

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
Студентка: Шипилова Татьяна
Преподаватель: Миронов Е.С.
Оценка: _____
Дата: 03.10.2023

Москва, 2023

Постановка задачи

Вариант 16.

Родительский процесс создает дочерний процесс. Первой строкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия File с таким именем на запись. Перенаправление стандартных потоков ввода-вывода показано на картинке выше. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс принимает от пользователя строки произвольной длины и пересылает их в pipe1. Процесс child проверяет строки на валидность правилу. Если строка соответствует правилу, то она выводится в стандартный поток вывода дочернего процесса, иначе в pipe2 выводится информация об ошибке. Родительский процесс полученные от child ошибки выводит в стандартный поток вывода. Правило проверки: строка должна оканчиваться на «.» или «;»

Общий метод и алгоритм решения

Дочерний процесс создаем через fork(). Далее отделяем дочерний процесс. Закрываем неиспользуемые каналы и запускаем дочернюю программу. while в дочерней и в родительской. Родитель читает getline'ом пока не встретит EOF или в pipe2 не придет ошибка от дочернего процесса. Дочерний процесс читает из стандартного для него ввода, пока не появится строка, не соответствующая проверке, или длина считываемой строки не окажется -1, что свидетельствует о конце чтения.

Использованные системные вызовы:

- **pid_t fork(void)**; – создает дочерний процесс.
- **int pipe(int *fd)**; – создает канал для межпроцессного взаимодействия.
- **istream& getline(istream& is, string& str)**; – считывает символы до \n или EOF.
- **int select(int nfds, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout)**; – используется для ожидания событий ввода/вывода на одном или нескольких файловых дескрипторах.
- **int open(const char *pathname, int flags, mode_t mode)**; – используется для открытия(создания) файла.
- **int execl(const char *path, const char *arg0, ... /* (char *) NULL */)**; – используется для запуска другой программы.
- **int close(int fd)**; – используется для закрытия файловых дескрипторов.
- **int dup2(int oldfd, int newfd)**; – используется для создания копии файлового дескриптора.
- **ssize_t write(int fd, const void *buf, size_t count)**; – используется для записи данных в файловый дескриптор.
- **ssize_t read(int fd, void *buf, size_t count)**; – используется для чтения данных из файлового дескриптора.

Код программы

main.c

```
#include <iostream>

#include <unistd.h>

#include <sys/wait.h>

#include <fcntl.h>

#include "isPipeEmpty.h"


using namespace std;


pid_t create_process() {
    pid_t pid = fork();
    if (-1 == pid)
    {
        perror("fork");
        exit(EXIT_FAILURE);
    }
    return pid;
}


int main(){
    string filename;
    cout << "Введите имя файла: \n";
    cin >> filename;

    int fileFd = open(filename.c_str(), O_CREAT | O_WRONLY, 0777);
    if (fileFd == -1){
        cerr << "can't open file: " << filename;
        exit(EXIT_FAILURE);
    }
}
```

```

int pipe1[2], pipe2[2];
if (pipe(pipe1) == -1 || pipe(pipe2) == -1){
    cerr << "fail pipe()";
    exit(EXIT_FAILURE);
}

pid_t childPid = create_process();

if (childPid == 0) { // child process
    close(pipe1[1]);
    close(pipe2[0]);
    dup2(pipe1[0], STDIN_FILENO);
    dup2(pipe2[1], STDERR_FILENO);
    dup2(fileFd, STDOUT_FILENO);

    if(execl("./child", "./child", NULL) == -1) { // exec child process
        cout << "can't exec child process";
        exit(-1);
    }

    close(pipe1[0]);
    close(pipe2[1]);
} else { // parent process
    close(pipe1[0]);
    close(pipe2[1]);

    cin.ignore(); //для \n, который остался после введения имени файла
    bool childErr = false;

    string inputLine;
    int inputLineLen = 0;
    while (getline(cin, inputLine)){

```

```

        if(false == isPipeEmpty(pipe2[0])) {
            childErr = true;
            break;
        }
        inputLineLen = inputLine.length();
        write(pipe1[1], &inputLineLen, sizeof(inputLineLen));
        write(pipe1[1], inputLine.c_str(), inputLine.length());
    }
    if (!childErr){
        int likeEnd = -1;
        write(pipe1[1], &likeEnd, sizeof(likeEnd));
    }

    wait(NULL);

    int errorLen = 0;
    if(read(pipe2[0], &errorLen, sizeof(errorLen)) > 0){
        char error[errorLen];
        read(pipe2[0], error, errorLen);
        cout << error << endl;
    }

    close(pipe1[1]);
    close(pipe2[0]);
}
close(fileFd);
return 0;
}

```

child.cpp

```

#include <unistd.h>
#include <string.h>
#include <iostream>

```

```

using namespace std;

#define LIKEEND -1

int main() {

    string error {"Строка не оканчивается на \".\" или \";\"; "};
    int inputLineLen;
    while (1) {
        read(STDIN_FILENO, &inputLineLen, sizeof(inputLineLen));
        if(inputLineLen == LIKEEND) break;

        char inputLine[inputLineLen];
        read(STDIN_FILENO, inputLine, inputLineLen);

        if (inputLine[inputLineLen - 1] == '.' || inputLine[inputLineLen - 1] ==
';') {

            inputLine[inputLineLen] = '\n'; inputLineLen++;
            write(STDOUT_FILENO, inputLine, inputLineLen);
        } else {
            for (int i = 0; i < inputLineLen; i++){
                error += inputLine[i];
            }
            error += '\0'; // без него добавляется мусор
            int errorLen = error.length();
            write(STDERR_FILENO, &errorLen, sizeof(errorLen));
            write(STDERR_FILENO, error.c_str(), error.length());
            break;
        }
    }
}

```

```
        return 0;
    }
}
```

isPipeEmpty.cpp

```
#include "isPipeEmpty.h"

#include <iostream>

bool isPipeEmpty(int fd) {

    fd_set rfds;

    FD_ZERO(&rfds);

    FD_SET(fd, &rfds);

    struct timeval tv;

    tv.tv_sec = 0;

    tv.tv_usec = 0;

    return (select(fd + 1, &rfds, NULL, NULL, &tv) == 0);

}
```

Протокол работы программы

Тестирование:

```
- $ ./main
```

Введите имя файла:

```
foutput.txt
```

```
maimai;
```

```
maimai.
```

```
chill;
```

```
ban
```

```
aaa;
```

Строка не оканчивается на "." или ";": ban

```
$ cat foutput.txt
```

```
maimai;
```

```
maimai.
```

```
chill;
```

```
- $ ./main
```

Введите имя файла:

```
foutput.txt
```

```
ban1
```

```
bban
```

Строка не оканчивается на "." или ";": ban1

```
$ cat foutput.txt
```

```
- $ ./main < f1.in
```

Введите имя файла:

Строка не оканчивается на "." или ";":
,wejafra,wejafragbferjagraeghuirhelagjreuhlbvrrugheriubguerugfhuierhg,wejafragb
ferjagraeghuirhelagjreuhlbvrrugheriubguerugfhuierhg,wejafragbferjagraeghuirhela
gjreuhlbvrrugheriubguerugfhuierhg,

```
$ cat f1.in
```

```
outputFile.txt
```

```
rgarg.
```

```
sgrr.
```

```
agf.
```


[illegible]

[illegible]

```
- $ ./main < f1.in
```

Введите имя файла:

```
$ cat f1.in
```

outputFile.txt

rgarg.

sgrr.

[illegible]

Strace:

```
$ strace -f ./main

execve("./main", [ "./main" ], 0x7ffce4389438 /* 35 vars */) = 0

brk(NULL)                               = 0x560696bb9000

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc6ce60700) = -1 EINVAL (Invalid
argument)

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x7f1f851d7000

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=26299, ...},
AT_EMPTY_PATH) = 0

mmap(NULL, 26299, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f1f851d0000

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\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"... , 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2252096, ...},
AT_EMPTY_PATH) = 0

mmap(NULL, 2267328, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f1f84fa6000

mmap(0x7f1f85040000, 1114112, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9a000) = 0x7f1f85040000

mmap(0x7f1f85150000, 454656, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1aa000) = 0x7f1f85150000

mmap(0x7f1f851bf000, 57344, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x218000) = 0x7f1f851bf000

mmap(0x7f1f851cd000, 10432, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f1f851cd000

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\3\0>\0\1\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) =
0x7f1f84f86000

    mmap(0x7f1f84f89000, 94208, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f1f84f89000

    mmap(0x7f1f84fa0000, 16384, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7f1f84fa0000

    mmap(0x7f1f84fa4000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x7f1f84fa4000

    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\3\0>\0\1\0\0\0P\237\2\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"... , 48, 848) = 48

    pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\360\352,\223\340."... ,
68, 896) = 68

    newfstatat(3, "", {st_mode=S_IFREG|0644, 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) =
0x7f1f84d5e000

    mmap(0x7f1f84d86000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f1f84d86000

    mmap(0x7f1f84f1b000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f1f84f1b000

    mmap(0x7f1f84f73000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7f1f84f73000

```



```

mmap(0x7f1f84f79000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f1f84f79000

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\3\0>\0\1\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) =
0x7f1f84c77000

mmap(0x7f1f84c85000, 507904, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7f1f84c85000

mmap(0x7f1f84d01000, 372736, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f1f84d01000

mmap(0x7f1f84d5c000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7f1f84d5c000

close(3) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x7f1f84c75000

arch_prctl(ARCH_SET_FS, 0x7f1f84c763c0) = 0

set_tid_address(0x7f1f84c76690) = 2456

set_robust_list(0x7f1f84c766a0, 24) = 0

rseq(0x7f1f84c76d60, 0x20, 0, 0x53053053) = 0

mprotect(0x7f1f84f73000, 16384, PROT_READ) = 0

mprotect(0x7f1f84d5c000, 4096, PROT_READ) = 0

mprotect(0x7f1f84fa4000, 4096, PROT_READ) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x7f1f84c73000

mprotect(0x7f1f851bf000, 45056, PROT_READ) = 0

mprotect(0x560694e74000, 4096, PROT_READ) = 0

mprotect(0x7f1f85211000, 8192, PROT_READ) = 0

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

```

```

munmap(0x7f1f851d0000, 26299) = 0

getrandom("\x63\x38\xee\x06\xa4\x0c\xf5\x05", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x560696bb9000

brk(0x560696bda000) = 0x560696bda000

futex(0x7f1f851cd77c, FUTEX_WAKE_PRIVATE, 2147483647) = 0

newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x3),
...}, AT_EMPTY_PATH) = 0

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\270\320\274\321\217 \321\204\320\260\320\271\320\273\320\260"...
, 35Введите имя файла:

) = 35

newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x3),
...}, AT_EMPTY_PATH) = 0

read(0, 0x560696bcb2c0, 1024) = ? ERESTARTSYS (To be restarted
if SA_RESTART is set)

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

read(0, 0x560696bcb2c0, 1024) = ? ERESTARTSYS (To be restarted
if SA_RESTART is set)

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

read(0, foutput.txt
"foutput.txt\n", 1024) = 12

openat(AT_FDCWD, "foutput.txt", O_WRONLY|O_CREAT, 0777) = 3

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

pipe2([6, 7], 0) = 0

clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace: Process 2624
attached

, child_tidptr=0x7f1f84c76690) = 2624

[pid 2624] set_robust_list(0x7f1f84c766a0, 24 <unfinished ...>

[pid 2456] close(4 <unfinished ...>

[pid 2624] <... set_robust_list resumed>) = 0

[pid 2456] <... close resumed>) = 0

[pid 2624] close(5 <unfinished ...>

```

```

[pid 2456] close(7 <unfinished ...>

[pid 2624] <... close resumed>          = 0

[pid 2456] <... close resumed>          = 0

[pid 2624] close(6 <unfinished ...>

[pid 2456] read(0, <unfinished ...>

[pid 2624] <... close resumed>          = 0

[pid 2624] dup2(4, 0)                   = 0

[pid 2624] dup2(7, 2)                   = 2

[pid 2624] dup2(3, 1)                   = 1

[pid 2624] execve("./child", ["./child"], 0x7ffc6ce608d8 /* 35 vars */)
= 0

[pid 2624] brk(NULL)                    = 0x55b2ccdbb000

[pid 2624] arch_prctl(0x3001 /* ARCH_??? */ , 0x7ffcdee8ea50) = -1 EINVAL
(Invalid argument)

[pid 2624] mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f44a3e1a000

[pid 2624] access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file
or directory)

[pid 2624] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 5

[pid 2624] newfstatat(5, "", {st_mode=S_IFREG|0644, st_size=26299, ...},
AT_EMPTY_PATH) = 0

[pid 2624] mmap(NULL, 26299, PROT_READ, MAP_PRIVATE, 5, 0) =
0x7f44a3e13000

[pid 2624] close(5)                     = 0

[pid 2624] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libstdc++.so.6",
O_RDONLY|O_CLOEXEC) = 5

[pid 2624] read(5,
"\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 2624] newfstatat(5, "", {st_mode=S_IFREG|0644, st_size=2252096,
...}, AT_EMPTY_PATH) = 0

[pid 2624] mmap(NULL, 2267328, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 5,
0) = 0x7f44a3be9000

[pid 2624] mmap(0x7f44a3c83000, 1114112, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x9a000) = 0x7f44a3c83000

```

```

    [pid 2624] mmap(0x7f44a3d93000, 454656, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x1aa000) = 0x7f44a3d93000

    [pid 2624] mmap(0x7f44a3e02000, 57344, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x218000) = 0x7f44a3e02000

    [pid 2624] mmap(0x7f44a3e10000, 10432, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f44a3e10000

    [pid 2624] close(5)                                = 0

    [pid 2624] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1",
O_RDONLY|O_CLOEXEC) = 5

    [pid 2624] read(5,
"\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0"... , 832) = 832

    [pid 2624] newfstatat(5, "", {st_mode=S_IFREG|0644, st_size=125488,
...}, AT_EMPTY_PATH) = 0

    [pid 2624] mmap(NULL, 127720, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 5,
0) = 0x7f44a3bc9000

    [pid 2624] mmap(0x7f44a3bcc000, 94208, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x3000) = 0x7f44a3bcc000

    [pid 2624] mmap(0x7f44a3be3000, 16384, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x1a000) = 0x7f44a3be3000

    [pid 2624] mmap(0x7f44a3be7000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x1d000) = 0x7f44a3be7000

    [pid 2624] close(5)                                = 0

    [pid 2624] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6",
O_RDONLY|O_CLOEXEC) = 5

    [pid 2624] read(5,
"\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"... , 832) =
832

    [pid 2624] pread64(5,
"\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 2624] pread64(5, "\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 2624] pread64(5,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\360\352,\223\340..." ,
68, 896) = 68

    [pid 2624] newfstatat(5, "", {st_mode=S_IFREG|0644, st_size=2216304,
...}, AT_EMPTY_PATH) = 0

```

```

[pid 2624] pread64(5,
"\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 2624] mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 5,
0) = 0x7f44a39a1000

[pid 2624] mmap(0x7f44a39c9000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x28000) = 0x7f44a39c9000

[pid 2624] mmap(0x7f44a3b5e000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x1bd000) = 0x7f44a3b5e000

[pid 2624] mmap(0x7f44a3bb6000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x214000) = 0x7f44a3bb6000

[pid 2624] mmap(0x7f44a3bbc000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f44a3bbc000

[pid 2624] close(5) = 0

[pid 2624] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6",
O_RDONLY|O_CLOEXEC) = 5

[pid 2624] read(5,
"\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 2624] newfstatat(5, "", {st_mode=S_IFREG|0644, st_size=940560,
...}, AT_EMPTY_PATH) = 0

[pid 2624] mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 5,
0) = 0x7f44a38ba000

[pid 2624] mmap(0x7f44a38c8000, 507904, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0xe000) = 0x7f44a38c8000

[pid 2624] mmap(0x7f44a3944000, 372736, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0x8a000) = 0x7f44a3944000

[pid 2624] mmap(0x7f44a399f000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 5, 0xe4000) = 0x7f44a399f000

[pid 2624] close(5) = 0

[pid 2624] mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f44a38b8000

[pid 2624] arch_prctl(ARCH_SET_FS, 0x7f44a38b93c0) = 0

[pid 2624] set_tid_address(0x7f44a38b9690) = 2624

[pid 2624] set_robust_list(0x7f44a38b96a0, 24) = 0

[pid 2624] rseq(0x7f44a38b9d60, 0x20, 0, 0x53053053) = 0

[pid 2624] mprotect(0x7f44a3bb6000, 16384, PROT_READ) = 0

```

```

[pid 2624] mprotect(0x7f44a399f000, 4096, PROT_READ) = 0

[pid 2624] mprotect(0x7f44a3be7000, 4096, PROT_READ) = 0

[pid 2624] mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f44a38b6000

[pid 2624] mprotect(0x7f44a3e02000, 45056, PROT_READ) = 0

[pid 2624] mprotect(0x55b2cb736000, 4096, PROT_READ) = 0

[pid 2624] mprotect(0x7f44a3e54000, 8192, PROT_READ) = 0

[pid 2624] prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0

[pid 2624] munmap(0x7f44a3e13000, 26299) = 0

[pid 2624] getrandom("\x98\x9b\x5b\x98\xeb\xc3\x0e\x5e", 8,
GRND_NONBLOCK) = 8

[pid 2624] brk(NULL) = 0x55b2ccdbb000

[pid 2624] brk(0x55b2ccddc000) = 0x55b2ccddc000

[pid 2624] futex(0x7f44a3e1077c, FUTEX_WAKE_PRIVATE, 2147483647) = 0

[pid 2624] read(0, pivot;
<unfinished ...>

[pid 2456] <... read resumed>"pivot;\n", 1024) = 7

[pid 2456] pselect6(7, [6], NULL, NULL, {tv_sec=0, tv_nsec=0}, NULL) = 0
(Timeout)

[pid 2456] write(5, "\6\0\0\0", 4) = 4

[pid 2624] <... read resumed>"\6\0\0\0", 4) = 4

[pid 2456] write(5, "pivot;", 6 <unfinished ...>

[pid 2624] read(0, <unfinished ...>

[pid 2456] <... write resumed>) = 6

[pid 2624] <... read resumed>"pivot;", 6) = 6

[pid 2456] read(0, <unfinished ...>

[pid 2624] write(1, "pivot;\n", 7) = 7

[pid 2624] read(0, hello;
<unfinished ...>

[pid 2456] <... read resumed>"hello;\n", 1024) = 7

```

```

    [pid 2456] pselect6(7, [6], NULL, NULL, {tv_sec=0, tv_nsec=0}, NULL) = 0
(Timeout)

    [pid 2456] write(5, "\6\0\0\0", 4)      = 4

    [pid 2624] <... read resumed>"\6\0\0\0", 4) = 4

    [pid 2456] write(5, "hello;", 6 <unfinished ...>

    [pid 2624] read(0, <unfinished ...>

    [pid 2456] <... write resumed>)          = 6

    [pid 2624] <... read resumed>"hello;", 6) = 6

    [pid 2456] read(0, <unfinished ...>

    [pid 2624] write(1, "hello;\n", 7)      = 7

    [pid 2624] read(0, ban?

    <unfinished ...>

    [pid 2456] <... read resumed>"ban?\n", 1024) = 5

    [pid 2456] pselect6(7, [6], NULL, NULL, {tv_sec=0, tv_nsec=0}, NULL) = 0
(Timeout)

    [pid 2456] write(5, "\4\0\0\0", 4)      = 4

    [pid 2624] <... read resumed>"\4\0\0\0", 4) = 4

    [pid 2456] write(5, "ban?", 4 <unfinished ...>

    [pid 2624] read(0, <unfinished ...>

    [pid 2456] <... write resumed>)          = 4

    [pid 2624] <... read resumed>"ban?", 4) = 4

    [pid 2456] read(0, <unfinished ...>

    [pid 2624] write(2, "E\0\0\0", 4)      = 4

    [pid 2624] write(2, "\320\241\321\202\321\200\320\276\320\272\320\260
\320\275\320\265 \320\276\320\272\320\260\320\275\321\207\320\270\320\262"...
, 69) = 69

    [pid 2624] exit_group(0)                  = ?

    [pid 2624] +++ exited with 0 +++

    <... read resumed>0x560696bcb2c0, 1024) = ? ERESTARTSYS (To be restarted
if SA_RESTART is set)

    --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=2624,
si_uid=1000, si_status=0, si_utime=1, si_stime=1} ---

```

```

read(0, baaan!

"baaan!\n", 1024)                = 7

pselect6(7, [6], NULL, NULL, {tv_sec=0, tv_nsec=0}, NULL) = 1 (in [6],
left {tv_sec=0, tv_nsec=0})

wait4(-1, NULL, 0, NULL)          = 2624

read(6, "E\0\0\0", 4)             = 4

read(6, "\320\241\321\202\321\200\320\276\320\272\320\260
\320\275\320\265 \320\276\320\272\320\260\320\275\321\207\320\270\320\262"... ,
69) = 69

write(1, "\320\241\321\202\321\200\320\276\320\272\320\260
\320\275\320\265 \320\276\320\272\320\260\320\275\321\207\320\270\320\262"... ,
69Строка не оканчивается на "." или ";": ban?

) = 69

close(5)                          = 0

close(6)                          = 0

close(3)                          = 0

exit_group(0)                     = ?

+++ exited with 0 +++

```


Вывод

В ходе выполнения лабораторной работы я научилась создавать межпроцессные каналы и работать с ними, запускать другой процесс из одного. При передаче строки из родительского процесса через канал столкнулась с тем, что со стороны дочернего неизвестно, какого размера будет строка. Для этого я первым байтом передавала размер строки в целочисленном формате. Также столкнулась с тем, что когда дочерний обнаруживал ошибку и добавлял неподходящую строку, то вместе с ней добавлялся мусор, но по отдельности выводились корректно. Для этого я добавляла неподходящую строку посимвольно, а также в конец `'\0'`. Поняла, что от скорости введения данных различаются поведение моей программы: при вводе из файла дочерний не успевает записать ошибку, когда родитель заканчивает чтение и завершается без ошибки. Для этого использовано `'wait()'`, который заставляет родителя ждать завершения дочернего процесса. Также решила не ждать каждый раз, пока дочерний процесс обработает строку, а сразу считывать следующую строку, поэтому при личном вводе при введении некорректной строки родитель сообщает только после того, как примет еще одну строку.