Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 “Компьютерные науки и прикладная математика”

Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №1 по курсу

«Операционные системы»

Группа: М8О-210Б-23

Студент: Нугаев М. Э.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 30.11.24

Москва, 2024

Постановка задачи

Вариант 20:

Пользователь вводит строки». Далее эти строки передаются от родительского процесса в дочерний. Дочерний процесс конвертирует эти строки и затем записывает полученный результат в файл.

Общий метод и алгоритм решения

Использованные системные вызовы:

- pid_t fork(void); – создает дочерний процесс.
- int pipe(int *fd); – создаёт пайп и помещает дескрипторы в fd[0], fd[1], для чтения и записи.
- ssize_t read(int fd, void* buff, int count); - считывает по дескриптору fd count байт из buff.
- ssize_t write(int fd, const void* buff, int count); – записывает по дескриптору fd count байт из buff.
- void exit(int number); – вызывает нормальное завершение программы с кодом number.
- int dup2(int fd1, int fd2); – делает эквиваентными дескрипторы fd1 и fd2.
- int execl(const char* path, const char* argc); – заменяет текущий процесс на процесс path, с аргументами argc;
- int close(int fd); – закрывает дескриптор fd.
- pid_t wait(int *status) — функция, которая приостанавливает выполнение текущего процесса до тех пор, пока дочерний процесс не завершится,

Я создал два файла parent и child.

Программа создает неименованный канал (pipe) для передачи данных между родительским и дочерним процессами. Она запрашивает у пользователя ввод имени файла, после чего создаёт дочерний процесс с помощью fork(). Дочерний процесс перенаправляет стандартный ввод на чтение из канала и запускает другую программу (дочернюю программу), передавая ей имя файла. Родительский процесс читает данные из стандартного ввода до тех пор, пока пользователь не остановит ввод и записывает эти данные в канал. После завершения записи родительский процесс закрывает канал и ожидает завершения дочернего процесса.

В файле child я обрабатываю получнные из родительского процесса данные и записываю их в файл.

Код программы

Parent.c

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

#define BUFFER_SIZE 256
```

```c
int main() {
    int pipe1[2], pipe2[2];
    char filename1[BUFFER_SIZE];
    char filename2[BUFFER_SIZE];
    ssize_t bytes_read;
    const char *prompt1 = "Enter the file name for child1: ";
    write(STDOUT_FILENO, prompt1, strlen(prompt1));
    bytes_read = read(STDIN_FILENO, filename1, BUFFER_SIZE - 1);
    filename1[bytes_read - 1] = '\0';
    const char *prompt2 = "Enter the file name for child2: ";
    write(STDOUT_FILENO, prompt2, strlen(prompt2));
    bytes_read = read(STDIN_FILENO, filename2, BUFFER_SIZE - 1);
    filename2[bytes_read - 1] = '\0';
    int file_check1 = open(filename1, O_WRONLY | O_CREAT | O_TRUNC, 0666);
    if (file_check1 == -1) {
        const char *error_msg = "Error: could not open the file for child1\n";
        write(STDERR_FILENO, error_msg, strlen(error_msg));
        perror("open");
        return 1;
    }
    close(file_check1);
    int file_check2 = open(filename2, O_WRONLY | O_CREAT | O_TRUNC, 0666);
    if (file_check2 == -1) {
        const char *error_msg = "Error: could not open the file for child2\n";
        write(STDERR_FILENO, error_msg, strlen(error_msg));
        perror("open");
        return 1;
    }
    close(file_check2);
    if (pipe(pipe1) == -1 || pipe(pipe2) == -1) {
        const char *error_msg = "Error during creation pipe\n";
        write(STDERR_FILENO, error_msg, strlen(error_msg));
        return 1;
    }
    pid_t pid1 = fork();
    if (pid1 == -1) {
        const char *error_msg = "Error when creating a child process 1\n";
        write(STDERR_FILENO, error_msg, strlen(error_msg));
        return 1;
    } else if (pid1 == 0) {
        close(pipe1[1]);
        dup2(pipe1[0], STDIN_FILENO);
        execl("./child.out", "./child.out", filename1, NULL);
        const char *error_msg = "Error when starting child1\n";
        write(STDERR_FILENO, error_msg, strlen(error_msg));
        exit(1);
    }
    pid_t pid2 = fork();
    if (pid2 == -1) {
        const char *error_msg = "Error when creating a child process 2\n";
        write(STDERR_FILENO, error_msg, strlen(error_msg));
        return 1;
    } else if (pid2 == 0) {
        close(pipe2[1]);
```

```
        dup2(pipe2[0], STDIN_FILENO);
        execl("./child.out", "./child.out", filename2, NULL);
        const char *error_msg = "Error when starting child2\n";
        write(STDERR_FILENO, error_msg, strlen(error_msg));
        exit(1);
    }
    close(pipe1[0]);
    close(pipe2[0]);
    char input[BUFFER_SIZE];
    int randomInt = rand() % 5 + 1;
    while (1) {
        bytes_read = read(STDIN_FILENO, input, sizeof(input) - 1);
        if (bytes_read <= 0) {
            const char *error_msg = "Error while reading the string\n";
            write(STDERR_FILENO, error_msg, strlen(error_msg));
            break;
        }
        if (bytes_read > 0 && input[bytes_read - 1] == '\n') {
            input[bytes_read - 1] = '\0';
        }
        if (strcmp(input, "exit") == 0) {
            write(pipe2[1], input, strlen(input) + 1);
            write(pipe1[1], input, strlen(input) + 1);
            break;
        }
        if (strlen(input) > 10) {
            write(pipe2[1], input, strlen(input) + 1);
        } else {
            write(pipe1[1], input, strlen(input) + 1);
        }
    }
    close(pipe1[1]);
    close(pipe2[1]);
    waitpid(pid1, NULL, 0);
    waitpid(pid2, NULL, 0);

    return 0;
}
```

Child.c

```
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define BUFFER_SIZE 256

bool is_vowel(char c) {
    c = (c >= 'A' && c <= 'Z') ? c + 32 : c;
```

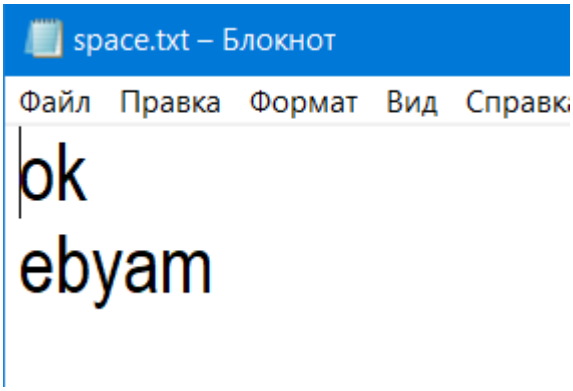
```c
    return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';
}

void invert_string(char *str) {
    int len = strlen(str);
    for (int i = 0; i < len / 2; i++) {
        char temp = str[i];
        str[i] = str[len - i - 1];
        str[len - i - 1] = temp;
    }
}

int main(int argc, char *argv[]) {
    if (argc < 2) {
        const char *error_msg = "Error: The file name was not passed\n";
        write(STDERR_FILENO, error_msg, strlen(error_msg));
        return 1;
    }
    int file = open(argv[1], O_WRONLY | O_CREAT | O_TRUNC, 0666);
    if (file == -1) {
        const char *error_msg = "Error: the file could not be opened for writing\n";
        write(STDERR_FILENO, error_msg, strlen(error_msg));
        perror("open");
        return 1;
    }
    char buffer[BUFFER_SIZE];
    ssize_t bytes_read;
    while ((bytes_read = read(STDIN_FILENO, buffer, sizeof(buffer) - 1)) > 0) {
        buffer[bytes_read - 1] = '\0';
        if(strcmp(buffer, "exit") == 0){
            close(file);
            return 0;
        }
        invert_string(buffer);
        if (write(file, buffer, strlen(buffer)) == -1 || write(file, "\n", 1) == -1) {
            const char *error_msg = "Error: failed to write to a file\n";
            write(STDERR_FILENO, error_msg, strlen(error_msg));
            close(file);
            return 1;
        }
    }
    close(file);
    return 0;
}
```

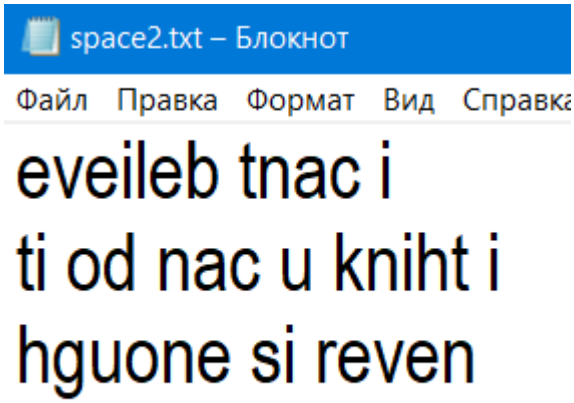
Протокол работы программы

Тестирование:

```
miron@DESKTOP-GD72A05:~/LABS/lab 1$ ./parent
Введите имя файла для child1: space.txt
Введите имя файла для child2: space2.txt
Введите строку: i cant believe
Введите строку: ko
Введите строку: maybe
Введите строку: i think u can do it
Введите строку: never is enough
Введите строку: ^C
miron@DESKTOP-GD72A05:~/LABS/lab 1$
```

space.txt	20.11.2024 22:10	Текстовый докум...	1 КБ
space2.txt	20.11.2024 22:10	Текстовый докум...	1 КБ





Strace:

strace -f ./parent

execve("./parent", ["./parent"], 0x7ffdf77e7fe8 /* 26 vars */) = 0

brk(NULL) = 0x55e1a7b08000

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffcd35234f0) = -1 EINVAL (Invalid argument)

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

fstat(3, {st_mode=S_IFREG|0644, st_size=33395, ...}) = 0

mmap(NULL, 33395, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f205384c000

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\0\300A\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\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357_\243\335\2449\206V>\237\374\304"..., 68, 880) = 68

fstat(3, {st_mode=S_IFREG|0755, st_size=2029592, ...}) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f205384a000

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\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357_\243\335\2449\206V>\237\374\304"..., 68, 880) = 68

mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f2053658000

mmap(0x7f205367a000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = 0x7f205367a000

mmap(0x7f20537f2000, 319488, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19a000) = 0x7f20537f2000

mmap(0x7f2053840000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f2053840000

mmap(0x7f2053846000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f2053846000

close(3) = 0

arch_prctl(ARCH_SET_FS, 0x7f205384b540) = 0

mprotect(0x7f2053840000, 16384, PROT_READ) = 0

mprotect(0x55e1a6d37000, 4096, PROT_READ) = 0

mprotect(0x7f2053882000, 4096, PROT_READ) = 0

munmap(0x7f205384c000, 33395) = 0

write(1, "Enter the file name for child1: ", 32Enter the file name for child1:) = 32

read(0, file1.txt

"file1.txt\n", 255) = 10

write(1, "Enter the file name for child2: ", 32Enter the file name for child2:) = 32

read(0, file2.txt

"file2.txt\n", 255) = 10

openat(AT_FDCWD, "file1.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3

close(3) = 0

openat(AT_FDCWD, "file2.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3

close(3) = 0

pipe([3, 4]) = 0

pipe([5, 6]) = 0

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace: Process 818 attached

, child_tidptr=0x7f205384b810) = 818

[pid 817] clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD <unfinished ...>

[pid 818] close(4strace: Process 819 attached

) = 0

[pid 817] <... clone resumed>, child_tidptr=0x7f205384b810) = 819

[pid 819] close(6 <unfinished ...>

[pid 817] close(3 <unfinished ...>

[pid 818] dup2(3, 0 <unfinished ...>

[pid 817] <... close resumed>) = 0

[pid 819] <... close resumed>) = 0

[pid 817] close(5 <unfinished ...>

[pid 818] <... dup2 resumed>) = 0

[pid 817] <... close resumed>) = 0

[pid 819] dup2(5, 0 <unfinished ...>

[pid 817] read(0, <unfinished ...>

[pid 818] execve("./child.out", ["./child.out", "file1.txt"], 0x7ffcd35235d8 /* 26 vars */ <unfinished ...>

[pid 819] <... dup2 resumed>) = 0

[pid 819] execve("./child.out", ["./child.out", "file2.txt"], 0x7ffcd35235d8 /* 26 vars */ <unfinished ...>

[pid 818] <... execve resumed>) = 0

[pid 818] brk(NULL <unfinished ...>

[pid 819] <... execve resumed>) = 0

[pid 818] <... brk resumed>) = 0x5562da561000

[pid 819] brk(NULL <unfinished ...>

[pid 818] arch_prctl(0x3001 /* ARCH_??? */, 0x7fff251ab140 <unfinished ...>

[pid 819] <... brk resumed>) = 0x55ee16774000

[pid 818] <... arch_prctl resumed>) = -1 EINVAL (Invalid argument)

[pid 819] arch_prctl(0x3001 /* ARCH_??? */, 0x7fff05783250) = -1 EINVAL (Invalid argument)

[pid 818] access("/etc/ld.so.preload", R_OK <unfinished ...>

[pid 819] access("/etc/ld.so.preload", R_OK <unfinished ...>

[pid 818] <... access resumed>) = -1 ENOENT (No such file or directory)

[pid 819] <... access resumed>) = -1 ENOENT (No such file or directory)

[pid 818] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 819] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 818] <... openat resumed>) = 4

[pid 819] <... openat resumed>) = 6

[pid 818] fstat(4, <unfinished ...>

[pid 819] fstat(6, <unfinished ...>

[pid 818] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=33395, ...}) = 0

[pid 819] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=33395, ...}) = 0

[pid 818] mmap(NULL, 33395, PROT_READ, MAP_PRIVATE, 4, 0 <unfinished ...>

[pid 819] mmap(NULL, 33395, PROT_READ, MAP_PRIVATE, 6, 0 <unfinished ...>

[pid 818] <... mmap resumed>) = 0x7fa8b8184000

[pid 819] <... mmap resumed>) = 0x7f1572e73000

[pid 818] close(4 <unfinished ...>

[pid 819] close(6 <unfinished ...>

[pid 818] <... close resumed>) = 0

[pid 819] <... close resumed>) = 0

[pid 818] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 819] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 818] <... openat resumed>) = 4

[pid 819] <... openat resumed>) = 6

[pid 818] read(4, <unfinished ...>

[pid 819] read(6, <unfinished ...>

[pid 818] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0\0"..., 832) = 832

[pid 819] <... read resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\300A\2\0\0\0\0\0"..., 832) = 832

[pid 818] pread64(4, <unfinished ...>

[pid 819] pread64(6, <unfinished ...>

[pid 818] <... pread64 resumed>"\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 819] <... pread64 resumed>"\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 818] pread64(4, <unfinished ...>

[pid 819] pread64(6, <unfinished ...>

[pid 818] <... pread64 resumed>"\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

[pid 819] <... pread64 resumed>"\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

[pid 818] pread64(4, <unfinished ...>

[pid 819] pread64(6, <unfinished ...>

[pid 818] <... pread64 resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357_\243\335\2449\206V>\237\374\304"..., 68, 880) = 68

[pid 819] <... pread64 resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357_\243\335\2449\206V>\237\374\304"..., 68, 880) = 68

[pid 818] fstat(4, <unfinished ...>

[pid 819] fstat(6, <unfinished ...>

[pid 818] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=2029592, ...}) = 0

[pid 819] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=2029592, ...}) = 0

[pid 818] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 819] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 818] <... mmap resumed>) = 0x7fa8b8182000

[pid 819] <... mmap resumed>) = 0x7f1572e71000

[pid 818] pread64(4, <unfinished ...>

[pid 819] pread64(6, <unfinished ...>

[pid 818] <... pread64 resumed>"\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 819] <... pread64 resumed>"\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 818] pread64(4, <unfinished ...>

[pid 819] pread64(6, <unfinished ...>

[pid 818] <... pread64 resumed>"\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

[pid 819] <... pread64 resumed>"\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

[pid 818] pread64(4, <unfinished ...>

[pid 819] pread64(6, <unfinished ...>

[pid 818] <... pread64 resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357_\243\335\2449\206V>\237\374\304"..., 68, 880) = 68

[pid 819] <... pread64 resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\7\2C\n\357_\243\335\2449\206V>\237\374\304"..., 68, 880) = 68

[pid 818] mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 4, 0 <unfinished ...>

[pid 819] mmap(NULL, 2037344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 6, 0 <unfinished ...>

[pid 818] <... mmap resumed>) = 0x7fa8b7f90000

[pid 819] <... mmap resumed>) = 0x7f1572c7f000

[pid 818] mmap(0x7fa8b7fb2000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x22000 <unfinished ...>

[pid 819] mmap(0x7f1572ca1000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 0x22000 <unfinished ...>

[pid 818] <... mmap resumed>) = 0x7fa8b7fb2000

[pid 819] <... mmap resumed>) = 0x7f1572ca1000

[pid 818] mmap(0x7fa8b812a000, 319488, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x19a000 <unfinished ...>

[pid 819] mmap(0x7f1572e19000, 319488, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 0x19a000) = 0x7f1572e19000

[pid 818] <... mmap resumed>) = 0x7fa8b812a000

[pid 819] mmap(0x7f1572e67000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 0x1e7000 <unfinished ...>

[pid 818] mmap(0x7fa8b8178000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x1e7000 <unfinished ...>

[pid 819] <... mmap resumed>) = 0x7f1572e67000

[pid 818] <... mmap resumed>) = 0x7fa8b8178000

[pid 819] mmap(0x7f1572e6d000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 818] mmap(0x7fa8b817e000, 13920, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>

[pid 819] <... mmap resumed>) = 0x7f1572e6d000

[pid 818] <... mmap resumed>) = 0x7fa8b817e000

[pid 819] close(6 <unfinished ...>

[pid 818] close(4 <unfinished ...>

[pid 819] <... close resumed>) = 0

[pid 818] <... close resumed>) = 0

[pid 819] arch_prctl(ARCH_SET_FS, 0x7f1572e72540) = 0

[pid 818] arch_prctl(ARCH_SET_FS, 0x7fa8b8183540 <unfinished ...>

[pid 819] mprotect(0x7f1572e67000, 16384, PROT_READ <unfinished ...>

[pid 818] <... arch_prctl resumed>) = 0

[pid 819] <... mprotect resumed>) = 0

[pid 819] mprotect(0x55ee15975000, 4096, PROT_READ <unfinished ...>

[pid 818] mprotect(0x7fa8b8178000, 16384, PROT_READ <unfinished ...>

[pid 819] <... mprotect resumed>) = 0

[pid 818] <... mprotect resumed>) = 0

[pid 819] mprotect(0x7f1572ea9000, 4096, PROT_READ <unfinished ...>

[pid 818] mprotect(0x5562da4cf000, 4096, PROT_READ <unfinished ...>

[pid 819] <... mprotect resumed>) = 0

[pid 818] <... mprotect resumed>) = 0

[pid 819] munmap(0x7f1572e73000, 33395 <unfinished ...>

[pid 818] mprotect(0x7fa8b81ba000, 4096, PROT_READ <unfinished ...>

[pid 819] <... munmap resumed>) = 0

[pid 818] <... mprotect resumed>) = 0

[pid 819] openat(AT_FDCWD, "file2.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666 <unfinished ...>

[pid 818] munmap(0x7fa8b8184000, 33395 <unfinished ...>

[pid 819] <... openat resumed>) = 6

[pid 818] <... munmap resumed>) = 0

[pid 819] read(0, <unfinished ...>

[pid 818] openat(AT_FDCWD, "file1.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 4

[pid 818] read(0, string

<unfinished ...>

[pid 817] <... read resumed>"string\n", 255) = 7

[pid 817] write(4, "string\0", 7) = 7

[pid 818] <... read resumed>"string\0", 255) = 7

[pid 817] read(0, <unfinished ...>

[pid 818] write(4, "gnirts", 6) = 6

[pid 818] write(4, "\n", 1) = 1

[pid 818] read(0, one more string

<unfinished ...>

[pid 817] <... read resumed>"one more string\n", 255) = 16

[pid 817] write(6, "one more string\0", 16) = 16

[pid 819] <... read resumed>"one more string\0", 255) = 16

[pid 817] read(0, <unfinished ...>

[pid 819] write(6, "gnirts erom eno", 15) = 15

[pid 819] write(6, "\n", 1) = 1

[pid 819] read(0, one more more string

<unfinished ...>

[pid 817] <... read resumed>"one more more string\n", 255) = 21

[pid 817] write(6, "one more more string\0", 21) = 21

[pid 819] <... read resumed>"one more more string\0", 255) = 21

[pid 817] read(0, <unfinished ...>

[pid 819] write(6, "gnirts erom erom eno", 20) = 20

[pid 819] write(6, "\n", 1) = 1

[pid 819] read(0, one

<unfinished ...>

[pid 817] <... read resumed>"one\n", 255) = 4

[pid 817] write(4, "one\0", 4) = 4

[pid 818] <... read resumed>"one\0", 255) = 4

[pid 817] read(0, <unfinished ...>

[pid 818] write(4, "eno", 3) = 3

[pid 818] write(4, "\n", 1) = 1

[pid 818] read(0, more

<unfinished ...>

[pid 817] <... read resumed>"more\n", 255) = 5

[pid 817] write(4, "more\0", 5) = 5

[pid 818] <... read resumed>"more\0", 255) = 5

[pid 817] read(0, <unfinished ...>

[pid 818] write(4, "erom", 4) = 4

[pid 818] write(4, "\n", 1) = 1

[pid 818] read(0, 43124321432154154

<unfinished ...>

[pid 817] <... read resumed>"43124321432154154\n", 255) = 18

[pid 817] write(6, "43124321432154154\0", 18) = 18

[pid 819] <... read resumed>"43124321432154154\0", 255) = 18

[pid 817] read(0, <unfinished ...>

[pid 819] write(6, "45145123412342134", 17) = 17

[pid 819] write(6, "\n", 1) = 1

[pid 819] read(0, 321321312

<unfinished ...>

[pid 817] <... read resumed>"321321312\n", 255) = 10

[pid 817] write(4, "321321312\0", 10) = 10

[pid 818] <... read resumed>"321321312\0", 255) = 10

[pid 817] read(0, <unfinished ...>

[pid 818] write(4, "213123123", 9) = 9

[pid 818] write(4, "\n", 1) = 1

[pid 818] read(0, exit

<unfinished ...>

[pid 817] <... read resumed>"exit\n", 255) = 5

[pid 817] write(6, "exit\0", 5) = 5

[pid 819] <... read resumed>"exit\0", 255) = 5

[pid 817] write(4, "exit\0", 5 <unfinished ...>

[pid 819] close(6 <unfinished ...>

[pid 817] <... write resumed>) = 5

[pid 819] <... close resumed>) = 0

[pid 817] close(4 <unfinished ...>

[pid 818] <... read resumed>"exit\0", 255) = 5

[pid 817] <... close resumed>) = 0

[pid 819] exit_group(0 <unfinished ...>

[pid 817] close(6 <unfinished ...>

[pid 818] close(4 <unfinished ...>

[pid 817] <... close resumed>) = 0

[pid 819] <... exit_group resumed>) = ?

[pid 817] wait4(818, <unfinished ...>

[pid 818] <... close resumed>) = 0

[pid 819] +++ exited with 0 +++

[pid 818] exit_group(0 <unfinished ...>

[pid 817] <... wait4 resumed>NULL, 0, NULL) = ? ERESTARTSYS (To be restarted if SA_RESTART is set)

[pid 818] <... exit_group resumed>) = ?

[pid 817] --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=819, si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---

[pid 817] wait4(818, <unfinished ...>

[pid 818] +++ exited with 0 +++

<... wait4 resumed>NULL, 0, NULL) = 818

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=818, si_uid=1000, si_status=0, si_utime=0, si_stime=0} ---

wait4(819, NULL, 0, NULL) = 819

exit_group(0) = ?

+++ exited with 0 +++

Вывод

Было интересно решать лабораторную работу. Я научился использовать некоторые системные вызовы, а также обмениваться данными между процессами с помощью каналов. Было интересно узнать, как можно писать программы используя их. Возникли трудности с обработкой всех ошибок системных вызовов в программе.