[pid 45043] mprotect(0x7fb4683a5000, 45056, PROT\_READ <unfinished ...>

[pid 45042] <... mprotect resumed>) = 0

[pid 45043] <... mprotect resumed>) = 0

[pid 45043] mprotect(0x404000, 4096, PROT\_READ <unfinished ...>

[pid 45042] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 45043] <... mprotect resumed>) = 0

[pid 45042] <... mmap resumed>) = 0x7f17f365b000

[pid 45043] mprotect(0x7fb4683f0000, 8192, PROT\_READ) = 0

[pid 45042] mprotect(0x7f17f3ba8000, 45056, PROT\_READ) = 0

[pid 45043] prlimit64(0, RLIMIT\_STACK, NULL, <unfinished ...>

[pid 45042] mprotect(0x404000, 4096, PROT\_READ <unfinished ...>

[pid 45043] <... prlimit64 resumed>{rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

[pid 45042] <... mprotect resumed>) = 0

[pid 45043] munmap(0x7fb4683b7000, 26022 <unfinished ...>

[pid 45042] mprotect(0x7f17f3bf3000, 8192, PROT\_READ <unfinished ...>

[pid 45043] <... munmap resumed>) = 0

[pid 45042] <... mprotect resumed>) = 0

[pid 45043] **futex**(0x7fb4683b373c, FUTEX\_WAKE\_PRIVATE, 2147483647 <unfinished ...>

[pid 45042] prlimit64(0, RLIMIT\_STACK, NULL, <unfinished ...>

[pid 45043] <... futex resumed>) = 0

[pid 45042] <... prlimit64 resumed>{rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

[pid 45042] munmap(0x7f17f3bba000, 26022) = 0

[pid 45043] getrandom("\xe3\xe4\xac\x01\x3c\xc7\x65\xff", 8, GRND\_NONBLOCK) = 8

[pid 45042] **futex**(0x7f17f3bb673c, FUTEX\_WAKE\_PRIVATE, 2147483647 <unfinished ...>

[pid 45043] brk(NULL <unfinished ...>

[pid 45042] <... futex resumed>) = 0

[pid 45043] <... brk resumed>) = 0x1d06000

[pid 45043] brk(0x1d27000 <unfinished ...>

[pid 45042] getrandom( <unfinished ...>

[pid 45043] <... brk resumed>) = 0x1d27000

[pid 45042] <... getrandom resumed>"\x4d\x43\x20\xfa\x95\x8b\x97\x2b", 8, GRND\_NONBLOCK) = 8

[pid 45043] **openat(AT\_FDCWD, "pop2", O\_RDWR|O\_CREAT|O\_TRUNC, 0167140 <unfinished ...>**

[pid 45042] brk(NULL) = 0x9b6000

[pid 45042] brk(0x9d7000) = 0x9d7000

[**pid 45042] openat(AT\_FDCWD, "pop1", O\_RDWR|O\_CREAT|O\_TRUNC, 077500 <unfinished ...>**

[pid 45043] <... openat resumed>) = 3

**[pid 45043] openat(AT\_FDCWD, "/dev/shm/sem.sem2", O\_RDWR|O\_NOFOLLOW) = 4**

[pid 45043] newfstatat(4, "", {st\_mode=S\_IFREG|0644, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

**[pid 45043] mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7fb4683bd000**

[pid 45043] close(4) = 0

**[pid 45043] openat(AT\_FDCWD, "/dev/shm/OS2", O\_RDWR|O\_NOFOLLOW|O\_CLOEXEC) = 4**

**[pid 45043] mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7fb4683bc000**

[pid 45042] <... openat resumed>) = 3

[pid 45043] **futex**(0x7fb4683bd000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY <unfinished ...>

[pid 45042] openat(AT\_FDCWD, "/dev/shm/sem.sem1", O\_RDWR|O\_NOFOLLOW) = 4

[pid 45042] newfstatat(4, "", {st\_mode=S\_IFREG|0644, st\_size=32, ...}, AT\_EMPTY\_PATH) = 0

[pid 45042] mmap(NULL, 32, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7f17f3bc0000

[pid 45042] close(4) = 0

**[pid 45042] openat(AT\_FDCWD, "/dev/shm/OS1", O\_RDWR|O\_NOFOLLOW|O\_CLOEXEC) = 4**

**[pid 45042] mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 4, 0) = 0x7f17f3bbf000**

[pid 45042] **futex**(0x7f17f3bc0000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANYHello

<unfinished ...>

[pid 44944] <... read resumed>"Hello\n", 1024) = 6

[pid 44944] **futex**(0x7f380a4c2000, FUTEX\_WAKE, 1) = 1

[pid 45043] <... futex resumed>) = 0

[pid 44944] clock\_nanosleep(CLOCK\_REALTIME, 0, {tv\_sec=0, tv\_nsec=1000000}, <unfinished ...>

[pid 45043] write(3, "Hll\n", 4) = 4

[pid 44944] <... clock\_nanosleep resumed>NULL) = 0

[pid 45043] clock\_nanosleep(CLOCK\_REALTIME, 0, {tv\_sec=0, tv\_nsec=1000000}, <unfinished ...>

[pid 44944] read(0, <unfinished ...>

[pid 45043] <... clock\_nanosleep resumed>NULL) = 0

[pid 45043] **futex**(0x7fb4683bd000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANYWorld

<unfinished ...>

[pid 44944] <... read resumed>"World\n", 1024) = 6

[pid 44944] futex(0x7f380a4c2000, FUTEX\_WAKE, 1) = 1

[pid 45043] <... futex resumed>) = 0

[pid 44944] clock\_nanosleep(CLOCK\_REALTIME, 0, {tv\_sec=0, tv\_nsec=1000000}, <unfinished ...>

[pid 45043] write(3, "Wrld\n", 5) = 5

[pid 44944] <... clock\_nanosleep resumed>NULL) = 0

[pid 44944] read(0, <unfinished ...>

[pid 45043] clock\_nanosleep(CLOCK\_REALTIME, 0, {tv\_sec=0, tv\_nsec=1000000}, NULL) = 0

[pid 45043] **futex**(0x7fb4683bd000, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 0, NULL, FUTEX\_BITSET\_MATCH\_ANY!q

<unfinished ...>

[pid 44944] <... read resumed>"!q\n", 1024) = 3

[pid 44944] **futex**(0x7f380a4c3000, FUTEX\_WAKE, 1) = 1

[pid 45042] <... futex resumed>) = 0

[pid 44944] clock\_nanosleep(CLOCK\_REALTIME, 0, {tv\_sec=0, tv\_nsec=1000000}, <unfinished ...>

[pid 45042] **munmap(0x7f17f3bbf000, 4096) = 0**

[pid 45042] close(4) = 0

[pid 45042] **munmap(0x7f17f3bc0000, 32) = 0**

[pid 44944] <... clock\_nanosleep resumed>NULL) = 0

[pid 45042] **unlink("/dev/shm/sem.sem1" <unfinished ...>**

[pid 44944] **futex**(0x7f380a4c2000, FUTEX\_WAKE, 1 <unfinished ...>

[pid 45042] <... unlink resumed>) = 0

[pid 44944] <... futex resumed>) = 1

[pid 45043] <... futex resumed>) = 0

[pid 44944] clock\_nanosleep(CLOCK\_REALTIME, 0, {tv\_sec=0, tv\_nsec=1000000}, <unfinished ...>

[pid 45042] close(3 <unfinished ...>

[pid 45043] **munmap(0x7fb4683bc000, 4096) = 0**

[pid 45043] close(4) = 0

[pid 45043] **munmap(0x7fb4683bd000, 32) = 0**

[pid 45043] **unlink("/dev/shm/sem.sem2" <unfinished ...>**

[pid 44944] <... clock\_nanosleep resumed>NULL) = 0

[pid 45043] <... unlink resumed>) = 0

[pid 44944] **munmap(0x7f380a4c5000, 4096) = 0**

[pid 45043] close(3 <unfinished ...>

[pid 44944] close(3 <unfinished ...>

[pid 45042] <... close resumed>) = 0

[pid 44944] <... close resumed>) = 0

[pid 44944] **unlink("/dev/shm/OS1" <unfinished ...>**

[pid 45042] exit\_group(0) = ?

[pid 44944] <... unlink resumed>) = 0

[pid 44944] **munmap(0x7f380a4c4000, 4096) = 0**

[pid 44944] close(4) = 0

[pid 44944] **unlink("/dev/shm/OS2") = 0**

[pid 44944] **munmap(0x7f380a4c3000, 32) = 0**

[pid 45042] +++ exited with 0 +++

[pid 44944] --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=45042, si\_uid=0, si\_status=0, si\_utime=0, si\_stime=2 /\* 0.02 s \*/} ---

[pid 45043] <... close resumed>) = 0

[pid 44944] **munmap(0x7f380a4c2000, 32) = 0**

[pid 45043] exit\_group(0 <unfinished ...>

[pid 44944] **unlink("/dev/shm/sem.sem1" <unfinished ...>**

[pid 45043] <... exit\_group resumed>) = ?

[pid 44944] <... unlink resumed>) = -1 ENOENT (No such file or directory)

[pid 44944] unlink("/dev/shm/sem.sem2") = -1 ENOENT (No such file or directory)

[pid 44944] lseek(0, -1, SEEK\_CUR) = -1 ESPIPE (Illegal seek)

[pid 44944] exit\_group(0) = ?

[pid 45043] +++ exited with 0 +++

+++ exited with 0 +++

**Вывод**

Лабораторная работа оказалась очень интересной и познавательной. Было интересно сравнить разные подходы к передаче данных между процессами, а именно memory mapping и pipe. Оба способа эффективны по-своему и нужно с умом применять их. Так же было интересно реализовать синхронизацию процессов с помощью семафора.