

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-211Б-23

Студент: Бугренков В.П.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 06.11.24

Москва, 2024

Постановка задачи

Вариант 9.

Цель работы

Приобретение практических навыков в:

- Управление процессами в ОС
- Обеспечение обмена данных между процессами посредством каналов

Задание

Составить и отладить программу на языке Си, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. В результате работы программа (основной процесс) должен создать для решение задачи один или несколько дочерних процессов. Взаимодействие между процессами осуществляется через системные сигналы/события и/или каналы (pipe). Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы.

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 float. 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 однонаправленный канал для обмена данными между процессами.
- `int dup2(int oldfd, int newfd)`; - переназначение файлового дескриптора.
- `ssize_t write(int fd, const void *buf, size_t count)`; - записывает count байт из буфера в файл.
- `ssize_t read(int fd, void *buf, size_t count)`; - читает до count байт из файла
- `int open(const char *pathname, int flags, mode_t mode)`; - открытие или создание файла.
- `int close(int fd)`; — закрытие файла, связанного с файловым дескриптором fd.
- `void exit(int status)`; — завершение выполнения процесса с возвратом кода status.
- `int execl(const char *filename, char *const argv[])`; — замена образа памяти текущего процесса программой, указанной в filename.

Главная программа `parent.c` открывает файл для чтения, путь к которому передается первым аргументом, создает канал (pipe) для передачи данных между процессами и запускает дочерний процесс с помощью `fork()`. Родительский процесс закрывает конец канала для чтения и начинает читать строки из файла, передавая их в канал через его левый конец (`pipe[1]`). В то же время дочерний процесс закрывает конец канала для записи, перенаправляет стандартный ввод (`STDIN_FILENO`) на чтение из канала с помощью `dup2(pipe[0], STDIN_FILENO)` и читает строки, поступающие через `STDIN`. После получения строки дочерний процесс разбивает её на отдельные числа, проверяет корректность каждого числа, делит их и выводит результат через стандартный вывод (`STDOUT_FILENO`).

Код программы

parent.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <fcntl.h>
5  #include <string.h>
6  #include <sys/wait.h>
7
8  #define BUFFER_SIZE 512
9
10 /* ERRORS_EXIT_CODES или enum Errors - КОДЫ ВОЗВРАТА ДЛЯ ФУНКЦИЙ */
11 typedef enum Errors
12 {
13     E_SUCCESS = 0,          /* Успешное завершение */
14     E_INVALID_INPUT,        /* Ошибка: Некорректный ввод */
15     E_NOT_ENOUGH_PARAMS,    /* Ошибка: Отсутствует аргумент */
16     E_INVALID_ARG,          /* Ошибка: Неправильный аргумент */
17     E_TYPE_OVERFLOW,        /* Ошибка: Недостаточно памяти для записи значения */
18     E_MEMORY_ALLOCATION,     /* Ошибка: Выделении памяти */
19     E_CANNOT_OPEN_FILE,     /* Ошибка: ОТКРЫТИЯ ФАЙЛА */
20     E_INVALID_EPSILON,      /* Ошибка: Невалидного числа эpsilon */
21     E_DEREFENCE_NULL_POINTER, /* Ошибка: Попытка разыменовать Dereference NULL */
22     E_INT_OVERFLOW,         /* Ошибка: Переполнение типа INT */
23     E_LONG_OVERFLOW,        /* Ошибка: Переполнение типа LONG INT */
24     E_FLOAT_OVERFLOW,       /* Ошибка: Переполнение типа FLOAT */
25     E_DOUBLE_OVERFLOW,      /* Ошибка: Переполнение типа DOUBLE */
26     E_LONG_DOUBLE_OVERFLOW, /* Ошибка: Переполнение типа DOUBLE */
27     E_INVALID_FLAG_ARG,     /* Ошибка: Неккоректный ввод аргументов */
28     E_SAME_FILE_NAMES,      /* Ошибка: Одинаковые имена файлов */
29     E_BUFFER_OVERFLOW,      /* Ошибка: Переполнение буфера */
30     E_SAME_FILES,           /* Ошибка: Ввод однокровых файлов */
31     E_FALSE,                /* BOOL_FALSE */
32 } ERRORS_EXIT_CODES;
33
34 void write_error(const char *error_string)
35 {
36     if (error_string == NULL)
37     {
38         write(STDERR_FILENO, "ERROR", 6);
39     }
40     write(STDERR_FILENO, error_string, strlen(error_string));
41 }
42
43 int main(int args, char *argv[])
44 {
45     if (args != 2)
46     {
47         write_error("ERROR: User: where is a filename????");
48         exit(EXIT_FAILURE);
49     }
50
51     FILE *file = fopen(argv[1], "r");
52     if (file == NULL)
53     {
54         write_error("ERROR: cannot open file");
55         exit(EXIT_FAILURE);
56     }
57
58     int fd[2];
```

```

61  /*
62  fd[0] read
63  fd[1] write
64  */
65  if (pipe(fd) == -1)
66  {
67      write_error("ERROR: Pipe failed");
68      exit(EXIT_FAILURE);
69  }
70
71  pid_t pid = fork();
72  if (pid < 0)
73  {
74      write_error("ERROR: Fork failed");
75      exit(EXIT_FAILURE);
76  }
77
78  if (pid == 0)
79  {
80      // CHILD
81      close(fd[1]);
82
83      dup2(fd[0], STDIN_FILENO);
84      close(fd[0]);
85
86      execl("./child", "", NULL);
87
88      write_error("ERROR: execl failed, use gcc child.c with flag -o child");
89      exit(EXIT_FAILURE);
90  }
91
92  else
93  {
94      // PARENT
95      close(fd[0]);
96
97      char file_buffer[BUFFER_SIZE];
98      while (fgets(file_buffer, sizeof(file_buffer), file) != NULL)
99      {
100          write(fd[1], file_buffer, strlen(file_buffer));
101      }
102      close(fd[1]);
103
104      fclose(file);
105      wait(NULL);
106  }
107
108  return E_SUCCESS;
109 }

```

child.c:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <ctype.h>
5  #include "string.h"
6  #include <limits.h>
7  #include <float.h>
8  #include <math.h>
9  #include <float.h>
10
11 #define BUFFER_SIZE 512

```

```

12
13 /* ERRORS_EXIT_CODES или enum Errors - КОДЫ ВОЗВРАТА ДЛЯ ФУНКЦИЙ */
14 typedef enum Errors
15 {
16     E_SUCCESS = 0,          /* Успешное завершение */
17     E_INVALID_INPUT,        /* Ошибка: Некорректный ввод */
18     E_NOT_ENOUGH_PARAMS,     /* Ошибка: Отсутствует аргумент */
19     E_INVALID_ARG,          /* Ошибка: Неправильный аргумент */
20     E_TYPE_OVERFLOW,        /* Ошибка: Недостаточно памяти для записи значения */
21     E_MEMORY_ALLOCATION,      /* Ошибка: Выделении памяти */
22     E_CANNOT_OPEN_FILE,     /* Ошибка: ОТКРЫТИЯ ФАЙЛА */
23     E_INVALID_EPSILON,      /* Ошибка: Невалидного числа эpsilon */
24     E_DEREFENCE_NULL_POINTER, /* Ошибка: Попытка разыменовать Dereference NULL */
25     E_INT_OVERFLOW,         /* Ошибка: Переполнение типа INT */
26     E_LONG_OVERFLOW,        /* Ошибка: Переполнение типа LONG INT */
27     E_FLOAT_OVERFLOW,       /* Ошибка: Переполнение типа FLOAT */
28     E_DOUBLE_OVERFLOW,      /* Ошибка: Переполнение типа DOUBLE */
29     E_LONG_DOUBLE_OVERFLOW, /* Ошибка: Переполнение типа DOUBLE */
30     E_INVALID_FLAG_ARG,     /* Ошибка: Неккоректный ввод аргументов */
31     E_SAME_FILE_NAMES,      /* Ошибка: Одинаковые имена файлов */
32     E_BUFFER_OVERFLOW,      /* Ошибка: Переполнение буфера */
33     E_SAME_FILES,           /* Ошибка: Ввод одинаковых файлов */
34     E_FALSE,                /* BOOL_FALSE */
35
36 } ERRORS_EXIT_CODES;
37
38 /* Перевод строки в int */
39 ERRORS_EXIT_CODES string_to_int(const char *str_number, int *int_result_number, int base)
40 {
41     if (str_number == NULL || int_result_number == NULL)
42         return E_INVALID_INPUT;
43     char *endptr;
44     *int_result_number = strtol(str_number, &endptr, base);
45     if (*int_result_number == INT_MAX || *int_result_number == INT_MIN)
46         return E_TYPE_OVERFLOW;
47     else if (*endptr != '\0')
48         return E_INVALID_INPUT;
49     return E_SUCCESS;
50 }
51
52 int is_float_overflow(float value)
53 {
54
55     // Проверка на превышение предельных значений для типа float
56     if (value > FLT_MAX || value < -FLT_MAX || value == HUGE_VALF || value == -HUGE_VALF)
57     {
58         return 1;
59     }
60     if (isinf(value))
61     {
62         return 1;
63     }
64
65     return 0;
66 }
67
68 /* Перевод строки в double */
69 ERRORS_EXIT_CODES string_to_float(const char *str, float *num)
70 {
71     if (str == NULL || num == NULL)
72         return E_DEREFENCE_NULL_POINTER;
73
74     char *endptr;
75     float value = strtod(str, &endptr);
76

```

```

77  if (*endptr != '\0')
78      return E_INVALID_INPUT;
79
80  if (is_float_overflow(value))
81      return E_DOUBLE_OVERFLOW;
82
83  *num = value;
84  return E_SUCCESS;
85 }
86
87 void write_error(const char *error_string)
88 {
89     if (error_string == NULL)
90     {
91         write(STDOUT_FILENO, "ERROR", 6);
92     }
93     write(STDOUT_FILENO, error_string, strlen(error_string));
94 }
95
96 void write_devision_result(float res)
97 {
98     char string[BUFFER_SIZE];
99     int len_str = snprintf(string, sizeof(string), "Devision result is: %f\n", res);
100    write(STDOUT_FILENO, string, len_str);
101 }
102
103 int main()
104 {
105     char file_buffer[BUFFER_SIZE];
106
107     while (fgets(file_buffer, sizeof(file_buffer), stdin) != NULL)
108     {
109
110         float devision_res = 0;
111         int flag = 1;
112
113         file_buffer[strcspn(file_buffer, "\n")] = '\0'; // Получаем строку
114         char *token = strtok(file_buffer, " "); // Сплитим по пробелам
115
116         while (token != NULL)
117         {
118             float res = 0;
119             ERRORS_EXIT_CODES error = string_to_float(token, &res);
120
121             if (error == E_DOUBLE_OVERFLOW)
122             {
123                 write_error("ERROR_DOUBLE_OVERFLOW\n");
124                 return error;
125             }
126             else if (error == E_INVALID_INPUT)
127             {
128                 write_error("INVALID INPUT IN FILE\n");
129                 return error;
130             }
131             else if (error == E_SUCCESS)
132             {
133                 if (res == 0.0 || fabsf(res) < FLT_EPSILON)
134                 {
135                     write_error("ERROR DEVISION BY ZERO\n");
136                     return E_INVALID_INPUT;
137                 }
138                 if (flag)
139                 {
140                     flag = 0;
141                     devision_res = res;

```

```
142     }
143     else
144     {
145         devision_res /= res;
146     }
147 }
148 else
149 {
150     write_error("ERROR\n");
151     return error;
152 }
153
154 token = strtok(NULL, " ");
155 }
156
157 write_devision_result(devision_res);
158 }
159 return E_SUCCESS; }
160
161
```

Протокол работы программы

```
qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ gcc parent.c -o parent
```

```
qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ gcc child.c -o child -lm
```

```
qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ ls
```

```
child child.c parent parent.c test1.txt test2.txt test3.txt
```

```
qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ ./parent
```

```
ERROR: User: where is a filename????
```

```
qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ ./parent no_exist_file.txt
```

```
ERROR: cannot open file
```

```
qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ ./parent test3.txt^C
```

```
qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ cat test1.txt
```

```
128 2 2 2 2
```

```
1024 2 2 2 2 2 2
```

```
8192 1024 2
```

```
27 3 3
```

```
100 3 3 3
```

```
111 5 5 5 5
```

```
1534453 3 3
```

```
1298312 2 3 4 5 6
```

```
0 90 9
```

```
qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ ./parent test1.txt
```

```
Devision result is: 8.000000
```

```
Devision result is: 16.000000
```

```
Devision result is: 4.000000
```

```
Devision result is: 3.000000
```

```
Devision result is: 3.703704
```

```
Devision result is: 0.177600
```

```
Devision result is: 170494.781250
```

```
Devision result is: 1803.211060
```

```
ERROR DEVISION BY ZERO
```

```
qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ cat test2.txt
```


1298312 2 3 4 5 6

asdasda 90 9

128 2 2 2 2

qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src\$./parent test3.txt

Devision result is: 8.000000

Devision result is: 16.000000

Devision result is: 4.000000

Devision result is: 3.000000

Devision result is: 3.703704

Devision result is: 0.177600

Devision result is: 170494.781250

Devision result is: 1803.211060

INVALID INPUT IN FILE

qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src\$ cat test4.txt

128 2 2 2 2

1024 2 2 2 2 2

8192 1024 2

27 3 3

100 3 3 3

990000 99 100

qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src\$./parent test4.txt

Devision result is: 8.000000

Devision result is: 16.000000

Devision result is: 4.000000

Devision result is: 3.000000

Devision result is: 3.703704

Devision result is: 100.000000

Strace:

qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src\$ strace ./parent test1.txt

```

execve("./parent", ["/parent", "test1.txt"], 0x7ffdb57b75b8 /* 30 vars */) = 0

brk(NULL) = 0x55672f8cc000

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffef3aebc40) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f205ffc3000

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=30263, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 30263, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f205ffb000

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"..., 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\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"..., 68,
896) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, 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"..., 784, 64) = 784

mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f205fd92000

mprotect(0x7f205fdb000, 2023424, PROT_NONE) = 0

mmap(0x7f205fdb000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f205fdb000

mmap(0x7f205ff4f000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1bd000) = 0x7f205ff4f000

mmap(0x7f205ffa8000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7f205ffa8000

mmap(0x7f205ffae000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f205ffae000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f205fd8f000

arch_prctl(ARCH_SET_FS, 0x7f205fd8f740) = 0

set_tid_address(0x7f205fd8fa10) = 7585

set_robust_list(0x7f205fd8fa20, 24) = 0

```

```

rseq(0x7f205fd900e0, 0x20, 0, 0x53053053) = 0

mprotect(0x7f205ffa8000, 16384, PROT_READ) = 0

mprotect(0x55672e7fe000, 4096, PROT_READ) = 0

mprotect(0x7f205fffd000, 8192, PROT_READ) = 0

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

munmap(0x7f205ffbb000, 30263) = 0

getrandom("\xd6\x7e\xa7\xca\x71\x6b\xb3\x3c", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x55672f8cc000

brk(0x55672f8ed000) = 0x55672f8ed000

openat(AT_FDCWD, "test1.txt", O_RDONLY) = 3

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

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f205fd8fa10) = 7586

close(4) = 0

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=108, ...}, AT_EMPTY_PATH) = 0

read(3, "128 2 2 2 2\n1024 2 2 2 2 2 2\n819"... , 4096) = 108

write(5, "128 2 2 2 2\n", 12Devision result is: 8.000000

) = 12

write(5, "1024 2 2 2 2 2 2\n", 17Devision result is: 16.000000

) = 17

write(5, "8192 1024 2 \n", 13Devision result is: 4.000000

) = 13

write(5, "27 3 3\n", 7Devision result is: 3.000000

) = 7

write(5, "100 3 3 3\n", 10Devision result is: 3.703704

) = 10

write(5, "111 5 5 5 5\n", 12Devision result is: 0.177600

) = 12

write(5, "1534453 3 3\n", 12Devision result is: 170494.781250

) = 12

write(5, "1298312 2 3 4 5 6\n", 18Devision result is: 1803.211060

```

) = 18

write(5, "0 90 9\n", 7ERROR DEVISION BY ZERO

) = 7

read(3, "", 4096) = 0

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=7586, si_uid=1000, si_status=1, si_utime=0, si_stime=0} ---

close(5) = 0

close(3) = 0

wait4(-1, NULL, 0, NULL) = 7586

exit_group(0) = ?

+++ exited with 0 +++

qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src\$ strace ./parent test2.txt

execve("./parent", ["/parent", "test2.txt"], 0x7ffcc3bb99e8 /* 30 vars */) = 0

brk(NULL) = 0x5566910f7000

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffdae9b77c0) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fb95c54a000

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=30263, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 30263, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fb95c542000

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\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\05\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\0I\17\357\204\3\$\f\221\2039x\324\224\323\236S"..., 68, 896) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, 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, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb95c319000

mprotect(0x7fb95c341000, 2023424, PROT_NONE) = 0

```

mmap(0x7fb95c341000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7fb95c341000

mmap(0x7fb95c4d6000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1bd000) = 0x7fb95c4d6000

mmap(0x7fb95c52f000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7fb95c52f000

mmap(0x7fb95c535000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fb95c535000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fb95c316000

arch_prctl(ARCH_SET_FS, 0x7fb95c316740) = 0

set_tid_address(0x7fb95c316a10) = 7595

set_robust_list(0x7fb95c316a20, 24) = 0

rseq(0x7fb95c3170e0, 0x20, 0, 0x53053053) = 0

mprotect(0x7fb95c52f000, 16384, PROT_READ) = 0

mprotect(0x55668f4a5000, 4096, PROT_READ) = 0

mprotect(0x7fb95c584000, 8192, PROT_READ) = 0

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

munmap(0x7fb95c542000, 30263) = 0

getrandom("\x26\x6d\xfa\x17\xe9\x36\x5d\x51", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x5566910f7000

brk(0x556691118000) = 0x556691118000

openat(AT_FDCWD, "test2.txt", O_RDONLY) = 3

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

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fb95c316a10) = 7596

close(4) = 0

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=219, ...}, AT_EMPTY_PATH) = 0

read(3, "128 2 2 2 2\n1024 2 2 2 2 2 2\n819"... , 4096) = 219

write(5, "128 2 2 2 2\n", 12) = 12

Devison result is: 8.000000

write(5, "1024 2 2 2 2 2 2\n", 17Devison result is: 16.000000

) = 17

```



```

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\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"..., 68,
896) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, 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"..., 784, 64) = 784

mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f2ff04c9000

mprotect(0x7f2ff04f1000, 2023424, PROT_NONE) = 0

mmap(0x7f2ff04f1000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f2ff04f1000

mmap(0x7f2ff0686000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1bd000) = 0x7f2ff0686000

mmap(0x7f2ff06df000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7f2ff06df000

mmap(0x7f2ff06e5000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f2ff06e5000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f2ff04c6000

arch_prctl(ARCH_SET_FS, 0x7f2ff04c6740) = 0

set_tid_address(0x7f2ff04c6a10) = 7605

set_robust_list(0x7f2ff04c6a20, 24) = 0

rseq(0x7f2ff04c70e0, 0x20, 0, 0x53053053) = 0

mprotect(0x7f2ff06df000, 16384, PROT_READ) = 0

mprotect(0x5626ae40a000, 4096, PROT_READ) = 0

mprotect(0x7f2ff0734000, 8192, PROT_READ) = 0

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

munmap(0x7f2ff06f2000, 30263) = 0

getrandom("\x28\x27\x75\x22\xfb\x62\xe9\x69", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x5626afbc1000

brk(0x5626afbe2000) = 0x5626afbe2000

openat(AT_FDCWD, "test3.txt", O_RDONLY) = 3

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

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f2ff04c6a10) = 7606

```



```

close(4) = 0

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=127, ...}, AT_EMPTY_PATH) = 0

read(3, "128 2 2 2 2\n1024 2 2 2 2 2\n819"..., 4096) = 127

write(5, "128 2 2 2 2\n", 12) = 12

Devison result is: 8.000000

write(5, "1024 2 2 2 2 2\n", 17Devison result is: 16.000000

) = 17

write(5, "8192 1024 2\n", 13Devison result is: 4.000000

) = 13

write(5, "27 3 3\n", 7Devison result is: 3.000000

) = 7

write(5, "100 3 3 3\n", 10Devison result is: 3.703704

) = 10

write(5, "111 5 5 5 5\n", 12Devison result is: 0.177600

) = 12

write(5, "1534453 3 3\n", 12Devison result is: 170494.781250

) = 12

write(5, "1298312 2 3 4 5 6\n", 18Devison result is: 1803.211060

) = 18

write(5, "asdasda 90 9\n", 13INVALID INPUT IN FILE

) = 13

write(5, "128 2 2 2 2\n", 12) = -1 EPIPE (Broken pipe)

--- SIGPIPE {si_signo=SIGPIPE, si_code=SI_USER, si_pid=7605, si_uid=1000} ---

+++ killed by SIGPIPE +++

qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$ strace ./parent test4.txt

execve("./parent", ["/parent", "test4.txt"], 0x7ffe539a68b8 /* 30 vars */) = 0

brk(NULL) = 0x5557bad65000

arch_prctl(0x3001 /* ARCH_??? */, 0x7fff85104e80) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fb56a238000

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=30263, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 30263, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fb56a230000
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\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"..., 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\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"..., 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, 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"..., 784, 64) = 784
mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb56a007000
mprotect(0x7fb56a02f000, 2023424, PROT_NONE) = 0
mmap(0x7fb56a02f000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7fb56a02f000
mmap(0x7fb56a1c4000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7fb56a1c4000
mmap(0x7fb56a21d000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7fb56a21d000
mmap(0x7fb56a223000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fb56a223000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fb56a004000
arch_prctl(ARCH_SET_FS, 0x7fb56a004740) = 0
set_tid_address(0x7fb56a004a10) = 7615
set_robust_list(0x7fb56a004a20, 24) = 0
rseq(0x7fb56a0050e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7fb56a21d000, 16384, PROT_READ) = 0
mprotect(0x5557b92c7000, 4096, PROT_READ) = 0
mprotect(0x7fb56a272000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

```

```

munmap(0x7fb56a230000, 30263)      = 0

getrandom("\x3f\xe5\xaf\x25\x4b\x7b\x28\xa0", 8, GRND_NONBLOCK) = 8

brk(NULL)                        = 0x5557bad65000

brk(0x5557bad86000)              = 0x5557bad86000

openat(AT_FDCWD, "test4.txt", O_RDONLY) = 3

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

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fb56a004a10) = 7616

close(4)                         = 0

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=73, ...}, AT_EMPTY_PATH) = 0

read(3, "128 2 2 2 2\n1024 2 2 2 2 2\n819"... , 4096) = 73

write(5, "128 2 2 2 2\n", 12)     = 12

Devison result is: 8.000000

write(5, "1024 2 2 2 2 2\n", 17Devison result is: 16.000000

)      = 17

write(5, "8192 1024 2 \n", 13Devison result is: 4.000000

)      = 13

write(5, "27 3 3\n", 7Devison result is: 3.000000

)      = 7

write(5, "100 3 3 3\n", 10Devison result is: 3.703704

)      = 10

write(5, "990000 99 100\n", 14Devison result is: 100.000000

)      = 14

read(3, "", 4096)                = 0

close(5)                         = 0

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=7616, si_uid=1000, si_status=0,
si_utime=0, si_stime=0} ---

close(3)                         = 0

wait4(-1, NULL, 0, NULL)         = 7616

exit_group(0)                    = ?

+++ exited with 0 +++

qwental@DESKTOP-NKF1EUK:~/workspace/OS_LABS/lab1/src$

```

Вывод

В ходе выполнения лабораторной работы я освоил навыки управления процессами в операционной системе и научился организовывать обмен данными между процессами с помощью каналов. В процессе выполнения работы я разработал и отладил программу на языке Си, обеспечивающую взаимодействие между процессами через передачу данных через каналы.