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: М80-206Б-20

Студент: Каримов А.А.

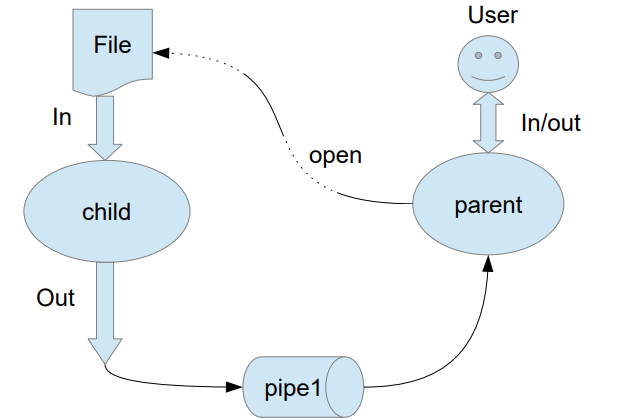
Преподаватель: Миронов Е.С.

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

Дата: 05.10.23

Москва, 2023

**Постановка задачи**

**Вариант 10.**

В файле записаны команды вида: «число». Дочерний процесс производит проверку этого числа на простоту. Если число составное, то дочерний процесс пишет это число в стандартный поток вывода. Если число отрицательное или простое, то тогда дочерний и родительский процессы завершаются. Количество чисел может быть произвольным

**Общий метод и алгоритм решения**

Использованные системные вызовы:

* pid\_t fork(void); – создает дочерний процесс
* int pipe(int \*fd); – создает pipe, через который будем передавать данные от одного процесса к другому
* read(int fd, void buf, size\_t count); - пытается прочитать count байт по файловому дескриптору fd, в буфер buf
* close(int fd); - закрывает дескриптор fd
* write(int fd, void \*buf, int count); - пишет по дескриптору fd, count байт из buf
* open(const char \*pathname, int flags); - открывает файл по filename, возвращает файл дескриптор для обращения к файлу
* execl(char \*fname, char \*fname, ..., char \*argN, NULL); - запуск другой программы для выполнения
* dup2(int oldfd, int newfd); - ссылаем новый дескриптор на старый

В лабораторной работе я написал программу, которая создает еще один процесс внутри себя. Стандартный ввод и вывод для дочернего процесса определяются на текстовый файл и на pipe. Дочерний процесс, читая числа из файла, проверяет их на простоту, если число составное, то он пишет его в pipe, иначе программа завершается и выводит составные числа в стандартный вывод.

**Код программы**

**main.c**

#include <iostream>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <cstring>

#define RED\_COLOR "\x1b[31m" // child process output color

#define GREEN\_COLOR "\x1b[32m" // parent process output color

#define RESET\_COLOR "\x1b[0m" // reset color

#define MAX\_PATH\_SIZE 64

using namespace std;

using ll = int64\_t;

int create\_process() {

pid\_t pid = fork();

if (-1 == pid)

{

perror("fork");

exit(-1);

}

return pid;

}

int main() {

char str\_file\_path[MAX\_PATH\_SIZE];

cout << "Enter file path:" << endl;

cin >> str\_file\_path;

int input\_file = open(str\_file\_path, O\_RDONLY);

if(input\_file == -1) {

perror("Can't open file");

exit(-1);

}

int pipe\_fd[2];

if(pipe(pipe\_fd) == -1) {

perror("pipe");

exit(-1);

}

pid\_t pid = create\_process();

if(pid == 0) { // Child process (writable) fd(1)

close(pipe\_fd[0]);

if(dup2(input\_file, STDIN\_FILENO) == -1) { // set txt-file as stdin

perror("dup2 can't redirect stdin to input\_file");

exit(-1);

}

if(dup2(pipe\_fd[1], STDOUT\_FILENO) == -1) { // set pipe fd as stdout

perror("dup2 can't redirect stdout to pipe");

exit(-1);

}

if(execl("./child", "./child", NULL) == -1) { // exec child process

perror("can't exec child process");

exit(-1);

}

close(pipe\_fd[1]);

} else { // Parent process (readable) fd(0)

close(pipe\_fd[1]);

close(input\_file); // close unused file

int tmp;

while(read(pipe\_fd[0], &tmp, sizeof(int)) == sizeof(int)) { // getting numbers from pipe

cout << GREEN\_COLOR << tmp << endl;

}

close(pipe\_fd[0]);

}

}

**child.cpp**

#include <iostream>

#include <unistd.h>

using namespace std;

bool IsPrime(int);

int main() {

int tmp;

while(scanf("%d", &tmp) != EOF) {

if(!IsPrime(tmp)) {

write(STDOUT\_FILENO, &tmp, sizeof(int));

} else {

break;

}

}

}

**checker.cpp**

bool IsPrime(int x) {

if(x < 0) {

return true;

} else if(x == 0 or x == 1) {

return false;

}

for(int i = 2; i \* i <= x; ++i) {

if(x % i == 0) {

return false;

}

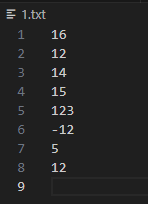
}

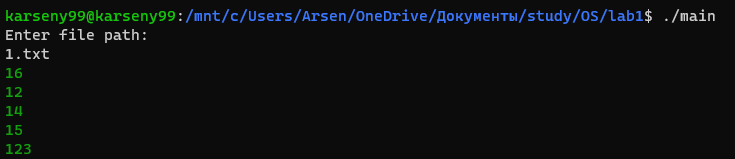
return true;

}

**Протокол работы программы**

**Тестирование:**

1. ****



1. Первое число - простое



**Strace:**

execve("./main", ["./main"], 0x7fff7442c5b8 /\* 28 vars \*/) = 0

brk(NULL) = 0x55b8fc4bb000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffe076e8750) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3d3e3bf000

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=18439, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 18439, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f3d3e3ba000

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) = 0x7f3d3e18e000

mprotect(0x7f3d3e228000, 1576960, PROT\_NONE) = 0

mmap(0x7f3d3e228000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7f3d3e228000

mmap(0x7f3d3e339000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7f3d3e339000

mmap(0x7f3d3e3a9000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7f3d3e3a9000

mmap(0x7f3d3e3b7000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3d3e3b7000

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\0\244;\374\204(\337f#\315I\214\234\f\256\271\32"..., 68, 896) = 68

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2216304, ...}, 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, 2260560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f3d3df66000

mmap(0x7f3d3df8e000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f3d3df8e000

mmap(0x7f3d3e123000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7f3d3e123000

mmap(0x7f3d3e17b000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7f3d3e17b000

mmap(0x7f3d3e181000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f3d3e181000

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) = 0x7f3d3de7f000

mmap(0x7f3d3de8d000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f3d3de8d000

mmap(0x7f3d3df09000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7f3d3df09000

mmap(0x7f3d3df64000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7f3d3df64000

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) = 0x7f3d3de5f000

mmap(0x7f3d3de62000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f3d3de62000

mmap(0x7f3d3de79000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7f3d3de79000

mmap(0x7f3d3de7d000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7f3d3de7d000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3d3de5d000

arch\_prctl(ARCH\_SET\_FS, 0x7f3d3de5e3c0) = 0

set\_tid\_address(0x7f3d3de5e690) = 2920

set\_robust\_list(0x7f3d3de5e6a0, 24) = 0

rseq(0x7f3d3de5ed60, 0x20, 0, 0x53053053) = 0

mprotect(0x7f3d3e17b000, 16384, PROT\_READ) = 0

mprotect(0x7f3d3de7d000, 4096, PROT\_READ) = 0

mprotect(0x7f3d3df64000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f3d3de5b000

mprotect(0x7f3d3e3a9000, 45056, PROT\_READ) = 0

mprotect(0x55b8fb79c000, 4096, PROT\_READ) = 0

mprotect(0x7f3d3e3f9000, 8192, PROT\_READ) = 0

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

munmap(0x7f3d3e3ba000, 18439) = 0

getrandom("\x68\x03\xad\x9e\x7d\xdd\xcd\x15", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55b8fc4bb000

brk(0x55b8fc4dc000) = 0x55b8fc4dc000

futex(0x7f3d3e3b777c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

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

**write(1, "Enter file path:\n", 17Enter file path:**

**) = 17**

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

**read(0, 1.txt**

**"1.txt\n", 1024) = 6**

openat(AT\_FDCWD, "1.txt", O\_RDONLY) = 3

**pipe2([4, 5], 0) = 0**

**clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f3d3de5e690) = 2921**

**strace: Process 2921 attached**

[pid 2920] close(5 <unfinished ...>

[pid 2921] set\_robust\_list(0x7f3d3de5e6a0, 24 <unfinished ...>

[pid 2920] <... close resumed>) = 0

[pid 2921] <... set\_robust\_list resumed>) = 0

[pid 2920] close(3 <unfinished ...>

[pid 2921] close(4 <unfinished ...>

[pid 2920] <... close resumed>) = 0

[pid 2921] <... close resumed>) = 0

[pid 2920] read(4, <unfinished ...>

**[pid 2921] dup2(3, 0) = 0**

**[pid 2921] dup2(5, 1) = 1**

**[pid 2921] execve("./child", ["./child"], 0x7ffe076e8928 /\* 28 vars \*/) = 0**

[pid 2921] brk(NULL) = 0x563541519000

[pid 2921] arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fff31ebec90) = -1 EINVAL (Invalid argument)

[pid 2921] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fab5961c000

[pid 2921] access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

[pid 2921] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 4

[pid 2921] newfstatat(4, "", {st\_mode=S\_IFREG|0644, st\_size=18439, ...}, AT\_EMPTY\_PATH) = 0

[pid 2921] mmap(NULL, 18439, PROT\_READ, MAP\_PRIVATE, 4, 0) = 0x7fab59617000

[pid 2921] close(4) = 0

[pid 2921] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 4

[pid 2921] read(4, "\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 2921] newfstatat(4, "", {st\_mode=S\_IFREG|0644, st\_size=2260296, ...}, AT\_EMPTY\_PATH) = 0

[pid 2921] mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 4, 0) = 0x7fab593eb000

[pid 2921] mprotect(0x7fab59485000, 1576960, PROT\_NONE) = 0

[pid 2921] mmap(0x7fab59485000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x9a000) = 0x7fab59485000

[pid 2921] mmap(0x7fab59596000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x1ab000) = 0x7fab59596000

[pid 2921] mmap(0x7fab59606000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x21a000) = 0x7fab59606000

[pid 2921] mmap(0x7fab59614000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fab59614000

[pid 2921] close(4) = 0

[pid 2921] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 4

[pid 2921] read(4, "\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 2921] pread64(4, "\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 2921] pread64(4, "\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 2921] pread64(4, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\244;\374\204(\337f#\315I\214\234\f\256\271\32"..., 68, 896) = 68

[pid 2921] newfstatat(4, "", {st\_mode=S\_IFREG|0755, st\_size=2216304, ...}, AT\_EMPTY\_PATH) = 0

[pid 2921] pread64(4, "\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 2921] mmap(NULL, 2260560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 4, 0) = 0x7fab591c3000

[pid 2921] mmap(0x7fab591eb000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x28000) = 0x7fab591eb000

[pid 2921] mmap(0x7fab59380000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x1bd000) = 0x7fab59380000

[pid 2921] mmap(0x7fab593d8000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x214000) = 0x7fab593d8000

[pid 2921] mmap(0x7fab593de000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fab593de000

[pid 2921] close(4) = 0

[pid 2921] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 4

[pid 2921] read(4, "\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 2921] newfstatat(4, "", {st\_mode=S\_IFREG|0644, st\_size=940560, ...}, AT\_EMPTY\_PATH) = 0

[pid 2921] mmap(NULL, 942344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 4, 0) = 0x7fab590dc000

[pid 2921] mmap(0x7fab590ea000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0xe000) = 0x7fab590ea000

[pid 2921] mmap(0x7fab59166000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x8a000) = 0x7fab59166000

[pid 2921] mmap(0x7fab591c1000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0xe4000) = 0x7fab591c1000

[pid 2921] close(4) = 0

[pid 2921] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 4

[pid 2921] read(4, "\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 2921] newfstatat(4, "", {st\_mode=S\_IFREG|0644, st\_size=125488, ...}, AT\_EMPTY\_PATH) = 0

[pid 2921] mmap(NULL, 127720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 4, 0) = 0x7fab590bc000

[pid 2921] mmap(0x7fab590bf000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x3000) = 0x7fab590bf000

[pid 2921] mmap(0x7fab590d6000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x1a000) = 0x7fab590d6000

[pid 2921] mmap(0x7fab590da000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x1d000) = 0x7fab590da000

[pid 2921] close(4) = 0

[pid 2921] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fab590ba000

[pid 2921] arch\_prctl(ARCH\_SET\_FS, 0x7fab590bb3c0) = 0

[pid 2921] set\_tid\_address(0x7fab590bb690) = 2921

[pid 2921] set\_robust\_list(0x7fab590bb6a0, 24) = 0

[pid 2921] rseq(0x7fab590bbd60, 0x20, 0, 0x53053053) = 0

[pid 2921] mprotect(0x7fab593d8000, 16384, PROT\_READ) = 0

[pid 2921] mprotect(0x7fab590da000, 4096, PROT\_READ) = 0

[pid 2921] mprotect(0x7fab591c1000, 4096, PROT\_READ) = 0

[pid 2921] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fab590b8000

[pid 2921] mprotect(0x7fab59606000, 45056, PROT\_READ) = 0

[pid 2921] mprotect(0x56353fc58000, 4096, PROT\_READ) = 0

[pid 2921] mprotect(0x7fab59656000, 8192, PROT\_READ) = 0

[pid 2921] prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

[pid 2921] munmap(0x7fab59617000, 18439) = 0

[pid 2921] getrandom("\xce\xad\x70\x96\x2c\xfe\xc7\x64", 8, GRND\_NONBLOCK) = 8

[pid 2921] brk(NULL) = 0x563541519000

[pid 2921] brk(0x56354153a000) = 0x56354153a000

[pid 2921] futex(0x7fab5961477c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

[pid 2921] newfstatat(0, "", {st\_mode=S\_IFREG|0777, st\_size=2, ...}, AT\_EMPTY\_PATH) = 0

**[pid 2921] read(0, "3\n", 4096) = 2**

[pid 2921] lseek(0, -1, SEEK\_CUR) = 1

[pid 2921] exit\_group(0) = ?

[pid 2920] <... read resumed>"", 4) = 0

**[pid 2920] close(4) = 0**

[pid 2920] lseek(0, -1, SEEK\_CUR) = -1 ESPIPE (Illegal seek)

[pid 2920] exit\_group(0) = ?

[pid 2920] +++ exited with 0 +++

+++ exited with 0 +++

**Вывод**

В ходе лабораторной работы я научился работать с процессами в языке Си. Также я использовал pipe для общения между процессами и системные вызовы для передачи данных в pipe. Еще мне было необходимо переопределить вывод и ввод дочернего процесса, поэтому я разобрался в системном вызове dup2. При выполнении работы я использовал язык C++, в котором доступны функции cin/cout, которые я старался использовать везде, поэтому я столкнулся с тем, что не мог понять, почему не работает передача данных через pipe.