Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №1 по курсу**

**«Операционные системы»**

Группа: М8О-215Б-23

Студент: Самарский Я.В

Преподаватель: Миронов Е.С.

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

Дата: 31.10.24

Москва, 2024

**Постановка задачи**

**Вариант 9.**

Родительский процесс создает дочерний процесс. Первой строчкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия файла с таким именем на чтение. Стандартный поток ввода дочернего процесса переопределяется открытым файлом. Дочерний процесс читает команды из стандартного потока ввода. Стандартный поток вывода дочернего процесса перенаправляется в pipe1. Родительский процесс читает из pipe1 и прочитанное выводит в свой стандартный поток вывода. Родительский и дочерний процесс должны быть представлены разными программами.

В файле записаны команды вида:«число число число». Дочерний процесс производит деление первого числа команда, на последующие числа в команде, а результат выводит в стандартный поток вывода. Если происходит деление на 0, то тогда дочерний и родительский процесс завершают свою работу. Проверка деления на 0 должна осуществляться на стороне дочернего процесса. Числа имеют тип float. Количество чисел может быть произвольным.

**Общий метод и алгоритм решения**

Использованные системные вызовы:

* pid\_t fork(void); – создает дочерний процесс.
* execl(const char \*path, const char \*arg, ...) – замена памяти процесса
* pid\_t waitpid(pid\_t pid, int \*stat\_loc, int options) - ожидание завершения дочернего процесса
* int pipe(int pipefd[2]) - создание неименованного канала для передачи данных между процессами
* int dup2(int oldfd, int newfd) - переназначение файлового дескриптора
* int open(const char \*pathname, int flags, mode\_t mode) - открытие\создание файла
* int close(int fd) - закрыть файл
* size\_t read (int fd, void\* buf, size\_t cnt) – чтение из файла
* size\_t write (int fd, void\* buf, size\_t cnt) – запись в файл

В родительском процессе считываем имя файла. Далее этот файл открываем на чтение. Создаём канал. Создаём дочерний процесс. Стандартный поток ввода дочернего процесса переопределяется открытым файлом. Стандартный поток вывода дочернего процесса перенаправляется в pipe1. Используем execl для замены памяти процесса на программу, написанную для дочернего процесса. В случае неудачи продолжит выполняться старый код, будет выведено сообщение об ошибке, и программа завершится.

В родительском процессе закрываем дескриптор канала на запись. Считываем из канала символы и выводим их пользователю. Закрываем дескриптор на чтение и ожидаем завершение дочернего процесса.

В программе дочернего процесса считываем из входного потока числа и выполняем деление. В случае деления на 0 программа завершается с кодом 1.

При создании процесса, открытии файла, создания канала, переопределения дескрипторов, чтении из файла должны обрабатываться случаи ошибок: должно выводиться сообщение об ошибке, открытые дескрипторы должны явно закрываться, программа должна завершаться.

**Код программы**

**main/main.cpp**

#include <iostream>

#include <unistd.h>

#include <fcntl.h>

#include <sys/types.h>

#include <sys/wait.h>

int main() {

int fd[2];

if (pipe(fd)) {

std::cerr << "ERROR: Pipe creation fault" << std::endl;

return 1;

}

// Первой строчкой пользователь в консоль родительского процесса вводит имя файла,

// которое будет использовано для открытия файла с таким именем на чтение

std::string name;

std::getline(std::cin, name);

const int file = open(name.c\_str(), O\_RDONLY);

// Родительский процесс создает дочерний процесс.

const pid\_t pid = fork();

if (pid == -1) {

std::cerr << "ERROR: Error while creating child process" << std::endl;

return 1;

}

// Дочерний процесс

if (pid == 0) {

// Стандартный поток ввода дочернего процесса переопределяется открытым файлом.

dup2(file, STDIN\_FILENO);

// Стандартный поток вывода дочернего процесса перенаправляется в pipe1

close(fd[0]);

dup2(fd[1], STDOUT\_FILENO);

// Родительский и дочерний процесс должны быть представлены разными программами.

execl("./child", "./child", nullptr);

std::cerr << "ERROR: Can't execute child process" << std::endl;

return 1;

}

// Родительский процесс

close(fd[1]);

// Родительский процесс читает из pipe1 и прочитанное выводит в свой стандартный поток вывода.

char a[10];

while (const ssize\_t size = read(fd[0], a, sizeof(a) / sizeof(char))) {

std::string str(a, size);

std::cout << str;

}

int status;

waitpid(pid, &status, 0);

if (WIFEXITED(status) && WEXITSTATUS(status) == 1) {

std::cerr << "ERROR: Division by zero" << std::endl;

return 1;

}

close(fd[0]);

return 0;

}

**child/main.cpp**

#include <iostream>

int main() {

while (true) {

float result;

if (!(std::cin >> result))

break;

while (std::cin.peek() != '\n') {

float divider;

std::cin >> divider;

if (divider == 0)

return 1;

result /= divider;

}

std::cout << result << std::endl;

}

return 0;

}

**Протокол работы программы**

**Тест 1:**

$ ./mai\_os

/home/user/projects/mai\_os/input.txt

2

2

5

2

$ cat < /home/user/projects/mai\_os/input.txt

4 2

5 2.5

5 1 1 1 1

12 3 2

**Тест 2:**

$ ./mai\_os

/home/user/projects/mai\_os/input.txt

2

7

32

3.5

$ cat < ../input.txt

21 3 3.5

7 1

16 0.5

14 2 2

**Тест 3:**

$ ./mai\_os

/home/user/projects/mai\_os/input.txt

$ cat < ../input.txt

**Strace (Тест 1):**

$ strace -f ./main

execve("./mai\_os", ["./mai\_os"], 0x7ffd1e3ed008 /\* 32 vars \*/) = 0

brk(NULL) = 0x5654e0740000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fffffcede40) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fcaa43b7000

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=18463, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 18463, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fcaa43b2000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/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|0644, st\_size=2260296, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fcaa4186000

mprotect(0x7fcaa4220000, 1576960, PROT\_NONE) = 0

mmap(0x7fcaa4220000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7fcaa4220000

mmap(0x7fcaa4331000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7fcaa4331000

mmap(0x7fcaa43a1000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7fcaa43a1000

mmap(0x7fcaa43af000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fcaa43af000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/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=125488, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 127720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fcaa4166000

mmap(0x7fcaa4169000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fcaa4169000

mmap(0x7fcaa4180000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7fcaa4180000

mmap(0x7fcaa4184000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7fcaa4184000

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) = 0x7fcaa3f3d000

mprotect(0x7fcaa3f65000, 2023424, PROT\_NONE) = 0

mmap(0x7fcaa3f65000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7fcaa3f65000

mmap(0x7fcaa40fa000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7fcaa40fa000

mmap(0x7fcaa4153000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x215000) = 0x7fcaa4153000

mmap(0x7fcaa4159000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fcaa4159000

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=940560, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 942344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fcaa3e56000

mmap(0x7fcaa3e64000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7fcaa3e64000

mmap(0x7fcaa3ee0000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7fcaa3ee0000

mmap(0x7fcaa3f3b000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7fcaa3f3b000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fcaa3e54000

arch\_prctl(ARCH\_SET\_FS, 0x7fcaa3e553c0) = 0

set\_tid\_address(0x7fcaa3e55690) = 25126

set\_robust\_list(0x7fcaa3e556a0, 24) = 0

rseq(0x7fcaa3e55d60, 0x20, 0, 0x53053053) = 0

mprotect(0x7fcaa4153000, 16384, PROT\_READ) = 0

mprotect(0x7fcaa3f3b000, 4096, PROT\_READ) = 0

mprotect(0x7fcaa4184000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fcaa3e52000

mprotect(0x7fcaa43a1000, 45056, PROT\_READ) = 0

mprotect(0x5654dec86000, 4096, PROT\_READ) = 0

mprotect(0x7fcaa43f1000, 8192, PROT\_READ) = 0

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

munmap(0x7fcaa43b2000, 18463) = 0

getrandom("\xa9\x70\xf3\x5a\x63\x71\x14\x59", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x5654e0740000

brk(0x5654e0761000) = 0x5654e0761000

futex(0x7fcaa43af77c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

**pipe2([3, 4], 0) = 0**

newfstatat(0, "", {st\_mode=S\_IFIFO|0600, st\_size=0, ...}, AT\_EMPTY\_PATH) = 0

**read(0, "/home/user/projects/mai\_os/input"..., 4096) = 38**

openat(AT\_FDCWD, "/home/user/projects/mai\_os/input.txt", O\_RDONLY) = 5

**clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLDstrace: Process 25127 attached**

**, child\_tidptr=0x7fcaa3e55690) = 25127**

[pid 25127] set\_robust\_list(0x7fcaa3e556a0, 24 <unfinished ...>

**[pid 25126] close(4 <unfinished ...>**

[pid 25127] <... set\_robust\_list resumed>) = 0

[pid 25126] <... close resumed>) = 0

**[pid 25127] dup2(5, 0 <unfinished ...>**

**[pid 25126] read(3, <unfinished ...>**

[pid 25127] <... dup2 resumed>) = 0

**[pid 25127] close(3) = 0**

**[pid 25127] dup2(4, 1) = 1**

**[pid 25127] execve("./child", ["./child"], 0x7fffffcee018 /\* 32 vars \*/) = 0**

[pid 25127] brk(NULL) = 0x55f1ca5b7000

[pid 25127] arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffd5690f880) = -1 EINVAL (Invalid argument)

[pid 25127] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f1ff5ee1000

[pid 25127] access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

[pid 25127] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

[pid 25127] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=18463, ...}, AT\_EMPTY\_PATH) = 0

[pid 25127] mmap(NULL, 18463, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f1ff5edc000

**[pid 25127] close(3) = 0**

[pid 25127] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3

**[pid 25127] 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**

[pid 25127] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2260296, ...}, AT\_EMPTY\_PATH) = 0

[pid 25127] mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f1ff5cb0000

[pid 25127] mprotect(0x7f1ff5d4a000, 1576960, PROT\_NONE) = 0

[pid 25127] mmap(0x7f1ff5d4a000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7f1ff5d4a000

[pid 25127] mmap(0x7f1ff5e5b000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7f1ff5e5b000

[pid 25127] mmap(0x7f1ff5ecb000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7f1ff5ecb000

[pid 25127] mmap(0x7f1ff5ed9000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f1ff5ed9000

**[pid 25127] close(3) = 0**

[pid 25127] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

**[pid 25127] 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**

[pid 25127] 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

[pid 25127] 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

[pid 25127] 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

[pid 25127] newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2220400, ...}, AT\_EMPTY\_PATH) = 0

[pid 25127] 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

[pid 25127] mmap(NULL, 2264656, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f1ff5a87000

[pid 25127] mprotect(0x7f1ff5aaf000, 2023424, PROT\_NONE) = 0

[pid 25127] mmap(0x7f1ff5aaf000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f1ff5aaf000

[pid 25127] mmap(0x7f1ff5c44000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7f1ff5c44000

[pid 25127] mmap(0x7f1ff5c9d000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x215000) = 0x7f1ff5c9d000

[pid 25127] mmap(0x7f1ff5ca3000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f1ff5ca3000

**[pid 25127] close(3) = 0**

[pid 25127] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 3

**[pid 25127] 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**

[pid 25127] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=940560, ...}, AT\_EMPTY\_PATH) = 0

[pid 25127] mmap(NULL, 942344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f1ff59a0000

[pid 25127] mmap(0x7f1ff59ae000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f1ff59ae000

[pid 25127] mmap(0x7f1ff5a2a000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7f1ff5a2a000

[pid 25127] mmap(0x7f1ff5a85000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7f1ff5a85000

**[pid 25127] close(3) = 0**

[pid 25127] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

**[pid 25127] 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**

[pid 25127] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=125488, ...}, AT\_EMPTY\_PATH) = 0

[pid 25127] mmap(NULL, 127720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f1ff5980000

[pid 25127] mmap(0x7f1ff5983000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f1ff5983000

[pid 25127] mmap(0x7f1ff599a000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7f1ff599a000

[pid 25127] mmap(0x7f1ff599e000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7f1ff599e000

**[pid 25127] close(3) = 0**

[pid 25127] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f1ff597e000

[pid 25127] arch\_prctl(ARCH\_SET\_FS, 0x7f1ff597f3c0) = 0

[pid 25127] set\_tid\_address(0x7f1ff597f690) = 25127

[pid 25127] set\_robust\_list(0x7f1ff597f6a0, 24) = 0

[pid 25127] rseq(0x7f1ff597fd60, 0x20, 0, 0x53053053) = 0

[pid 25127] mprotect(0x7f1ff5c9d000, 16384, PROT\_READ) = 0

[pid 25127] mprotect(0x7f1ff599e000, 4096, PROT\_READ) = 0

[pid 25127] mprotect(0x7f1ff5a85000, 4096, PROT\_READ) = 0

[pid 25127] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f1ff597c000

[pid 25127] mprotect(0x7f1ff5ecb000, 45056, PROT\_READ) = 0

[pid 25127] mprotect(0x55f1ca4cc000, 4096, PROT\_READ) = 0

[pid 25127] mprotect(0x7f1ff5f1b000, 8192, PROT\_READ) = 0

[pid 25127] prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

[pid 25127] munmap(0x7f1ff5edc000, 18463) = 0

[pid 25127] getrandom("\xa9\xca\x8a\x30\x9e\x44\x30\xbb", 8, GRND\_NONBLOCK) = 8

[pid 25127] brk(NULL) = 0x55f1ca5b7000

[pid 25127] brk(0x55f1ca5d8000) = 0x55f1ca5d8000

[pid 25127] futex(0x7f1ff5ed977c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

[pid 25127] newfstatat(0, "", {st\_mode=S\_IFREG|0644, st\_size=27, ...}, AT\_EMPTY\_PATH) = 0

**[pid 25127] read(0, "4 2\n5 2.5\n5 1 1 1 1\n12 3 2\n", 4096) = 27**

[pid 25127] newfstatat(1, "", {st\_mode=S\_IFIFO|0600, st\_size=0, ...}, AT\_EMPTY\_PATH) = 0

**[pid 25127] write(1, "2\n", 2 <unfinished ...>**

[pid 25126] <... read resumed>"2\n", 10) = 2

[pid 25127] <... write resumed>) = 2

[pid 25126] newfstatat(1, "", <unfinished ...>

**[pid 25127] write(1, "2\n", 2 <unfinished ...>**

[pid 25126] <... newfstatat resumed>{st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x3), ...}, AT\_EMPTY\_PATH) = 0

[pid 25127] <... write resumed>) = 2

**[pid 25126] write(1, "2\n", 2 <unfinished ...>**

2

**[pid 25127] write(1, "5\n", 2 <unfinished ...>**

[pid 25126] <... write resumed>) = 2

[pid 25127] <... write resumed>) = 2

**[pid 25126] read(3, <unfinished ...>**

**[pid 25127] write(1, "2\n", 2 <unfinished ...>**

[pid 25126] <... read resumed>"2\n5\n", 10) = 4

[pid 25127] <... write resumed>) = 2

**[pid 25126] write(1, "2\n5\n", 4 <unfinished ...>**

2

5

**[pid 25127] read(0, <unfinished ...>**

[pid 25126] <... write resumed>) = 4

[pid 25127] <... read resumed>"", 4096) = 0

**[pid 25126] read(3, "2\n", 10) = 2**

**[pid 25126] write(1, "2\n", 22**

) = 2

**[pid 25126] read(3, <unfinished ...>**

[pid 25127] exit\_group(0) = ?

[pid 25126] <... read resumed>"", 10) = 0

[pid 25127] +++ exited with 0 +++

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=25127, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

**wait4(25127, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 25127**

**close(3) = 0**

lseek(0, -1, SEEK\_CUR) = -1 ESPIPE (Illegal seek)

exit\_group(0) = ?

+++ exited with 0 +++

**Вывод**

При выполнении работы познакомился с системными функциями Linux. Возникли проблемы с чтением данных из дочернего процесса. Потребовалась внимательность не запутаться в дескрипторах**.**