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: Жаднов М. Д.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 06.10.23

Москва, 2023

Постановка задачи

Группа вариантов 2.

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.

Вариант 8.

В файле записаны команды вида:«число число число<endline>». 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 int. 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 неименованный канал, у которого первое поле отвечает за чтение, а второе - за запись.
- int execv(const char *__path, char *const *__argv); - предоставляет новой программе список аргументов в виде массива указателей на строки, заканчивающиеся (char *)0.
- int dup2(int, int); - создает копию файлового дескриптора oldfd *(1 поле)*, используя для нового дескриптора newfd *(2 поле)* файловый дескриптор (они становятся взаимозаменяемыми).

- _exit(int status); – выходит из процесса с заданным статусом.

- pid_t wait(int *status); – приостаналивает выполнение текущего процесса до тех пор, пока дочерний процесс не завершится.

- int read(int fd, void *buffer, int nbyte); – читает nbyte байтов из файлового дескриптора fd в буффер buffer.

Программа parent.c принимает аргументом название файла, который нужно будет прочитать. Далее происходит проверка поданного файла на чтение, и если прочитался успешно, создаётся pipe и дочерний процесс (с дальнейшими проверками их создания, конечно же). Потом происходит перераспределение файловых дескрипторов стандартного ввода (на файл) и вывода (на pipe) в дочернем процессе. Следующим шагом дочерний процесс запускает программу child.c и обрабатывает свой стандартный ввод. В то же время, родительский процесс читает pipe и выводит полученные результаты в стандартный поток вывода, а если встречает -1 (что значит завершение программы дочернего процесса неудачей), то выводит сообщение об ошибке (“Division by zero”) и завершает работу.

Код программы

parent.c

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <fcntl.h>
#include <string.h>
#include <stdbool.h>


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

    if (argc != 2) {
        perror("\nError: no filename\n");
        exit(EXIT_FAILURE);
    }

    int fd = open(argv[1], O_RDONLY);
    if(fd == -1){
        perror("\nCan't open file\n");
        _exit(EXIT_FAILURE);
    }

    int pipe_fd[2];
    if (pipe(pipe_fd) == -1){
        perror("\npipe: Here is a problem\n");
        _exit(EXIT_FAILURE);
    }

    pid_t pid = fork();
```

```
    if (pid == -1) {
        perror("\nfork: Here is a problem\n");
        _exit(EXIT_FAILURE);
    }
    else if(pid == 0){ //child
        close(pipe_fd[0]);
        dup2(fd, STDIN_FILENO);
        dup2(pipe_fd[1], STDOUT_FILENO);
        char* args[] = {"./child", NULL};
        if (execv(args[0], args) == -1){
            fprintf(stderr, "Unable to exec\n");
            _exit(EXIT_FAILURE);
        }
    }else{ //parent
        close(pipe_fd[1]);
        wait(0);
        int result = 0;
        while(read(pipe_fd[0], &result, sizeof(int))){
            if(result == -1){
                printf("Division by zero\n");
                _exit(EXIT_FAILURE);
            }
            else printf("%d\n", result);
        }
    }
    return 0;
}
```

child.c

```
#include "stdio.h"
#include "stdlib.h"
#include "unistd.h"

int main(){
    int c = '\0';
    int tmp = 0, res = 0;
    int end_of_str = 0;

    do{
        if(!end_of_str){
            if(c>='0' && c<='9'){
                tmp = tmp*10 + c - '0';
            }
```

```
            if(c == ' ' || c == '\n' || c == EOF){
                if(res == 0 && tmp != 0){
                    res = tmp;
                }
                else if(res != 0 && tmp != 0){
                    res /= tmp;
                }
                else if(res == 0 && tmp == 0){
                    end_of_str = 1;
                }
                else if(res != 0 && tmp == 0){
                    res = -1;
                    write(STDOUT_FILENO, &res, sizeof(int));
                    _exit(EXIT_FAILURE);
                }
                tmp = 0;
            }
        }
        if(c == '\n' || c == EOF){
            write(STDOUT_FILENO, &res, sizeof(int));
            end_of_str = 0;
            res = 0;
        }
    }while(read(STDIN_FILENO, &c, sizeof(char)) > 0);

    return 0;
}
```

Протокол работы программы

Тестирование:

```
mishazhadnov@McB-airmi scr % ./parent test.txt
1
3
0

Division by zero
```

Dtrace (аналог strace):

```
mishazhadnov@McB-airmi scr % sudo dtruss -f ./parent test.txt

dtrace: system integrity protection is on, some features will not be available


  PID/THRD  SYSCALL(args)                 = return

51526/0x4e6cb5:  fork()          = 0 0

51526/0x4e6cb5:  munmap(0x113F14000, 0x9C000)           = 0 0

51526/0x4e6cb5:  munmap(0x113FB0000, 0x8000)            = 0 0

51526/0x4e6cb5:  munmap(0x113FB8000, 0x4000)            = 0 0

51526/0x4e6cb5:  munmap(0x113FBC000, 0x4000)            = 0 0

51526/0x4e6cb5:  munmap(0x113FC0000, 0x54000)           = 0 0

51526/0x4e6cb5:  open(".\0", 0x100000, 0x0)             = 3 0

51526/0x4e6cb5:  fcntl(0x3, 0x32, 0x7FF7B8F17230)               = 0 0

51526/0x4e6cb5:  close(0x3)             = 0 0

51526/0x4e6cb5:  fsgetpath(0x7FF7B8F17240, 0x400, 0x7FF7B8F17228)
= 58 0

51526/0x4e6cb5:  fsgetpath(0x7FF7B8F17240, 0x400, 0x7FF7B8F17228)
= 14 0

51526/0x4e6cb5:  csrctl(0x0, 0x7FF7B8F1764C, 0x4)               = -1 1

51526/0x4e6cb5:  __mac_syscall(0x7FF810C2E11B, 0x2, 0x7FF7B8F174C0)
= 0 0
```

```
51526/0x4e6cb5:  csrctl(0x0, 0x7FF7B8F1765C, 0x4)               = -1 1

51526/0x4e6cb5:  __mac_syscall(0x7FF810C2B0A8, 0x5A, 0x7FF7B8F175F0)
= 0 0

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

51526/0x4e6cb5:  open("/\0", 0x20100000, 0x0)            = 3 0

51526/0x4e6cb5:  openat(0x3, "System/Cryptexes/OS\0", 0x100000, 0x0)
= 4 0

1

3

0

Devision by zero

51526/0x4e6cb5:  dup(0x4, 0x0, 0x0)              = 5 0

51526/0x4e6cb5:  fstatat64(0x4, 0x7FF7B8F16391, 0x7FF7B8F16790)          = 0 0

51526/0x4e6cb5:  openat(0x4, "System/Library/dyld/\0", 0x100000, 0x0)
= 6 0

51526/0x4e6cb5:  fcntl(0x6, 0x32, 0x7FF7B8F16420)               = 0 0

51526/0x4e6cb5:  dup(0x6, 0x0, 0x0)              = 7 0

51526/0x4e6cb5:  dup(0x5, 0x0, 0x0)              = 8 0

51526/0x4e6cb5:  close(0x3)              = 0 0

51526/0x4e6cb5:  close(0x5)              = 0 0

51526/0x4e6cb5:  close(0x4)              = 0 0
```

```
51526/0x4e6cb5:  close(0x6)              = 0 0

51526/0x4e6cb5:  shared_region_check_np(0x7FF7B8F16D18, 0x0, 0x0)
= 0 0

51526/0x4e6cb5:  fsgetpath(0x7FF7B8F17270, 0x400, 0x7FF7B8F171A8)
= 83 0

51526/0x4e6cb5:  fcntl(0x8, 0x32, 0x7FF7B8F17270)               = 0 0

51526/0x4e6cb5:  close(0x8)              = 0 0

51526/0x4e6cb5:  close(0x7)              = 0 0

51526/0x4e6cb5:  getfsstat64(0x0, 0x0, 0x2)              = 8 0

51526/0x4e6cb5:  getfsstat64(0x106FEEA10, 0x43C0, 0x2)           = 8 0

51526/0x4e6cb5:  getattrlist("/\0", 0x7FF7B8F17100, 0x7FF7B8F17070)
= 0 0

51526/0x4e6cb5:  fsgetpath(0x7FF7B8F16EF0, 0x400, 0x7FF7B8F16ED8)
= 83 0

51526/0x4e6cb5:
stat64("/System/Volumes/Preboot/Cryptexes/OS/System/Library/dyld/dyld_shared_cac
he_x86_64h\0", 0x7FF7B8F17358, 0x0)             = 0 0

51526/0x4e6cb5:
stat64("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/parent\0",
0x7FF7B8F16980, 0x0)              = 0 0

51526/0x4e6cb5:
open("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/parent\0", 0x0, 0x0)
= 3 0

51526/0x4e6cb5:  mmap(0x0, 0x361C, 0x1, 0x40002, 0x3, 0x0)               =
0x10702D000 0

51526/0x4e6cb5:  fcntl(0x3, 0x32, 0x7FF7B8F16A90)               = 0 0

51526/0x4e6cb5:  close(0x3)              = 0 0

51526/0x4e6cb5:  munmap(0x10702D000, 0x361C)             = 0 0
```

```
51526/0x4e6cb5:
stat64("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/parent\0",
0x7FF7B8F16EE0, 0x0)              = 0 0

51526/0x4e6cb5:  stat64("/usr/lib/libSystem.B.dylib\0", 0x7FF7B8F15F30, 0x0)
= -1 2

51526/0x4e6cb5:
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libSystem.B.dylib\0",
0x7FF7B8F15EE0, 0x0)                 = -1 2

51526/0x4e6cb5:  stat64("/usr/lib/system/libdispatch.dylib\0", 0x7FF7B8F13B30,
0x0)              = -1 2

51526/0x4e6cb5:
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/system/libdispatch.dylib\0"
, 0x7FF7B8F13AE0, 0x0)          = -1 2

51526/0x4e6cb5:  stat64("/usr/lib/system/libdispatch.dylib\0", 0x7FF7B8F13B30,
0x0)              = -1 2

51526/0x4e6cb5:  open("/dev/dtracehelper\0", 0x2, 0x0)           = 3 0

51526/0x4e6cb5:  ioctl(0x3, 0x80086804, 0x7FF7B8F15B38)          = 0 0

51526/0x4e6cb5:  close(0x3)              = 0 0

51526/0x4e6cb5:  mprotect(0x106FE9000, 0x1000, 0x1)              = 0 0

51526/0x4e6cb5:  shared_region_check_np(0xFFFFFFFFFFFFFFFF, 0x0, 0x0)
= 0 0

51526/0x4e6cb5:  mprotect(0x106FEC000, 0x40000, 0x1)             = 0 0

51526/0x4e6cb5:  access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0)
= -1 2

51526/0x4e6cb5:  bsdthread_register(0x7FF810EF5BC4, 0x7FF810EF5BB0, 0x2000)
= 1073742303 0

51526/0x4e6cb5:  shm_open(0x7FF810DA0F5A, 0x0, 0x10D9F465)               = 3 0

51526/0x4e6cb5:  fstat64(0x3, 0x7FF7B8F15D80, 0x0)               = 0 0

51526/0x4e6cb5:  mmap(0x0, 0x3000, 0x1, 0x40001, 0x3, 0x0)               =
0x10702F000 0
```

```
51526/0x4e6cb5:  close(0x3)              = 0 0

51526/0x4e6cb5:  ioctl(0x2, 0x4004667A, 0x7FF7B8F15E34)          = 0 0

51526/0x4e6cb5:  mprotect(0x107037000, 0x1000, 0x0)              = 0 0

51526/0x4e6cb5:  mprotect(0x10703E000, 0x1000, 0x0)              = 0 0

51526/0x4e6cb5:  mprotect(0x10703F000, 0x1000, 0x0)              = 0 0

51526/0x4e6cb5:  mprotect(0x107046000, 0x1000, 0x0)              = 0 0

51526/0x4e6cb5:  mprotect(0x107032000, 0x98, 0x1)                = 0 0

51526/0x4e6cb5:  mprotect(0x107032000, 0x98, 0x3)                = 0 0

51526/0x4e6cb5:  mprotect(0x107032000, 0x98, 0x1)                = 0 0

51526/0x4e6cb5:  mprotect(0x107047000, 0x1000, 0x1)              = 0 0

51526/0x4e6cb5:  mprotect(0x107048000, 0x98, 0x1)                = 0 0

51526/0x4e6cb5:  mprotect(0x107048000, 0x98, 0x3)                = 0 0

51526/0x4e6cb5:  mprotect(0x107048000, 0x98, 0x1)                = 0 0

51526/0x4e6cb5:  mprotect(0x107032000, 0x98, 0x3)                = 0 0

51526/0x4e6cb5:  mprotect(0x107032000, 0x98, 0x1)                = 0 0

51526/0x4e6cb5:  mprotect(0x107047000, 0x1000, 0x3)              = 0 0

51526/0x4e6cb5:  mprotect(0x107047000, 0x1000, 0x1)              = 0 0

51526/0x4e6cb5:  mprotect(0x106FEC000, 0x40000, 0x3)             = 0 0

51526/0x4e6cb5:  mprotect(0x106FEC000, 0x40000, 0x1)             = 0 0

51526/0x4e6cb5:  issetugid(0x0, 0x0, 0x0)                 = 0 0

51526/0x4e6cb5:  mprotect(0x106FEC000, 0x40000, 0x3)             = 0 0

51526/0x4e6cb5:  getentropy(0x7FF7B8F158E0, 0x20, 0x0)           = 0 0

51526/0x4e6cb5:  mprotect(0x106FEC000, 0x40000, 0x1)             = 0 0

51526/0x4e6cb5:  getpid(0x0, 0x0, 0x0)            = 51526 0

51526/0x4e6cb5:  mprotect(0x106FEC000, 0x40000, 0x3)             = 0 0
```

```
51526/0x4e6cb5:  mprotect(0x106FEC000, 0x40000, 0x1)             = 0 0

51526/0x4e6cb5:
getattrlist("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/parent\0",
0x7FF7B8F15D60, 0x7FF7B8F15D78)              = 0 0

51526/0x4e6cb5:  access("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr\0",
0x4, 0x0)                = 0 0

51526/0x4e6cb5:  open("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr\0",
0x0, 0x0)          = 3 0

51526/0x4e6cb5:  fstat64(0x3, 0x7FA15AF04500, 0x0)               = 0 0

51526/0x4e6cb5:  csrctl(0x0, 0x7FF7B8F15FEC, 0x4)                = -1 1

51526/0x4e6cb5:  fgetattrlist(0x3, 0x7FF7B8F16000, 0x7FF7B8F16020)
= 0 0

51526/0x4e6cb5:  __mac_syscall(0x7FF81B4B2719, 0x2, 0x7FF7B8F16020)
= 0 0

51526/0x4e6cb5:  fcntl(0x3, 0x32, 0x7FF7B8F15C90)                = 0 0

51526/0x4e6cb5:  close(0x3)              = 0 0

51526/0x4e6cb5:
open("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/Info.plist\0", 0x0,
0x0)               = -1 2

51526/0x4e6cb5:  proc_info(0x2, 0xC946, 0xD)             = 64 0

51526/0x4e6cb5:  csops_audittoken(0xC946, 0x10, 0x7FF7B8F15F70)          = -1 22

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

51526/0x4e6cb5:  csops(0xC946, 0x0, 0x7FF7B8F163D4)              = 0 0

51526/0x4e6cb5:  sysctlbyname(kern.system_version_compat, 0x1A, 0x0, 0x0,
0x7FF7B8F16404)                = 0 0

51526/0x4e6cb5:  mprotect(0x106FEC000, 0x40000, 0x3)             = 0 0
```

```
51526/0x4e6cb5:  open("test.txt\0", 0x0, 0x0)            = 3 0

51526/0x4e6cb5:  pipe(0x0, 0x0, 0x0)             = 4 0

51526/0x4e6cb5:  fork()          = 51527 0

51527/0x4e6cbd:  fork()          = 0 0

51527/0x4e6cbd:  thread_selfid(0x0, 0x0, 0x0)            = 5139645 0

51526/0x4e6cb5:  close(0x5)              = 0 0

51527/0x4e6cbd:  bsdthread_register(0x7FF810EF5BC4, 0x7FF810EF5BB0, 0x2000)
= -1 22

51527/0x4e6cbd:  mprotect(0x107048000, 0x98, 0x3)                = 0 0

51527/0x4e6cbd:  mprotect(0x107048000, 0x98, 0x1)                = 0 0

51527/0x4e6cbd:  close(0x4)              = 0 0

51527/0x4e6cbd:  dup2(0x3, 0x0, 0x0)             = 0 0

51527/0x4e6cbd:  dup2(0x5, 0x1, 0x0)             = 1 0

dtrace: error on enabled probe ID 1688 (ID 285: syscall::execve:return): invalid
address (0x106fe8f83) in action #12 at DIF offset 12

51527/0x4e6cbe:  fork()          = 0 0

51527/0x4e6cbe:  mprotect(0x11171D000, 0x8000, 0x1)              = 0 0

51527/0x4e6cbe:  thread_selfid(0x0, 0x0, 0x0)            = 5139646 0

51527/0x4e6cbe:  shared_region_check_np(0x7FF7B21AB8D8, 0x0, 0x0)
= 0 0

51527/0x4e6cbe:  thread_selfid(0x0, 0x0, 0x0)            = 5139646 0

51527/0x4e6cbe:  getpid(0x0, 0x0, 0x0)           = 51527 0

51527/0x4e6cbe:  proc_info(0xF, 0xC947, 0x0)             = 0 0

51527/0x4e6cbe:  munmap(0x111681000, 0x9C000)            = 0 0

51527/0x4e6cbe:  munmap(0x11171D000, 0x8000)             = 0 0

51527/0x4e6cbe:  munmap(0x111725000, 0x4000)             = 0 0
```

```
51527/0x4e6cbe:  munmap(0x111729000, 0x4000)             = 0 0

51527/0x4e6cbe:  munmap(0x11172D000, 0x54000)            = 0 0

51527/0x4e6cbe:  open(".\0", 0x100000, 0x0)              = 4 0

51527/0x4e6cbe:  fcntl(0x4, 0x32, 0x7FF7B21AB250)                = 0 0

51527/0x4e6cbe:  close(0x4)              = 0 0

51527/0x4e6cbe:  fsgetpath(0x7FF7B21AB260, 0x400, 0x7FF7B21AB248)
= 57 0

51527/0x4e6cbe:  fsgetpath(0x7FF7B21AB260, 0x400, 0x7FF7B21AB248)
= 14 0

51527/0x4e6cbe:  csrctl(0x0, 0x7FF7B21AB66C, 0x4)                = -1 1

51527/0x4e6cbe:  __mac_syscall(0x7FF810C2E11B, 0x2, 0x7FF7B21AB4E0)
= 0 0

51527/0x4e6cbe:  csrctl(0x0, 0x7FF7B21AB67C, 0x4)                = -1 1

51527/0x4e6cbe:  __mac_syscall(0x7FF810C2B0A8, 0x5A, 0x7FF7B21AB610)
= 0 0

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid
kernel access in action #11 at DIF offset 28

51527/0x4e6cbe:  open("/\0", 0x20100000, 0x0)            = 4 0

51527/0x4e6cbe:  openat(0x4, "System/Cryptexes/OS\0", 0x100000, 0x0)
= 6 0

51527/0x4e6cbe:  dup(0x6, 0x0, 0x0)              = 7 0

51527/0x4e6cbe:  fstatat64(0x6, 0x7FF7B21AA3B1, 0x7FF7B21AA7B0)          = 0 0
```

```
51527/0x4e6cbe:  openat(0x6, "System/Library/dyld/\0", 0x100000, 0x0)
= 8 0

51527/0x4e6cbe:  fcntl(0x8, 0x32, 0x7FF7B21AA440)               = 0 0

51527/0x4e6cbe:  dup(0x8, 0x0, 0x0)             = 9 0

51527/0x4e6cbe:  dup(0x7, 0x0, 0x0)             = 10 0

51527/0x4e6cbe:  close(0x4)             = 0 0

51527/0x4e6cbe:  close(0x7)             = 0 0

51527/0x4e6cbe:  close(0x6)             = 0 0

51527/0x4e6cbe:  close(0x8)             = 0 0

51527/0x4e6cbe:  shared_region_check_np(0x7FF7B21AAD38, 0x0, 0x0)
= 0 0

51527/0x4e6cbe:  fsgetpath(0x7FF7B21AB290, 0x400, 0x7FF7B21AB1C8)
= 83 0

51527/0x4e6cbe:  fcntl(0xA, 0x32, 0x7FF7B21AB290)               = 0 0

51527/0x4e6cbe:  close(0xA)             = 0 0

51527/0x4e6cbe:  close(0x9)             = 0 0

51527/0x4e6cbe:  getfsstat64(0x0, 0x0, 0x2)             = 8 0

51527/0x4e6cbe:  getfsstat64(0x10DD5AA10, 0x43C0, 0x2)          = 8 0

51527/0x4e6cbe:  getattrlist("/\0", 0x7FF7B21AB120, 0x7FF7B21AB090)
= 0 0

51527/0x4e6cbe:  fsgetpath(0x7FF7B21AAF10, 0x400, 0x7FF7B21AAEF8)
= 83 0

51527/0x4e6cbe:
stat64("/System/Volumes/Preboot/Cryptexes/OS/System/Library/dyld/dyld_shared_cac
he_x86_64h\0", 0x7FF7B21AB378, 0x0)             = 0 0

51527/0x4e6cbe:
stat64("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/child\0",
0x7FF7B21AA9A0, 0x0)              = 0 0
```

```
51527/0x4e6cbe:
open("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/child\0", 0x0, 0x0)
= 4 0

51527/0x4e6cbe:  mmap(0x0, 0x3264, 0x1, 0x40002, 0x4, 0x0)               =
0x10DD99000 0

51527/0x4e6cbe:  fcntl(0x4, 0x32, 0x7FF7B21AAAB0)               = 0 0

51527/0x4e6cbe:  close(0x4)              = 0 0

51527/0x4e6cbe:  munmap(0x10DD99000, 0x3264)             = 0 0

51527/0x4e6cbe:
stat64("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/child\0",
0x7FF7B21AAF00, 0x0)               = 0 0

51527/0x4e6cbe:  stat64("/usr/lib/libSystem.B.dylib\0", 0x7FF7B21A9F50, 0x0)
= -1 2

51527/0x4e6cbe:
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/libSystem.B.dylib\0",
0x7FF7B21A9F00, 0x0)                 = -1 2

51527/0x4e6cbe:  stat64("/usr/lib/system/libdispatch.dylib\0", 0x7FF7B21A7B50,
0x0)              = -1 2

51527/0x4e6cbe:
stat64("/System/Volumes/Preboot/Cryptexes/OS/usr/lib/system/libdispatch.dylib\0"
, 0x7FF7B21A7B00, 0x0)          = -1 2

51527/0x4e6cbe:  stat64("/usr/lib/system/libdispatch.dylib\0", 0x7FF7B21A7B50,
0x0)              = -1 2

51527/0x4e6cbe:  open("/dev/dtracehelper\0", 0x2, 0x0)           = 4 0

51527/0x4e6cbe:  ioctl(0x4, 0x80086804, 0x7FF7B21A9B58)          = 0 0

51527/0x4e6cbe:  close(0x4)              = 0 0

51527/0x4e6cbe:  mprotect(0x10DD55000, 0x1000, 0x1)              = 0 0

51527/0x4e6cbe:  shared_region_check_np(0xFFFFFFFFFFFFFFFF, 0x0, 0x0)
= 0 0

51527/0x4e6cbe:  mprotect(0x10DD58000, 0x40000, 0x1)             = 0 0
```

```
51527/0x4e6cbe:  access("/AppleInternal/XBS/.isChrooted\0", 0x0, 0x0)
= -1 2

51527/0x4e6cbe:  bsdthread_register(0x7FF810EF5BC4, 0x7FF810EF5BB0, 0x2000)
= 1073742303 0

51527/0x4e6cbe:  shm_open(0x7FF810DA0F5A, 0x0, 0x10D9F465)              = 4 0

51527/0x4e6cbe:  fstat64(0x4, 0x7FF7B21A9DA0, 0x0)              = 0 0

51527/0x4e6cbe:  mmap(0x0, 0x3000, 0x1, 0x40001, 0x4, 0x0)              =
0x10DD9B000 0

51527/0x4e6cbe:  close(0x4)             = 0 0

51527/0x4e6cbe:  ioctl(0x2, 0x4004667A, 0x7FF7B21A9E54)         = 0 0

51527/0x4e6cbe:  mprotect(0x10DDA3000, 0x1000, 0x0)             = 0 0

51527/0x4e6cbe:  mprotect(0x10DDAA000, 0x1000, 0x0)             = 0 0

51527/0x4e6cbe:  mprotect(0x10DDAB000, 0x1000, 0x0)             = 0 0

51527/0x4e6cbe:  mprotect(0x10DDB2000, 0x1000, 0x0)             = 0 0

51527/0x4e6cbe:  mprotect(0x10DD9E000, 0x98, 0x1)               = 0 0

51527/0x4e6cbe:  mprotect(0x10DD9E000, 0x98, 0x3)               = 0 0

51527/0x4e6cbe:  mprotect(0x10DD9E000, 0x98, 0x1)               = 0 0

51527/0x4e6cbe:  mprotect(0x10DDB3000, 0x1000, 0x1)             = 0 0

51527/0x4e6cbe:  mprotect(0x10DDB4000, 0x98, 0x1)               = 0 0

51527/0x4e6cbe:  mprotect(0x10DDB4000, 0x98, 0x3)               = 0 0

51527/0x4e6cbe:  mprotect(0x10DDB4000, 0x98, 0x1)               = 0 0

51527/0x4e6cbe:  mprotect(0x10DD9E000, 0x98, 0x3)               = 0 0

51527/0x4e6cbe:  mprotect(0x10DD9E000, 0x98, 0x1)               = 0 0

51527/0x4e6cbe:  mprotect(0x10DDB3000, 0x1000, 0x3)             = 0 0

51527/0x4e6cbe:  mprotect(0x10DDB3000, 0x1000, 0x1)             = 0 0

51527/0x4e6cbe:  mprotect(0x10DD58000, 0x40000, 0x3)            = 0 0
```

```
51527/0x4e6cbe:  mprotect(0x10DD58000, 0x40000, 0x1)             = 0 0

51527/0x4e6cbe:  issetugid(0x0, 0x0, 0x0)                 = 0 0

51527/0x4e6cbe:  mprotect(0x10DD58000, 0x40000, 0x3)             = 0 0

51527/0x4e6cbe:  getentropy(0x7FF7B21A9900, 0x20, 0x0)           = 0 0

51527/0x4e6cbe:  mprotect(0x10DD58000, 0x40000, 0x1)             = 0 0

51527/0x4e6cbe:  getpid(0x0, 0x0, 0x0)            = 51527 0

51527/0x4e6cbe:  mprotect(0x10DD58000, 0x40000, 0x3)             = 0 0

51527/0x4e6cbe:  mprotect(0x10DD58000, 0x40000, 0x1)             = 0 0

51527/0x4e6cbe:
getattrlist("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/child\0",
0x7FF7B21A9D80, 0x7FF7B21A9D98)               = 0 0

51527/0x4e6cbe:  access("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr\0",
0x4, 0x0)                = 0 0

51527/0x4e6cbe:  open("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr\0",
0x0, 0x0)          = 4 0

51527/0x4e6cbe:  fstat64(0x4, 0x7FD45B704500, 0x0)               = 0 0

51527/0x4e6cbe:  csrctl(0x0, 0x7FF7B21AA00C, 0x4)                = -1 1

51527/0x4e6cbe:  fgetattrlist(0x4, 0x7FF7B21AA020, 0x7FF7B21AA040)
= 0 0

51527/0x4e6cbe:  __mac_syscall(0x7FF81B4B2719, 0x2, 0x7FF7B21AA040)
= 0 0

51527/0x4e6cbe:  fcntl(0x4, 0x32, 0x7FF7B21A9CB0)                = 0 0

51527/0x4e6cbe:  close(0x4)              = 0 0

51527/0x4e6cbe:
open("/Users/mishazhadnov/Desktop/wD/OS_labs_3t/lab1/scr/Info.plist\0", 0x0,
0x0)               = -1 2

51527/0x4e6cbe:  proc_info(0x2, 0xC947, 0xD)             = 64 0

51527/0x4e6cbe:  csops_audittoken(0xC947, 0x10, 0x7FF7B21A9F90)          = -1 22
```

```
dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid kernel access in action #11 at DIF offset 28

dtrace: error on enabled probe ID 1741 (ID 571: syscall::sysctl:return): invalid kernel access in action #11 at DIF offset 28

51527/0x4e6cbe:  csops(0xC947, 0x0, 0x7FF7B21AA3F4)              = 0 0

51527/0x4e6cbe:  sysctlbyname(kern.system_version_compat, 0x1A, 0x0, 0x0, 0x7FF7B21AA424)               = 0 0

51527/0x4e6cbe:  mprotect(0x10DD58000, 0x40000, 0x3)             = 0 0

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1711 (ID 175: syscall::write:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid kernel access in action #13 at DIF offset 68
```

```
dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1711 (ID 175: syscall::write:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1711 (ID 175: syscall::write:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68
```

```
dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1711 (ID 175: syscall::write:return): invalid
kernel access in action #13 at DIF offset 68

51526/0x4e6cb5:  wait4(0xFFFFFFFFFFFFFFFF, 0x0, 0x0)             = 51527 0

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

51526/0x4e6cb5:  getrlimit(0x1008, 0x7FF7B8F17380, 0x0)          = 0 0

51526/0x4e6cb5:  fstat64(0x1, 0x7FF7B8F17368, 0x0)               = 0 0

51526/0x4e6cb5:  ioctl(0x1, 0x4004667A, 0x7FF7B8F173B4)          = 0 0

dtrace: error on enabled probe ID 1712 (ID 961: syscall::write_nocancel:return):
invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1712 (ID 961: syscall::write_nocancel:return):
invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1712 (ID 961: syscall::write_nocancel:return):
invalid kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1713 (ID 173: syscall::read:return): invalid
kernel access in action #13 at DIF offset 68

dtrace: error on enabled probe ID 1712 (ID 961: syscall::write_nocancel:return):
invalid kernel access in action #13 at DIF offset 68
```

Вывод

Благодаря данной лабораторной работе я на практике изучил принципы работы с неименованными каналами для межпроцессного взаимодействия, разобрался, как перенаправлять потоки ввода/вывода, а также научился использовать системные вызовы и обращаться с файловыми дескрипторами (которые важно вовремя и уместно закрывать).

Очевидно, что в реальных, “рабочих” программах используется большее количество неименованных каналов и процессов. Эта лабораторная работа научила базовому обращению с ними.