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-22

Студент: Седов М.Д

Преподаватель: Миронов Е.С.

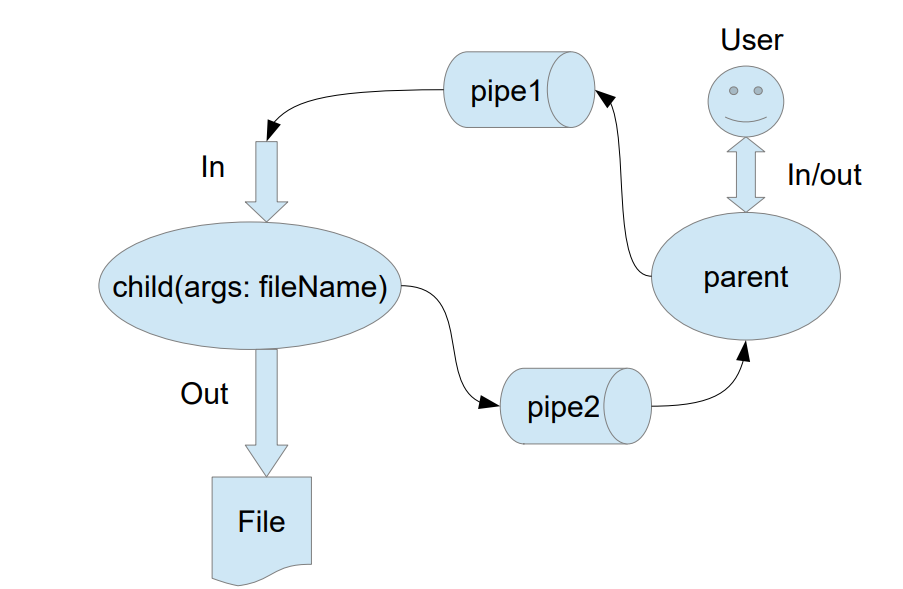
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 06.10.2023

Москва, 2023

**Постановка задачи**

**Вариант 5.**



Родительский процесс создает дочерний процесс. Первой строчкой пользователь в консоль родительского процесса пишет имя файла, которое будет передано при создании дочернего процесса. Родительский и дочерний процесс представлены разными программами. Пользователь вводит команды вида: «число<endline>». Родительский процесс передает команды пользователя через pipe1, который связан с стандартным входным потоком дочернего процесса. Дочерний процесс производит проверку на простоту. Если число составное, то в это число записывается в созданный/открытый им файл. Если число отрицательное или простое, то тогда дочерний и родительский процессы завершаются, при этом дочерний процесс передает сигнал в родительский процесс через pipe2.

**Общий метод и алгоритм решения**

Использованные системные вызовы:

* pid\_t fork(void); – создает дочерний процесс.
* int pipe(int \*fd); – создает канал для связи процессов.
* int close(int fd); - закрывает файловый дескриптор.
* int dup2(int oldfd, int newfd); - копирует файловый дескриптор;
* int execl(const char\* path, const char\* age, …); - запускает исполняемый файл
* ssize\_t write(int \_\_fd, const void \*\_\_buf, size\_t \_\_n); - записывает данные из буфера в файловый дескриптор.
* ssize\_t read(int \_\_fd, void \*\_\_buf, size\_t \_\_nbytes); - считывает данные из файлового дескриптора в буфер.
* int open(const char \*\_\_file, int \_\_oflag, ...); - открывает/создает файл.

Cоздал 2 канала с помощью pipe(), затем создал дочерний процесс в родительском с помощью fork() . Организовал ветвление с помощью идентификатора процесса {pid}. В дочернем процессе и родительском процессах закрыл ненужные файловые дескрипторы в каналах, используя close(). Затем в дочернем скопировал файловый дескриптор на чтение и на вывод, на потоки чтения и вывода {dup2()}, затем закрыл файловые дескрипторы и вызвал исполняемый файл child с помощью execl(). В родительском процессе считываю имя файла с консоли и записываю в файловый дескриптор канала с помощью write(), затем считываю с консоли числа и записываю их в файловый дескриптор, ожидая сообщения об успехе/неуспехе от дочернего процесса с помощью read(). В это время в дочернем процессе считывается имя файла полученное от родительского процесса и открывается/создается файл для записи с помощью open(). Затем дочерний процесс считывает числа от родительского, проверяет их на то, что они составные и записывает от результата или в выходной файл, или в файловый дескриптор для дальнейшей обработки в родительском процессе. Дальше если ввод в консоль прекращен, или число оказалось не составным, дочерний и родительские процессы завершаются, закрывая все файловые дескрипторы.

**Код программы**

**main.cpp**

#include <iostream>

#include <unistd.h>

#include <stdlib.h>

#include <sys/wait.h>

#include <fcntl.h>

// #include "composite.h"

// #define \_R\_            "\x1b[31m" // red color for child process output color

// #define \_G\_            "\x1b[32m" // green color for parent process output color

int create\_process(){

    pid\_t pid = fork();

    if( pid == -1){

        perror("fork");

        exit(-1);

    }

    return pid;

};

void create\_pipe(int\* fd){

    if(pipe(fd) == -1){

        perror("pipe");

        exit(-1);

    }

}

int main(){

    int fd1[2],fd2[2];

    create\_pipe(fd1);

    create\_pipe(fd2);

    pid\_t pid = create\_process();

    if(pid == 0){

        close(fd1[1]);

        close(fd2[0]);

        ;

        if (dup2(fd1[0], STDIN\_FILENO) == -1) {

            perror("dup2 in");

            exit(-1);

        }

        if (dup2(fd2[1], STDOUT\_FILENO) == -1) {

            perror("dup2 out ");

            exit(-1);

        }

        close(fd1[0]);

        close(fd2[1]);

        execl("../build/child", "../build/child", NULL);

    }

    else{

        close(fd1[0]);

        close(fd2[1]);

        char fname[40];

        printf("Enter file name:\n");

        scanf("%39s", fname);

        if (write(fd1[1], fname, sizeof(fname)) == 0) {

            perror("parent filename write");

            exit(-1);

        }

        int num;

        int is\_composite;

        printf("Enter num:\n");

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

            if (write(fd1[1], &num, sizeof(int)) == 0) {

                perror("parent write");

                break;

            }

            if(read(fd2[0], &is\_composite, sizeof(int)) == 0) {

                perror("parent read");

                break;

            }

            if (!is\_composite) {

                printf("Finish processes\n");

                break;

            }

        }

        close(fd1[1]);

        close(fd2[0]);

    }

    return 0;

}

**child.cpp**

#include <iostream>

#include <unistd.h>

#include <stdlib.h>

#include <fcntl.h>

#include "composite.h"

// #define \_R\_            "\x1b[31m" // red color for child process output color

// #define \_G\_            "\x1b[32m" // green color for parent process output color

int main() {

    char fname[40];

    if(read(STDIN\_FILENO, fname, sizeof(fname)) == 0){

        exit(-1);

    }

    int fdout = open(fname, O\_WRONLY | O\_CREAT | O\_TRUNC, 0777);

    if (fdout == -1) {

        perror("fdout");

        exit(-1);

    }

    int num;

    while (read(STDIN\_FILENO, &num, sizeof(int))) {

        int flag = isComposite(num);

        if(flag){

            if(write(fdout, &num, sizeof(int)) == 0){

                perror("fdout write");

                break;

            }

            if(write(STDOUT\_FILENO, &flag, sizeof(int)) == 0){

                perror("stdout write");

                break;

            }

        }

        else{

            if(write(STDOUT\_FILENO, &flag, sizeof(int)) == 0){

                perror("stdout write");

                break;

            }

            break;

        }

    }

    close(fdout);

    return 0;

}

**composite.h**

#pragma once

#include <iostream>

int isComposite(int num);

**composite.cpp**

#include <iostream>

#include "composite.h"

int isComposite(int num){

    if(num <= 0){

        return 0;

    }

    if(num == 1){

        return 1;

    }

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

        if(num % i == 0){

            return 1;

        }

    }

    return 0;

}

**Протокол работы программы**

**Тестирование:**

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./main

Enter file name:

toji

Enter num:

1000

26

64

3

Finish processes

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./flread

1000

26

64

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./main

Enter file name:

toji

Enter num:

46

86

90

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./flread

46

86

90

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./main

Enter file name:

toji

Enter num:

000000000000

Finish processes

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./flread

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./main

Enter file name:

toji

Enter num:

-1000

Finish processes

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./flread

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./main

Enter file name:

toji

Enter num:

-+^#

Finish processes

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./flread

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./main

Enter file name:

toji

Enter num:

75

54

67

Finish processes

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ ./flread

75

54

**STRACE:**

maksim@caseofpeace:~/course2/sem3/os/lab1/build$ strace -f ./main

execve("./main", ["./main"], 0x7fff22506918 /\* 37 vars \*/) = 0

brk(NULL) = 0x55d1bf50d000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffeafc60e60) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fd851990000

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=19407, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 19407, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fd85198b000

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) = 0x7fd85175f000

mprotect(0x7fd8517f9000, 1576960, PROT\_NONE) = 0

mmap(0x7fd8517f9000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7fd8517f9000

mmap(0x7fd85190a000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7fd85190a000

mmap(0x7fd85197a000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7fd85197a000

mmap(0x7fd851988000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fd851988000

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\"\233}\305\t\5?\344\337^)\350b\231\21\360"..., 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) = 0x7fd851537000

mmap(0x7fd85155f000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7fd85155f000

mmap(0x7fd8516f4000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7fd8516f4000

mmap(0x7fd85174c000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7fd85174c000

mmap(0x7fd851752000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fd851752000

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) = 0x7fd851450000

mmap(0x7fd85145e000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7fd85145e000

mmap(0x7fd8514da000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7fd8514da000

mmap(0x7fd851535000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7fd851535000

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) = 0x7fd851430000

mmap(0x7fd851433000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fd851433000

mmap(0x7fd85144a000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7fd85144a000

mmap(0x7fd85144e000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7fd85144e000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fd85142e000

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fd85142b000

arch\_prctl(ARCH\_SET\_FS, 0x7fd85142b740) = 0

set\_tid\_address(0x7fd85142ba10) = 24504

set\_robust\_list(0x7fd85142ba20, 24) = 0

rseq(0x7fd85142c0e0, 0x20, 0, 0x53053053) = 0

mprotect(0x7fd85174c000, 16384, PROT\_READ) = 0

mprotect(0x7fd85144e000, 4096, PROT\_READ) = 0

mprotect(0x7fd851535000, 4096, PROT\_READ) = 0

mprotect(0x7fd85197a000, 45056, PROT\_READ) = 0

mprotect(0x55d1be2c5000, 4096, PROT\_READ) = 0

mprotect(0x7fd8519ca000, 8192, PROT\_READ) = 0

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

munmap(0x7fd85198b000, 19407) = 0

getrandom("\x42\x37\xb5\x82\x0d\x02\x34\x55", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55d1bf50d000

brk(0x55d1bf52e000) = 0x55d1bf52e000

futex(0x7fd85198877c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

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

**pipe2([5, 6], 0) = 0**

**clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLDstrace: Process 24505 attached**

**, child\_tidptr=0x7fd85142ba10) = 24505**

[pid 24505] set\_robust\_list(0x7fd85142ba20, 24 <unfinished ...>

**[pid 24504] close(3 <unfinished ...>**

[pid 24505] <... set\_robust\_list resumed>) = 0

[pid 24504] <... close resumed>) = 0

**[pid 24504] close(6 <unfinished ...>**

**[pid 24505] close(4 <unfinished ...>**

[pid 24504] <... close resumed>) = 0

[pid 24505] <... close resumed>) = 0

[pid 24504] newfstatat(1, "", <unfinished ...>

**[pid 24505] close(5 <unfinished ...>**

[pid 24504] <... newfstatat resumed>{st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x8), ...}, AT\_EMPTY\_PATH) = 0

[pid 24505] <... close resumed>) = 0

[pid 24504] write(1, "Enter file name:\n", 17 <unfinished ...>

Enter file name:

**[pid 24505] dup2(3, 0 <unfinished ...>**

[pid 24504] <... write resumed>) = 17

[pid 24505] <... dup2 resumed>) = 0

[pid 24504] newfstatat(0, "", <unfinished ...>

**[pid 24505] dup2(6, 1 <unfinished ...>**

[pid 24504] <... newfstatat resumed>{st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x8), ...}, AT\_EMPTY\_PATH) = 0

[pid 24505] <... dup2 resumed>) = 1

[pid 24504] read(0, <unfinished ...>

[pid 24505] close(3) = 0

[pid 24505] close(6) = 0

**[pid 24505] execve("../build/child", ["../build/child"], 0x7ffeafc61038 /\* 37 vars \*/) = 0**

[pid 24505] brk(NULL) = 0x55ec835b7000

[pid 24505] arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffecbddc820) = -1 EINVAL (Invalid argument)

[pid 24505] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f67f14b2000

[pid 24505] access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory)

[pid 24505] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

[pid 24505] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=19407, ...}, AT\_EMPTY\_PATH) = 0

[pid 24505] mmap(NULL, 19407, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f67f14ad000

[pid 24505] close(3) = 0

[pid 24505] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3

[pid 24505] 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 24505] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=2260296, ...}, AT\_EMPTY\_PATH) = 0

[pid 24505] mmap(NULL, 2275520, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f67f1281000

[pid 24505] mprotect(0x7f67f131b000, 1576960, PROT\_NONE) = 0

[pid 24505] mmap(0x7f67f131b000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7f67f131b000

[pid 24505] mmap(0x7f67f142c000, 454656, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7f67f142c000

[pid 24505] mmap(0x7f67f149c000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x21a000) = 0x7f67f149c000

[pid 24505] mmap(0x7f67f14aa000, 10432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f67f14aa000

[pid 24505] close(3) = 0

[pid 24505] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

[pid 24505] 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 24505] 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 24505] 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 24505] pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\"\233}\305\t\5?\344\337^)\350b\231\21\360"..., 68, 896) = 68

[pid 24505] newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2216304, ...}, AT\_EMPTY\_PATH) = 0

[pid 24505] 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 24505] mmap(NULL, 2260560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f67f1059000

[pid 24505] mmap(0x7f67f1081000, 1658880, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f67f1081000

[pid 24505] mmap(0x7f67f1216000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) = 0x7f67f1216000

[pid 24505] mmap(0x7f67f126e000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x214000) = 0x7f67f126e000

[pid 24505] mmap(0x7f67f1274000, 52816, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f67f1274000

[pid 24505] close(3) = 0

[pid 24505] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 3

[pid 24505] 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 24505] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=940560, ...}, AT\_EMPTY\_PATH) = 0

[pid 24505] mmap(NULL, 942344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f67f0f72000

[pid 24505] mmap(0x7f67f0f80000, 507904, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f67f0f80000

[pid 24505] mmap(0x7f67f0ffc000, 372736, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x8a000) = 0x7f67f0ffc000

[pid 24505] mmap(0x7f67f1057000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe4000) = 0x7f67f1057000

[pid 24505] close(3) = 0

[pid 24505] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3

[pid 24505] 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 24505] newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=125488, ...}, AT\_EMPTY\_PATH) = 0

[pid 24505] mmap(NULL, 127720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f67f0f52000

[pid 24505] mmap(0x7f67f0f55000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f67f0f55000

[pid 24505] mmap(0x7f67f0f6c000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7f67f0f6c000

[pid 24505] mmap(0x7f67f0f70000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7f67f0f70000

[pid 24505] close(3) = 0

[pid 24505] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f67f0f50000

[pid 24505] mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f67f0f4d000

[pid 24505] arch\_prctl(ARCH\_SET\_FS, 0x7f67f0f4d740) = 0

[pid 24505] set\_tid\_address(0x7f67f0f4da10) = 24505

[pid 24505] set\_robust\_list(0x7f67f0f4da20, 24) = 0

[pid 24505] rseq(0x7f67f0f4e0e0, 0x20, 0, 0x53053053) = 0

[pid 24505] mprotect(0x7f67f126e000, 16384, PROT\_READ) = 0

[pid 24505] mprotect(0x7f67f0f70000, 4096, PROT\_READ) = 0

[pid 24505] mprotect(0x7f67f1057000, 4096, PROT\_READ) = 0

[pid 24505] mprotect(0x7f67f149c000, 45056, PROT\_READ) = 0

[pid 24505] mprotect(0x55ec829b1000, 4096, PROT\_READ) = 0

[pid 24505] mprotect(0x7f67f14ec000, 8192, PROT\_READ) = 0

[pid 24505] prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

[pid 24505] munmap(0x7f67f14ad000, 19407) = 0

[pid 24505] getrandom("\x44\x17\x11\xbc\xb1\x8a\x13\x07", 8, GRND\_NONBLOCK) = 8

[pid 24505] brk(NULL) = 0x55ec835b7000

[pid 24505] brk(0x55ec835d8000) = 0x55ec835d8000

[pid 24505] futex(0x7f67f14aa77c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

[pid 24505] read(0, toji

<unfinished ...>

[pid 24504] <... read resumed>"toji\n", 1024) = 5

**[pid 24504] write(4, "toji\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 40) = 40**

**[pid 24505] <... read resumed>"toji\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"..., 40) = 40**

[pid 24504] write(1, "Enter num:\n", 11 <unfinished ...>

Enter num:

**[pid 24505] openat(AT\_FDCWD, "toji", O\_WRONLY|O\_CREAT|O\_TRUNC, 0777 <unfinished ...>**

[pid 24504] <... write resumed>) = 11

[pid 24504] read(0, <unfinished ...>

[pid 24505] <... openat resumed>) = 3

[pid 24505] read(0, 100

<unfinished ...>

[pid 24504] <... read resumed>"100\n", 1024) = 4

**[pid 24504] write(4, "d\0\0\0", 4) = 4**

[pid 24504] read(5, <unfinished ...>

**[pid 24505] <... read resumed>"d\0\0\0", 4) = 4**

**[pid 24505] write(3, "d\0\0\0", 4) = 4**

**[pid 24505] write(1, "\1\0\0\0", 4 <unfinished ...>**

**[pid 24504] <... read resumed>"\1\0\0\0", 4) = 4**

[pid 24505] <... write resumed>) = 4

[pid 24504] read(0, <unfinished ...>

[pid 24505] read(0, 24

<unfinished ...>

[pid 24504] <... read resumed>"24\n", 1024) = 3

**[pid 24504] write(4, "\30\0\0\0", 4) = 4**

[pid 24505] <... read resumed>"\30\0\0\0", 4) = 4

[pid 24504] read(5, <unfinished ...>

**[pid 24505] write(3, "\30\0\0\0", 4) = 4**

**[pid 24505] write(1, "\1\0\0\0", 4 <unfinished ...>**

**[pid 24504] <... read resumed>"\1\0\0\0", 4) = 4**

[pid 24505] <... write resumed>) = 4

[pid 24504] read(0, <unfinished ...>

[pid 24505] read(0, 25

<unfinished ...>

[pid 24504] <... read resumed>"25\n", 1024) = 3

**[pid 24504] write(4, "\31\0\0\0", 4) = 4**

[pid 24505] <... read resumed>"\31\0\0\0", 4) = 4

[pid 24504] read(5, <unfinished ...>

**[pid 24505] write(3, "\31\0\0\0", 4) = 4**

**[pid 24505] write(1, "\1\0\0\0", 4 <unfinished ...>**

**[pid 24504] <... read resumed>"\1\0\0\0", 4) = 4**

[pid 24505] <... write resumed>) = 4

[pid 24504] read(0, <unfinished ...>

[pid 24505] read(0, 53

<unfinished ...>

[pid 24504] <... read resumed>"53\n", 1024) = 3

**[pid 24504] write(4, "5\0\0\0", 4) = 4**

[pid 24505] <... read resumed>"5\0\0\0", 4) = 4

[pid 24504] read(5, <unfinished ...>

**[pid 24505] write(1, "\0\0\0\0", 4 <unfinished ...>**

**[pid 24504] <... read resumed>"\0\0\0\0", 4) = 4**

[pid 24505] <... write resumed>) = 4

[pid 24504] write(1, "Finish processes\n", 17 <unfinished ...>

Finish processes

**[pid 24505] close(3 <unfinished ...>**

[pid 24504] <... write resumed>) = 17

[pid 24505] <... close resumed>) = 0

**[pid 24504] close(4) = 0**

[pid 24505] exit\_group(0 <unfinished ...>

**[pid 24504] close(5 <unfinished ...>**

[pid 24505] <... exit\_group resumed>) = ?

[pid 24504] <... close resumed>) = 0

[pid 24505] +++ exited with 0 +++

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=24505, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

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

exit\_group(0) = ?

+++ exited with 0 +++

**Вывод**

Разобрался в межпроцессорном взаимодействии через создание процессов и использование каналов для их связи. Изначально мне было трудно понять, как все это работает "внутри" и как происходит взаимодействие между процессами, но благодаря пробам, ошибкам и тестированию я смог осознать это. Самой большой проблемой была моя невнимательность, так как не заметил, что неправильно реализовал передачу сообщения о проверке числа обратно в родительский процесс. Это привело к лишней работе и потере времени.