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: Коваленко Д.А

Преподаватель: Бахарев В.Д.

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

Дата: 17.11.24

Москва, 2024

Постановка задачи

Вариант 13.

Родительский процесс создает дочерний процесс. Первой строкой пользователь вводит имя файла через консоль родительского процесса. Файл открывается для чтения и перенаправляется в стандартный поток ввода дочернего процесса. Дочерний процесс считывает строки из этого файла, конвертирует их в верхний регистр и выводит в pipe1. Родительский процесс получает данные из pipe1, выполняет подсчет количества слов в каждой строке и выводит результат в консоль. Родительский и дочерний процессы должны быть представлены разными программами.

В файле находятся строки вида: «текст текст текст». Дочерний процесс должен преобразовывать символы каждой строки в верхний регистр. Родительский процесс должен считать количество слов в каждой полученной строке и выводить его вместе с преобразованной строкой. Числа, записанные в файле, игнорируются при подсчете слов.

Если входной файл пуст, дочерний и родительский процессы завершают свою работу без дополнительных действий.

Общий метод и алгоритм решения

Использованные системные вызовы:

* pid\_t fork(void) – создание нового процесса-потомка.
* execlp(const char \*file, const char \*arg, ...) – замена содержимого памяти текущего процесса другим исполняемым файлом.
* pid\_t waitpid(pid\_t pid, int \*stat\_loc, int options) – ожидание завершения процесса-потомка.
* int pipe(int pipefd[2]) – создание канала для межпроцессной передачи данных.
* int dup2(int oldfd, int newfd) – переопределение файлового дескриптора.
* int open(const char \*pathname, int flags) – открытие файла.
* int close(int fd) – закрытие файлового дескриптора.
* size\_t read(int fd, void \*buf, size\_t cnt) – чтение данных из файла.
* size\_t write(int fd, const void \*buf, size\_t cnt) – запись данных в файл.

В родительском процессе происходит чтение данных из файла, путь к которому вводит пользователь. Создаем каналы pipe1, pipe2 и pipe3 для связи между процессами. Создаем первый дочерний процесс с помощью вызова fork(). Стандартный поток ввода этого процесса перенаправляется в pipe1, чтобы принимать данные от родительского процесса. Стандартный поток вывода дочернего процесса перенаправляется в pipe2, чтобы передавать результаты промежуточной обработки второму дочернему процессу. Для выполнения новой программы в первом дочернем процессе используем execlp, что заменяет содержимое памяти текущего процесса на содержимое указанной программы. Если execlp не удается выполнить, выводится сообщение об ошибке, и процесс завершает свою работу с ошибкой.

Создаем второй дочерний процесс с помощью вызова fork(). Стандартный поток ввода этого процесса перенаправляется в pipe2, чтобы получать данные от первого дочернего процесса. Стандартный поток вывода перенаправляется в pipe3, чтобы передавать результаты обратно в родительский процесс. Используем execlp для запуска второй программы в этом процессе. При неудаче execlp выводится сообщение об ошибке, и процесс завершает выполнение с ошибкой.

В родительском процессе закрываем ненужные дескрипторы pipe1[0], pipe2[0], pipe2[1] и pipe3[1]. Родительский процесс записывает данные в pipe1, чтобы передать их первому дочернему процессу. Затем он считывает обработанные данные из pipe3 и выводит их в стандартный поток вывода, чтобы пользователь мог видеть результаты работы программы. После завершения передачи данных закрываем все дескрипторы, используемые родительским процессом, и вызываем waitpid, чтобы дождаться завершения работы обоих дочерних процессов.

Код программы

main/main.cpp

#include <cerrno>

#include <cstring>

#include <fcntl.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <unistd.h>

#include "error\_handling.h"

int main() {

char file\_path[1024];

write(STDOUT\_FILENO, "Введите путь к файлу: ", 25);

ssize\_t bytes\_read = read(STDIN\_FILENO, file\_path, sizeof(file\_path) - 1);

if (bytes\_read < 0) {

myPerror("Ошибка при чтении пути к файлу");

\_exit(errno);

}

if (bytes\_read > 0 && file\_path[bytes\_read - 1] == '\n') {

file\_path[bytes\_read - 1] = '\0';

}

int pipe1[2], pipe2[2], pipe3[2];

if (pipe(pipe1) == -1) {

myPerror("Ошибка при создании pipe1");

\_exit(errno);

}

if (pipe(pipe2) == -1) {

myPerror("Ошибка при создании pipe2");

\_exit(errno);

}

if (pipe(pipe3) == -1) {

myPerror("Ошибка при создании pipe3");

\_exit(errno);

}

pid\_t pid1 = fork();

if (pid1 < 0) {

myPerror("Ошибка при создании первого дочернего процесса");

\_exit(errno);

} else if (pid1 == 0) {

dup2(pipe1[0], STDIN\_FILENO);

dup2(pipe2[1], STDOUT\_FILENO);

close(pipe1[1]);

close(pipe2[0]);

close(pipe3[0]);

close(pipe3[1]);

execlp("./child1\_process", "./child1\_process", nullptr);

myPerror("Ошибка при выполнении exec для child1");

\_exit(errno);

}

pid\_t pid2 = fork();

if (pid2 < 0) {

myPerror("Ошибка при создании второго дочернего процесса");

\_exit(errno);

} else if (pid2 == 0) {

dup2(pipe2[0], STDIN\_FILENO);

dup2(pipe3[1], STDOUT\_FILENO);

close(pipe1[0]);

close(pipe1[1]);

close(pipe2[1]);

close(pipe3[0]);

execlp("./child2\_process", "./child2\_process", nullptr);

myPerror("Ошибка при выполнении exec для child2");

\_exit(errno);

}

close(pipe1[0]);

close(pipe2[0]);

close(pipe2[1]);

close(pipe3[1]);

int file\_fd = open(file\_path, O\_RDONLY);

if (file\_fd == -1) {

myPerror("Ошибка при открытии файла");

return 1;

}

char input\_buffer[1024];

ssize\_t file\_bytes\_read;

while ((file\_bytes\_read = read(file\_fd, input\_buffer, sizeof(input\_buffer))) >

0) {

if (write(pipe1[1], input\_buffer, file\_bytes\_read) == -1) {

myPerror("Ошибка при записи в pipe1 в родительском процессе");

break;

}

char result\_buffer[1024];

ssize\_t result\_read = read(pipe3[0], result\_buffer, sizeof(result\_buffer));

if (result\_read < 0) {

myPerror("Ошибка при чтении из pipe3 в родительском процессе");

break;

} else if (result\_read > 0) {

result\_buffer[result\_read] = '\0';

if (write(STDOUT\_FILENO, result\_buffer, result\_read) == -1) {

myPerror("Ошибка при выводе в STDOUT");

break;

}

}

}

if (file\_bytes\_read < 0) {

myPerror("Ошибка при чтении файла");

}

close(file\_fd);

close(pipe1[1]);

close(pipe3[0]);

int status;

if (waitpid(pid1, &status, 0) == -1) {

myPerror("Ошибка ожидания завершения child1");

}

if (waitpid(pid2, &status, 0) == -1) {

myPerror("Ошибка ожидания завершения child2");

}

return 0;

}

Child1\_process.cpp

#include "error\_handling.h"

#include <cctype>

#include <cerrno>

#include <cstring>

#include <unistd.h>

int main() {

char buffer[1024];

ssize\_t bytesRead = read(STDIN\_FILENO, buffer, sizeof(buffer));

if (bytesRead < 0) {

myPerror("Ошибка при чтении в child1");

\_exit(errno);

}

for (int i = 0; i < bytesRead; ++i) {

if (buffer[i] != '\n') {

buffer[i] = tolower(buffer[i]);

}

}

if (write(STDOUT\_FILENO, buffer, bytesRead) == -1) {

myPerror("Ошибка при записи в child1");

\_exit(errno);

}

return 0;

}

Child2\_process.cpp

#include "error\_handling.h"

#include <cctype>

#include <cerrno>

#include <cstring>

#include <unistd.h>

int main() {

char buffer[1024];

ssize\_t bytesRead = read(STDIN\_FILENO, buffer, sizeof(buffer));

if (bytesRead < 0) {

myPerror("Ошибка при чтении в child2");

\_exit(errno);

}

for (int i = 0; i < bytesRead; ++i) {

if (isspace(buffer[i]) && buffer[i] != '\n') {

buffer[i] = '\_';

}

}

if (write(STDOUT\_FILENO, buffer, bytesRead) == -1) {

myPerror("Ошибка при записи в child2");

\_exit(errno);

}

return 0;

}

Протокол работы программы

Тест 1:

$ input

123QWEWQe ewqwqWEQWE

rqwrqwWQRWQr rqwrqr

output

123QWEWQe\_ewqwqWEQWE

rqwrqwWQRWQr\_\_\_\_\_\_\_\_\_rqwrqr

Strace :  
  
  
strace -f ./main

execve("./main", ["./main"], 0x7ffd1a643648 /\* 55 vars \*/) = 0

brk(NULL) = 0x5f9192423000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffebf1ae990) = -1 EINVAL

(Недопустимый аргумент)

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x719010626000

access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (Нет такого файла или

каталога)

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

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=103491, ...},

AT\_EMPTY\_PATH) = 0

mmap(NULL, 103491, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x71901060c000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC)

= 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"..., 832)

= 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,

784, 64) = 784

pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 48,

848) = 48

pread64(3,

"\4\0\0\0\24\0\0\0\3\0\0\0GNU\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)

= 0x719010200000

mprotect(0x719010228000, 2023424, PROT\_NONE) = 0

mmap(0x719010228000, 1658880, PROT\_READ|PROT\_EXEC,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) =

0x719010228000

mmap(0x7190103bd000, 360448, PROT\_READ,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000) =

0x7190103bd000

mmap(0x719010416000, 24576, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x215000) =

0x719010416000

mmap(0x71901041c000, 52816, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x71901041c000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x719010609000

arch\_prctl(ARCH\_SET\_FS, 0x719010609740) = 0

set\_tid\_address(0x719010609a10) = 9499

set\_robust\_list(0x719010609a20, 24) = 0

rseq(0x71901060a0e0, 0x20, 0, 0x53053053) = 0

mprotect(0x719010416000, 16384, PROT\_READ) = 0

mprotect(0x5f91915d1000, 4096, PROT\_READ) = 0

mprotect(0x719010660000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024,

rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x71901060c000, 103491) = 0

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265

\320\277\321\203\321\202\321\214 \320", 25Введите путь � = 25

read(0, ./tes1.txt

"./tes1.txt\n", 1023) = 11

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

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

**pipe2([7, 8], 0) = 0**

clone(child\_stack=NULL,

flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLDstrace:

Process 9509 attached

, child\_tidptr=0x719010609a10) = 9509

[pid 9509] set\_robust\_list(0x719010609a20, 24) = 0

[pid 9499] clone(child\_stack=NULL,

flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD

<unfinished ...>

[pid 9509] dup2(3, 0) = 0

[pid 9509] dup2(6, 1strace: Process 9510 attached

<unfinished ...>

[pid 9499] <... clone resumed>, child\_tidptr=0x719010609a10) = 9510

[pid 9509] <... dup2 resumed>) = 1

[pid 9499] close(3 <unfinished ...>

[pid 9510] set\_robust\_list(0x719010609a20, 24 <unfinished ...>

[pid 9509] close(4 <unfinished ...>

[pid 9499] <... close resumed>) = 0

[pid 9510] <... set\_robust\_list resumed>) = 0

[pid 9509] <... close resumed>) = 0

[pid 9499] close(5 <unfinished ...>

[pid 9509] close(5 <unfinished ...>

[pid 9499] <... close resumed>) = 0

[pid 9509] <... close resumed>) = 0

[pid 9499] close(6 <unfinished ...>

[pid 9509] close(7 <unfinished ...>

[pid 9499] <... close resumed>) = 0

[pid 9509] <... close resumed>) = 0

[pid 9499] close(8 <unfinished ...>

[pid 9509] close(8 <unfinished ...>

[pid 9499] <... close resumed>) = 0

[pid 9509] <... close resumed>) = 0

[pid 9499] openat(AT\_FDCWD, "./tes1.txt", O\_RDONLY) = 3

**[pid 9509] execve("./child1\_process", ["./child1\_process"], 0x7ffebf1aeb68**

**/\* 55 vars \*/ <unfinished ...>**

[pid 9499] read(3, <unfinished ...>

[pid 9510] dup2(5, 0 <unfinished ...>

[pid 9499] <... read resumed>"123QWEWQe

ewqwqWEQWE\nrqwrqwWQRWQ"..., 1024) = 48

[pid 9510] <... dup2 resumed>) = 0

[pid 9499] write(4, "123QWEWQe ewqwqWEQWE\nrqwrqwWQRWQ"..., 48

<unfinished ...>

[pid 9509] <... execve resumed>) = 0

[pid 9499] <... write resumed>) = 48

[pid 9510] dup2(8, 1 <unfinished ...>

[pid 9499] read(7, <unfinished ...>

[pid 9510] <... dup2 resumed>) = 1

[pid 9509] brk(NULL <unfinished ...>

[pid 9510] close(3 <unfinished ...>

[pid 9509] <... brk resumed>) = 0x5f391a9ab000

[pid 9510] <... close resumed>) = 0

[pid 9509] arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fffdd3b4700 <unfinished ...>

[pid 9510] close(4 <unfinished ...>

[pid 9509] <... arch\_prctl resumed>) = -1 EINVAL (Недопустимый аргумент)

[pid 9510] <... close resumed>) = 0

[pid 9510] close(6) = 0

[pid 9509] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 9510] close(7 <unfinished ...>

[pid 9509] <... mmap resumed>) = 0x70947c793000

[pid 9510] <... close resumed>) = 0

[pid 9509] access("/etc/ld.so.preload", R\_OK <unfinished ...>

**[pid 9510] execve("./child2\_process", ["./child2\_process"], 0x7ffebf1aeb68**

**/\* 55 vars \*/ <unfinished ...>**

[pid 9509] <... access resumed>) = -1 ENOENT (Нет такого файла или

каталога)

[pid 9509] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) =

4

[pid 9509] newfstatat(4, "", {st\_mode=S\_IFREG|0644, st\_size=103491, ...},

AT\_EMPTY\_PATH) = 0

[pid 9509] mmap(NULL, 103491, PROT\_READ, MAP\_PRIVATE, 4, 0) =

0x70947c779000

[pid 9509] close(4 <unfinished ...>

[pid 9510] <... execve resumed>) = 0

[pid 9509] <... close resumed>) = 0

[pid 9509] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6",

O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 9510] brk(NULL) = 0x5c4db406e000

[pid 9509] <... openat resumed>) = 4

[pid 9510] arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffd206345e0 <unfinished ...>

[pid 9509] read(4, <unfinished ...>

[pid 9510] <... arch\_prctl resumed>) = -1 EINVAL (Недопустимый аргумент)

[pid 9509] <... read

resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"...,

832) = 832

[pid 9510] mmap(NULL, 8192, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 9509] pread64(4, <unfinished ...>

[pid 9510] <... mmap resumed>) = 0x7b2c80fc7000

[pid 9509] <... 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 9510] access("/etc/ld.so.preload", R\_OK <unfinished ...>

[pid 9509] pread64(4, <unfinished ...>

[pid 9510] <... access resumed>) = -1 ENOENT (Нет такого файла или

каталога)

[pid 9509] <... pread64 resumed>"\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 9510] openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC

<unfinished ...>

[pid 9509] pread64(4, <unfinished ...>

[pid 9510] <... openat resumed>) = 3

[pid 9509] <... pread64

resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0I\17\357\204\3$\f\221\2039x\324\224\3

23\236S"..., 68, 896) = 68

[pid 9510] newfstatat(3, "", <unfinished ...>

[pid 9509] newfstatat(4, "", <unfinished ...>

[pid 9510] <... newfstatat resumed>{st\_mode=S\_IFREG|0644, st\_size=103491,

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

[pid 9509] <... newfstatat resumed>{st\_mode=S\_IFREG|0755, st\_size=2220400,

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

[pid 9510] mmap(NULL, 103491, PROT\_READ, MAP\_PRIVATE, 3, 0

<unfinished ...>

[pid 9509] pread64(4, <unfinished ...>

[pid 9510] <... mmap resumed>) = 0x7b2c80fad000

[pid 9509] <... 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 9510] close(3 <unfinished ...>

[pid 9509] mmap(NULL, 2264656, PROT\_READ,

MAP\_PRIVATE|MAP\_DENYWRITE, 4, 0 <unfinished ...>

[pid 9510] <... close resumed>) = 0

[pid 9509] <... mmap resumed>) = 0x70947c400000

[pid 9510] openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6",

O\_RDONLY|O\_CLOEXEC <unfinished ...>

[pid 9509] mprotect(0x70947c428000, 2023424, PROT\_NONE <unfinished ...>

[pid 9510] <... openat resumed>) = 3

[pid 9510] read(3, <unfinished ...>

[pid 9509] <... mprotect resumed>) = 0

[pid 9510] <... read

resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"...,

832) = 832

[pid 9509] mmap(0x70947c428000, 1658880, PROT\_READ|PROT\_EXEC,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x28000 <unfinished ...>

[pid 9510] pread64(3,

"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

[pid 9509] <... mmap resumed>) = 0x70947c428000

[pid 9510] pread64(3, <unfinished ...>

[pid 9509] mmap(0x70947c5bd000, 360448, PROT\_READ,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x1bd000 <unfinished ...>

[pid 9510] <... pread64 resumed>"\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 9509] <... mmap resumed>) = 0x70947c5bd000

[pid 9510] pread64(3, <unfinished ...>

[pid 9509] mmap(0x70947c616000, 24576, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 4, 0x215000 <unfinished ...>

[pid 9510] <... pread64

resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0I\17\357\204\3$\f\221\2039x\324\224\3

23\236S"..., 68, 896) = 68

[pid 9509] <... mmap resumed>) = 0x70947c616000

[pid 9510] newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2220400, ...},

AT\_EMPTY\_PATH) = 0

[pid 9509] mmap(0x70947c61c000, 52816, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 9510] pread64(3,

"\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

[pid 9509] <... mmap resumed>) = 0x70947c61c000

[pid 9510] mmap(NULL, 2264656, PROT\_READ,

MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7b2c80c00000

[pid 9509] close(4 <unfinished ...>

[pid 9510] mprotect(0x7b2c80c28000, 2023424, PROT\_NONE <unfinished ...>

[pid 9509] <... close resumed>) = 0

[pid 9510] <... mprotect resumed>) = 0

[pid 9510] mmap(0x7b2c80c28000, 1658880, PROT\_READ|PROT\_EXEC,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000 <unfinished ...>

[pid 9509] mmap(NULL, 12288, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 9510] <... mmap resumed>) = 0x7b2c80c28000

[pid 9509] <... mmap resumed>) = 0x70947c776000

[pid 9510] mmap(0x7b2c80dbd000, 360448, PROT\_READ,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1bd000 <unfinished ...>

[pid 9509] arch\_prctl(ARCH\_SET\_FS, 0x70947c776740 <unfinished ...>

[pid 9510] <... mmap resumed>) = 0x7b2c80dbd000

[pid 9509] <... arch\_prctl resumed>) = 0

[pid 9510] mmap(0x7b2c80e16000, 24576, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x215000 <unfinished ...>

[pid 9509] set\_tid\_address(0x70947c776a10 <unfinished ...>

[pid 9510] <... mmap resumed>) = 0x7b2c80e16000

[pid 9509] <... set\_tid\_address resumed>) = 9509

[pid 9510] mmap(0x7b2c80e1c000, 52816, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 9509] set\_robust\_list(0x70947c776a20, 24 <unfinished ...>

[pid 9510] <... mmap resumed>) = 0x7b2c80e1c000

[pid 9509] <... set\_robust\_list resumed>) = 0

[pid 9509] rseq(0x70947c7770e0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 9510] close(3 <unfinished ...>

[pid 9509] <... rseq resumed>) = 0

[pid 9510] <... close resumed>) = 0

[pid 9510] mmap(NULL, 12288, PROT\_READ|PROT\_WRITE,

MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0 <unfinished ...>

[pid 9509] mprotect(0x70947c616000, 16384, PROT\_READ <unfinished ...>

[pid 9510] <... mmap resumed>) = 0x7b2c80faa000

[pid 9509] <... mprotect resumed>) = 0

[pid 9510] arch\_prctl(ARCH\_SET\_FS, 0x7b2c80faa740 <unfinished ...>

[pid 9509] mprotect(0x5f391a586000, 4096, PROT\_READ <unfinished ...>

[pid 9510] <... arch\_prctl resumed>) = 0

[pid 9509] <... mprotect resumed>) = 0

[pid 9510] set\_tid\_address(0x7b2c80faaa10 <unfinished ...>

[pid 9509] mprotect(0x70947c7cd000, 8192, PROT\_READ <unfinished ...>

[pid 9510] <... set\_tid\_address resumed>) = 9510

[pid 9509] <... mprotect resumed>) = 0

[pid 9510] set\_robust\_list(0x7b2c80faaa20, 24) = 0

[pid 9509] prlimit64(0, RLIMIT\_STACK, NULL, <unfinished ...>

[pid 9510] rseq(0x7b2c80fab0e0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 9509] <... prlimit64 resumed>{rlim\_cur=8192\*1024,

rlim\_max=RLIM64\_INFINITY}) = 0

[pid 9510] <... rseq resumed>) = 0

[pid 9509] munmap(0x70947c779000, 103491) = 0

[pid 9510] mprotect(0x7b2c80e16000, 16384, PROT\_READ) = 0

[pid 9509] read(0, <unfinished ...>

[pid 9510] mprotect(0x5c4db2a6b000, 4096, PROT\_READ <unfinished ...>

[pid 9509] <... read resumed>"123QWEWQe

ewqwqWEQWE\nrqwrqwWQRWQ"..., 1024) = 48

[pid 9510] <... mprotect resumed>) = 0

[pid 9510] mprotect(0x7b2c81001000, 8192, PROT\_READ <unfinished ...>

[pid 9509] write(1, "123qwewqe ewqwqweqwe\nrqwrqwwqrwq"..., 48 <unfinished

...>

[pid 9510] <... mprotect resumed>) = 0

[pid 9509] <... write resumed>) = 48

[pid 9509] exit\_group(0 <unfinished ...>

[pid 9510] prlimit64(0, RLIMIT\_STACK, NULL, <unfinished ...>

[pid 9509] <... exit\_group resumed>) = ?

[pid 9510] <... prlimit64 resumed>{rlim\_cur=8192\*1024,

rlim\_max=RLIM64\_INFINITY}) = 0

[pid 9510] munmap(0x7b2c80fad000, 103491) = 0

[pid 9509] +++ exited with 0 +++

[pid 9499] <... read resumed>0x7ffebf1ae630, 1024) = ? ERESTARTSYS (To be

restarted if SA\_RESTART is set)

[pid 9499] --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED,

si\_pid=9509, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

[pid 9510] read(0, <unfinished ...>

[pid 9499] read(7, <unfinished ...>

[pid 9510] <... read resumed>"123qwewqe ewqwqweqwe\nrqwrqwwqrwq"...,

1024) = 48

[pid 9510] write(1, "123qwewqe\_ewqwqweqwe\nrqwrqwwqrwq"..., 48

<unfinished ...>

[pid 9499] <... read resumed>"123qwewqe\_ewqwqweqwe\nrqwrqwwqrwq"...,

1024) = 48

[pid 9510] <... write resumed>) = 48

[pid 9499] write(1, "123qwewqe\_ewqwqweqwe\nrqwrqwwqrwq"...,

48123qwewqe\_ewqwqweqwe

rqwrqwwqrwqr\_\_\_\_\_\_\_\_rqwrqr

) = 48

[pid 9510] exit\_group(0 <unfinished ...>

[pid 9499] read(3, <unfinished ...>

[pid 9510] <... exit\_group resumed>) = ?

[pid 9499] <... read resumed>"", 1024) = 0

[pid 9499] close(3) = 0

[pid 9499] close(4) = 0

[pid 9499] close(7) = 0

[pid 9510] +++ exited with 0 +++

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=9510,

si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

wait4(9509, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 9509

wait4(9510, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 9510

exit\_group(0) = ?

+++ exited with 0 +++

При выполнении работы познакомился с системными функциями Linux. Возникли проблемы с чтением данных из дочернего процесса. Потребовалась внимательность не запутаться в дескрипторах.