

**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»**

Выполнил: А. В. Маркелов
Группа: М8О-207БВ-24
Преподаватель: Е. С. Миронов

Москва, 2025

Условие

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 программы (основной процесс) должен создать для решения 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.

Родительский процесс создает два дочерних процесса. Первой строкой пользователь в консоль родительского процесса вводит имя файла, которое будет использовано для открытия File с таким именем на запись для child1. Аналогично для второй строки и процесса child2. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс принимает от пользователя строки произвольной длины и пересыпает их в pipe1 или в pipe2 в зависимости от правила фильтрации. Процесс child1 и child2 производят работу над строками. Процессы пишут результаты своей работы в стандартный вывод. Правило фильтрации: нечетные строки отправляются в pipe1, четные в pipe2. Дочерние процессы инвертируют строки.

Вариант: 21

Метод решения

Родительский процесс принимает пользовательский ввод названий двух файлов — для нечетных и четных строк, создает или открывает их. Далее создаются два канала (p_odd, p_even) и два дочерних процесса (child_odd, child_even). Родитель читает строки из стандартного ввода и по очереди отправляет их в первый или второй канал. Дочерние процессы получают строки из своих каналов, инвертируют их и записывают в соответствующие файлы.

Описание программы

Общие интерфейсы системных вызовов оформлены в os.h. Функции Linux-реализации (pipe, fork и execve объединены в аналог CreateProcess (Windows), open, dup2, write, read, getline, waitpid, kill) описаны в os_linux.c.

В parent.c:

- Запрос имён файлов;
- Открытие файлов на запись;
- Создание каналов используя CreatePipe и дочерних процессов через CreateProc;
- Чтение строк и их пересылка в нужный канал согласно правилу фильтрации;
- Ожидание завершения дочерних процессов.

В child.c: - Чтение строк из STDIN канала через DoGetline;

- Инвертирование строки функцией Reverse;
- Запись результата в STDOUT (файл назначенный родителем).

Результаты

При запуске программы, открытии файлов odd.txt и even.txt и вводе тестовых строк нечётные строки попали в файл odd.txt в перевёрнутом виде, а четные — в файл even.txt также в перевернутом виде.

Пример работы:

Ввод:

```
odd  
even  
text  
asd  
123das
```

Test

^D

Содержимое odd.txt:

```
txet  
sad321
```

Содержимое even.txt:

```
dsa  
tseT
```

Выводы

Реализован обмен данными между процессами через безымянные каналы. Дочерние процессы корректно инвертируют строки и записывают их в разные файлы. Программа удовлетворяет требованиям задания. При выполнение приобретены и опробованы на практике навыки работы с процессами ОС и обменом данных между процессами посредством каналов

Исходная программа

```
1 #pragma once
2
3 #include <stddef.h>
4
5 #define _GNU_SOURCE
6
7 #ifdef _WIN32
8
9 #else
10 #include <errno.h>
11 #include <fcntl.h>
12 #include <stdio.h>
13 #include <stdlib.h>
14 #include <string.h>
15 #include <sys/wait.h>
16 #include <unistd.h>
17
18 typedef int pipe_t;
19 typedef int pid_t;
20
21 typedef struct {
22     pid_t pid;
23 } proc_info_t;
24 #endif
25
26 int CreatePipe(pipe_t new_pipe[2]);
27
28 proc_info_t CreateProc(const char* file, char* argv[], char* envp[],
29                         pipe_t stdin_pipe, pipe_t stdout_pipe);
30
31 int DoDup2(pipe_t old_fd, int new_fd);
32
33 int CloseObject(pipe_t fd);
34
35 int WaitObject(proc_info_t proc_info, int* status, int options);
36
37 pipe_t OpenObject(const char* path, int flags, int mode);
38
39 ssize_t PipeWrite(pipe_t fd, const void* line, size_t count);
40
41 ssize_t PipeRead(pipe_t fd, void* line, size_t count);
42
43 ssize_t DoGetline(char** line, size_t* n, FILE* stream);
44
45 void TerminateProc(int status);
```

Листинг 1: *Общий интерфейс*

```
1 #include "os.h"
2
3 int CreatePipe(pipe_t new_pipe[2]) {
4     int res = pipe(new_pipe);
5     if (res == -1) {
6         return -1;
7     }
```

```

8  if (fcntl(new_pipe[0], F_SETFD, FD_CLOEXEC) == -1 ||
9      fcntl(new_pipe[1], F_SETFD, FD_CLOEXEC) == -1) {
10     close(new_pipe[0]);
11     close(new_pipe[1]);
12     return -1;
13   }
14   return res;
15 }
16
17 proc_info_t CreateProc(const char* file, char* argv[], char* envp[],
18                        pipe_t stdin_pipe, pipe_t stdout_pipe) {
19   proc_info_t result;
20
21   pid_t pid = fork();
22
23   if (pid == -1) {
24     result.pid = -1;
25     return result;
26   }
27
28   if (pid == 0) {
29     if (dup2(stdin_pipe, STDIN_FILENO) == -1) {
30       perror("error while dup2 stdin");
31       TerminateProc(1);
32     }
33     if (dup2(stdout_pipe, STDOUT_FILENO) == -1) {
34       perror("error while dup2 stdout");
35       TerminateProc(1);
36     }
37
38     if (fcntl(stdin_pipe, F_SETFD, 0) == -1) {
39       perror("error while fcntl clear CLOEXEC flag from stdin_pipe");
40       TerminateProc(1);
41     }
42     if (fcntl(stdout_pipe, F_SETFD, 0) == -1) {
43       perror("error while fcntl clear CLOEXEC flag from stdout_pipe");
44       TerminateProc(1);
45     }
46
47     execve(file, argv, envp);
48     perror("error while execve");
49     TerminateProc(1);
50   }
51
52   result.pid = pid;
53   return result;
54 }
55
56 int DoDup2(pipe_t old_fd, int new_fd) { return dup2(old_fd, new_fd); }
57
58 int CloseObject(pipe_t fd) { return close(fd); }
59
60 int WaitObject(proc_info_t proc_info, int* status, int options) {
61   return waitpid(proc_info.pid, status, options);
62 }
63
64 pipe_t OpenObject(const char* path, int flags, int mode) {
65   return open(path, flags, mode);

```

```

66  }
67
68 ssize_t PipeWrite(pipe_t fd, const void* line, size_t count) {
69     return write(fd, line, count);
70 }
71
72 ssize_t PipeRead(pipe_t fd, void* line, size_t count) {
73     return read(fd, line, count);
74 }
75
76 ssize_t DoGetline(char** line, size_t* n, FILE* stream) {
77     return getline(line, n, stream);
78 }
79
80 void TerminateProc(int status) {
81     kill(getpid(), SIGTERM);
82     _exit(status);
83 }

```

Листинг 2: *Реализация системных вызовов только для POSIX-систем*

```

1 #include "os.h"
2
3 void WriteToPipe(pipe_t fd, const char *line, size_t n) {
4     size_t written = 0;
5
6     while (written < n) {
7         ssize_t s = PipeWrite(fd, line + written, n - written);
8
9         if (s == -1) {
10             if (errno == EINTR) {
11                 continue;
12             }
13             perror("error while write to pipe");
14             TerminateProc(1);
15         }
16
17         written += (size_t)s;
18     }
19 }
20
21 int main(int argc, char *argv[], char *envp[]) {
22     char file_odd[4096];
23     char file_even[4096];
24     printf("Enter a file name for odd lines: ");
25     fgets(file_odd, sizeof(file_odd), stdin);
26     printf("Enter a file name for even lines: ");
27     fgets(file_even, sizeof(file_even), stdin);
28
29     file_odd[strcspn(file_odd, "\n")] = 0;
30     file_even[strcspn(file_even, "\n")] = 0;
31
32     pipe_t fd_odd = OpenObject(file_odd, 0_CREAT | 0_TRUNC | 0_WRONLY, 00666);
33     if (fd_odd == -1) {
34         perror("error while create/open file for odd");
35         TerminateProc(1);
36     }

```

```

37 pipe_t fd_even = OpenObject(file_even, _CREATE | _TRUNC | _WRONLY, 00666);
38 if (fd_even == -1) {
39     perror("error while create/open file for even");
40     TerminateProc(1);
41 }
42
43 pipe_t p_odd[2];
44 pipe_t p_even[2];
45 if (CreatePipe(p_odd) == -1) {
46     perror("error while create pipe for odd");
47     TerminateProc(1);
48 }
49 if (CreatePipe(p_even) == -1) {
50     perror("error while create pipe for even");
51     TerminateProc(1);
52 }
53
54 char *args[] = {"child", NULL};
55
56 proc_info_t child_odd = CreateProc("./child", args, envp, p_odd[0], fd_odd);
57 if (child_odd.pid == -1) {
58     perror("error while create odd child process");
59     TerminateProc(1);
60 }
61
62 proc_info_t child_even =
63     CreateProc("./child", args, envp, p_even[0], fd_even);
64 if (child_even.pid == -1) {
65     perror("error while create even child process");
66     TerminateProc(1);
67 }
68
69 char *line = NULL;
70 int line_num = 1;
71 size_t capacity = 0;
72 ssize_t s;
73
74 while ((s = DoGetline(&line, &capacity, stdin)) != -1) {
75     if (line_num % 2 != 0) {
76         WriteToPipe(p_odd[1], line, (size_t)s);
77     } else {
78         WriteToPipe(p_even[1], line, (size_t)s);
79     }
80     line_num++;
81 }
82
83 CloseObject(p_odd[1]);
84 CloseObject(p_even[1]);
85
86 int status;
87 if (WaitObject(child_odd, &status, 0) == -1) {
88     perror("error in waitpid child_odd");
89     TerminateProc(1);
90 }
91 if (WaitObject(child_even, &status, 0) == -1) {
92     perror("error in waitpid child_even");
93     TerminateProc(1);
94 }
```

```
95 ||     free(line);
96 ||     return 0;
97 || }
```

Листинг 3: *Родительский процесс*

```
1 #include "os.h"
2
3 void Reverse(char* line, size_t n) {
4     int is_new_line = 0;
5     if (n > 0 && line[n - 1] == '\n') {
6         is_new_line = 1;
7         n--;
8     }
9
10    size_t i = 0;
11    size_t j = 0;
12    if (n > 0) {
13        j = n - 1;
14    }
15
16    while (i < j) {
17        char temp = line[i];
18        line[i] = line[j];
19        line[j] = temp;
20        j--;
21        i++;
22    }
23
24    if (is_new_line == 1) {
25        line[n] = '\n';
26        line[n + 1] = '\0';
27    } else {
28        line[n] = '\0';
29    }
30 }
31
32 int main() {
33     char* line = NULL;
34     size_t capacity = 0;
35     ssize_t s;
36
37     while ((s = DoGetline(&line, &capacity, stdin)) != -1) {
38         Reverse(line, (size_t)s);
39         if (fwrite(line, 1, strlen(line), stdout) != strlen(line)) {
40             perror("error while write in file");
41             TerminateProc(1);
42         }
43         fflush(stdout);
44     }
45
46     free(line);
47
48     return 0;
49 }
```

Листинг 4: *Дочерний процесс*


```
= 0x7f185656e000
21657 22:32:05.257541
mmap(0x7f18565c7000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0
= 0x7f18565c7000
21657 22:32:05.257778
mmap(0x7f18565cd000,52816,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1,0
= 0x7f18565cd000
21657 22:32:05.257974 close(3) = 0
21657 22:32:05.258169
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) =
0x7f18563ae000
21657 22:32:05.258416 arch_prctl(ARCH_SET_FS,0x7f18563ae740) = 0
21657 22:32:05.258652 set_tid_address(0x7f18563aea10) = 21657
21657 22:32:05.258859 set_robust_list(0x7f18563aea20,24) = 0
21657 22:32:05.259106 rseq(0x7f18563af0e0,0x20,0,0x53053053) = 0
21657 22:32:05.259429 mprotect(0x7f18565c7000,16384,PROT_READ) = 0
21657 22:32:05.259662 mprotect(0x5577a8383000,4096,PROT_READ) = 0
21657 22:32:05.259838 mprotect(0x7f185661c000,8192,PROT_READ) = 0
21657 22:32:05.260012
prlimit64(0,RLIMIT_STACK,NULL,{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY})
= 0
21657 22:32:05.260169 munmap(0x7f18565da000,29840) = 0
21657 22:32:05.260348
newfstatat(1,"",{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0x8),...},AT_EMPTY_PATH)
= 0
21657 22:32:05.260609
getrandom("\x62\x52\x12\xec\x54\xee\x63\xeb",8,GRND_NONBLOCK) = 8
21657 22:32:05.260871 brk(NULL) = 0x5577b2227000
21657 22:32:05.261210 brk(0x5577b2248000) = 0x5577b2248000
21657 22:32:05.261440
newfstatat(0,"",{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0x8),...},AT_EMPTY_PATH)
= 0
21657 22:32:05.261760 write(1,"Enter a file name for odd
lines:...",33) = 33
21657 22:32:05.262059 read(0,"odd\n",1024) = 4
21657 22:32:07.626601 write(1,"Enter a file name for even
lines...",34) = 34
21657 22:32:07.627084 read(0,"even\n",1024) = 5
21657 22:32:18.638922
openat(AT_FDCWD,"odd",O_WRONLY|O_CREAT|O_TRUNC,0666) = 3
21657 22:32:18.639383
openat(AT_FDCWD,"even",O_WRONLY|O_CREAT|O_TRUNC,0666) = 4
21657 22:32:18.639973 pipe2([5,6],0) = 0
21657 22:32:18.640291 fcntl(5,F_SETFD,FD_CLOEXEC) = 0
21657 22:32:18.640585 fcntl(6,F_SETFD,FD_CLOEXEC) = 0
21657 22:32:18.640937 pipe2([7,8],0) = 0
21657 22:32:18.641371 fcntl(7,F_SETFD,FD_CLOEXEC) = 0
21657 22:32:18.641663 fcntl(8,F_SETFD,FD_CLOEXEC) = 0
```

```
21657 22:32:18.641945
clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,child_tid
= 21767
    21767 22:32:18.642420 set_robust_list(0x7f18563aea20,24 <unfinished
...>
    21657 22:32:18.642471
clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD
<unfinished ...>
    21767 22:32:18.642517 <... set_robust_list resumed>) = 0
    21767 22:32:18.642693 dup2(5,0)          = 0
    21657 22:32:18.642862 <... clone
resumed>,child_tidptr=0x7f18563aea10) = 21768
    21768 22:32:18.642955 set_robust_list(0x7f18563aea20,24 <unfinished
...>
    21767 22:32:18.642998 dup2(3,1 <unfinished ...>
    21657 22:32:18.643039 read(0, <unfinished ...>
    21768 22:32:18.643101 <... set_robust_list resumed>) = 0
    21767 22:32:18.643143 <... dup2 resumed>) = 1
    21768 22:32:18.643185 dup2(7,0 <unfinished ...>
    21767 22:32:18.643226 fcntl(5,F_SETFD,0 <unfinished ...>
    21768 22:32:18.643270 <... dup2 resumed>) = 0
    21767 22:32:18.643328 <... fcntl resumed>) = 0
    21768 22:32:18.643370 dup2(4,1 <unfinished ...>
    21767 22:32:18.643411 fcntl(3,F_SETFD,0 <unfinished ...>
    21768 22:32:18.643453 <... dup2 resumed>) = 1
    21767 22:32:18.643516 <... fcntl resumed>) = 0
    21768 22:32:18.643558 fcntl(7,F_SETFD,0 <unfinished ...>
    21767 22:32:18.643599 execve("./child",["child"],0x7fff95509278 /*

37 vars */ <unfinished ...>
    21768 22:32:18.643685 <... fcntl resumed>) = 0
    21768 22:32:18.643803 fcntl(4,F_SETFD,0) = 0
    21767 22:32:18.644020 <... execve resumed>) = 0
    21768 22:32:18.644064 execve("./child",["child"],0x7fff95509278 /*

37 vars */ <unfinished ...>
    21767 22:32:18.644146 brk(NULL)          = 0x55d3e77d4000
    21768 22:32:18.644427 <... execve resumed>) = 0
    21767 22:32:18.644498 arch_prctl(0x3001 /* ARCH_???
*/ ,0x7ffdde7f63110 <unfinished ...>
    21768 22:32:18.644545 brk(NULL <unfinished ...>
    21767 22:32:18.644586 <... arch_prctl resumed>) = -1 EINVAL
(Invalid argument)
    21768 22:32:18.644635 <... brk resumed>) = 0x55fb5508b000
    21767 22:32:18.644723
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0
<unfinished ...>
    21768 22:32:18.644790 arch_prctl(0x3001 /* ARCH_???
*/ ,0x7ffd114f7df0 <unfinished ...>
    21767 22:32:18.644831 <... mmap resumed>) = 0x7f8eb9278000
```

```
21768 22:32:18.644873 <... arch_prctl resumed>) = -1 EINVAL
(Invalid argument)
21767 22:32:18.644939 access("/etc/ld.so.preload",R_OK <unfinished
...>
21768 22:32:18.644990
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0
<unfinished ...>
21767 22:32:18.645055 <... access resumed>) = -1 ENOENT (No such
file or directory)
21768 22:32:18.645098 <... mmap resumed>) = 0x7f1dea5f1000
21767 22:32:18.645177
openat(AT_FDCWD,"/etc/ld.so.cache",O_RDONLY|O_CLOEXEC <unfinished ...>
21768 22:32:18.645227 access("/etc/ld.so.preload",R_OK <unfinished
...>
21767 22:32:18.645273 <... openat resumed>) = 6
21768 22:32:18.645336 <... access resumed>) = -1 ENOENT (No such
file or directory)
21767 22:32:18.645378 newfstatat(6,"", <unfinished ...>
21768 22:32:18.645423
openat(AT_FDCWD,"/etc/ld.so.cache",O_RDONLY|O_CLOEXEC <unfinished ...>
21767 22:32:18.645491 <... newfstatat
resumed>{st_mode=S_IFREG|0644,st_size=29840,...},AT_EMPTY_PATH) = 0
21768 22:32:18.645539 <... openat resumed>) = 5
21767 22:32:18.645580 mmap(NULL,29840,PROT_READ,MAP_PRIVATE,6,0
<unfinished ...>
21768 22:32:18.645643 newfstatat(5,"", <unfinished ...>
21767 22:32:18.645688 <... mmap resumed>) = 0x7f8eb9270000
21768 22:32:18.645728 <... newfstatat
resumed>{st_mode=S_IFREG|0644,st_size=29840,...},AT_EMPTY_PATH) = 0
21767 22:32:18.645800 close(6 <unfinished ...>
21768 22:32:18.645843 mmap(NULL,29840,PROT_READ,MAP_PRIVATE,5,0
<unfinished ...>
21767 22:32:18.645912 <... close resumed>) = 0
21768 22:32:18.645952 <... mmap resumed>) = 0x7f1dea5e9000
21768 22:32:18.646105 close(5 <unfinished ...>
21767 22:32:18.646228
openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY|O_CLOEXEC
<unfinished ...>
21768 22:32:18.646324 <... close resumed>) = 0
21767 22:32:18.646422 <... openat resumed>) = 6
21768 22:32:18.646554
openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY|O_CLOEXEC
<unfinished ...>
21767 22:32:18.646605 read(6, <unfinished ...>
21768 22:32:18.646672 <... openat resumed>) = 5
21767 22:32:18.646714 <... 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
```

```
21768 22:32:18.646788 read(5, <unfinished ...>
21767 22:32:18.646831 pread64(6, <unfinished ...>
21768 22:32:18.646873 <... 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
    21767 22:32:18.646920 <... 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\0\0\0\0"...,784,64)
= 784
    21768 22:32:18.646987 pread64(5, <unfinished ...>
    21767 22:32:18.647033 pread64(6, <unfinished ...>
    21768 22:32:18.647075 <... 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"...,784,64)
= 784
    21767 22:32:18.647143 <... 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"...,48,848) = 48
    21768 22:32:18.647190 pread64(5, <unfinished ...>
    21767 22:32:18.647232 pread64(6, <unfinished ...>
    21768 22:32:18.647295 <... 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"...,48,848) = 48
    21767 22:32:18.647342 <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\00{`f\225`\=\201\327\312\301P\32$\230\266\235".
= 68
    21768 22:32:18.647410 pread64(5, <unfinished ...>
    21767 22:32:18.647453 newfstatat(6,"", <unfinished ...>
    21768 22:32:18.647521 <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\00{`f\225`\=\201\327\312\301P\32$\230\266\235".
= 68
    21767 22:32:18.647570 <... newfstatat
resumed>{st_mode=S_IFREG|0755,st_size=2220400,...},AT_EMPTY_PATH) = 0
    21768 22:32:18.647668 newfstatat(5,"", <unfinished ...>
    21767 22:32:18.647715 pread64(6, <unfinished ...>
    21768 22:32:18.647757 <... newfstatat
resumed>{st_mode=S_IFREG|0755,st_size=2220400,...},AT_EMPTY_PATH) = 0
    21767 22:32:18.647804 <... 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"...,784,64)
= 784
    21768 22:32:18.647849 pread64(5, <unfinished ...>
    21767 22:32:18.647892
mmap(NULL,2264656,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,6,0 <unfinished
...>
    21768 22:32:18.647936 <... 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"...,784,64)
= 784
    21768 22:32:18.648014
mmap(NULL,2264656,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,5,0 <unfinished
...>
    21767 22:32:18.648077 <... mmap resumed>) = 0x7f8eb9047000
    21768 22:32:18.648118 <... mmap resumed>) = 0x7f1dea3c0000
```

```
21767 22:32:18.648160 mprotect(0x7f8eb906f000,2023424,PROT_NONE
<unfinished ...>
21768 22:32:18.648221 mprotect(0x7f1dea3e8000,2023424,PROT_NONE
<unfinished ...>
21767 22:32:18.648263 <... mprotect resumed>) = 0
21768 22:32:18.648304 <... mprotect resumed>) = 0
21767 22:32:18.648345
mmap(0x7f8eb906f000,1658880,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,6
<unfinished ...>
21768 22:32:18.648389
mmap(0x7f1dea3e8000,1658880,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,5
<unfinished ...>
21767 22:32:18.648431 <... mmap resumed>) = 0x7f8eb906f000
21768 22:32:18.648472 <... mmap resumed>) = 0x7f1dea3e8000
21767 22:32:18.648513
mmap(0x7f8eb9204000,360448,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,6,0x1bd000
<unfinished ...>
21768 22:32:18.648556
mmap(0x7f1dea57d000,360448,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,5,0x1bd000
<unfinished ...>
21767 22:32:18.648599 <... mmap resumed>) = 0x7f8eb9204000
21768 22:32:18.648656 <... mmap resumed>) = 0x7f1dea57d000
21767 22:32:18.648699
mmap(0x7f8eb925d000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,6,
<unfinished ...>
21768 22:32:18.648738
mmap(0x7f1dea5d6000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,5,
<unfinished ...>
21767 22:32:18.648777 <... mmap resumed>) = 0x7f8eb925d000
21768 22:32:18.648815 <... mmap resumed>) = 0x7f1dea5d6000
21767 22:32:18.648865
mmap(0x7f8eb9263000,52816,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1
<unfinished ...>
21768 22:32:18.648904
mmap(0x7f1dea5dc000,52816,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1
<unfinished ...>
21767 22:32:18.648943 <... mmap resumed>) = 0x7f8eb9263000
21768 22:32:18.648981 <... mmap resumed>) = 0x7f1dea5dc000
21767 22:32:18.649020 close(6 <unfinished ...>
21768 22:32:18.649058 close(5 <unfinished ...>
21767 22:32:18.649100 <... close resumed>) = 0
21768 22:32:18.649139 <... close resumed>) = 0
21767 22:32:18.649177
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0
<unfinished ...>
21768 22:32:18.649260
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0
<unfinished ...>
```

```
21767 22:32:18.649317 <... mmap resumed>) = 0x7f8eb9044000
21768 22:32:18.649354 <... mmap resumed>) = 0x7f1dea3bd000
21767 22:32:18.649394 arch_prctl(ARCH_SET_FS,0x7f8eb9044740
<unfinished ...>
    21768 22:32:18.649455 arch_prctl(ARCH_SET_FS,0x7f1dea3bd740
<unfinished ...>
    21767 22:32:18.649494 <... arch_prctl resumed>) = 0
    21768 22:32:18.649532 <... arch_prctl resumed>) = 0
    21767 22:32:18.649572 set_tid_address(0x7f8eb9044a10 <unfinished
...>
    21768 22:32:18.649610 set_tid_address(0x7f1dea3bda10 <unfinished
...>
    21767 22:32:18.649648 <... set_tid_address resumed>) = 21767
    21768 22:32:18.649687 <... set_tid_address resumed>) = 21768
    21767 22:32:18.649728 set_robust_list(0x7f8eb9044a20,24 <unfinished
...>
    21768 22:32:18.649765 set_robust_list(0x7f1dea3bda20,24 <unfinished
...>
    21767 22:32:18.649800 <... set_robust_list resumed>) = 0
    21768 22:32:18.649837 <... set_robust_list resumed>) = 0
    21767 22:32:18.649873 rseq(0x7f8eb90450e0,0x20,0,0x53053053
<unfinished ...>
    21768 22:32:18.649909 rseq(0x7f1dea3be0e0,0x20,0,0x53053053
<unfinished ...>
    21767 22:32:18.649947 <... rseq resumed>) = 0
    21768 22:32:18.649983 <... rseq resumed>) = 0
    21768 22:32:18.650132 mprotect(0x7f1dea5d6000,16384,PROT_READ
<unfinished ...>
    21767 22:32:18.650208 mprotect(0x7f8eb925d000,16384,PROT_READ
<unfinished ...>
    21768 22:32:18.650256 <... mprotect resumed>) = 0
    21767 22:32:18.650292 <... mprotect resumed>) = 0
    21768 22:32:18.650329 mprotect(0x55fb2d3ef000,4096,PROT_READ
<unfinished ...>
    21767 22:32:18.650387 mprotect(0x55d3af477000,4096,PROT_READ
<unfinished ...>
    21768 22:32:18.650420 <... mprotect resumed>) = 0
    21767 22:32:18.650456 <... mprotect resumed>) = 0
    21768 22:32:18.650492 mprotect(0x7f1dea62b000,8192,PROT_READ
<unfinished ...>
    21767 22:32:18.650533 mprotect(0x7f8eb92b2000,8192,PROT_READ
<unfinished ...>
    21768 22:32:18.650571 <... mprotect resumed>) = 0
    21767 22:32:18.650619 <... mprotect resumed>) = 0
    21768 22:32:18.650685 prlimit64(0,RLIMIT_STACK,NULL, <unfinished
...>
    21767 22:32:18.650724 prlimit64(0,RLIMIT_STACK,NULL, <unfinished
...>
```

```
21768 22:32:18.650763 <... prlimit64
resumed>{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY}) = 0
21767 22:32:18.650835 <... prlimit64
resumed>{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY}) = 0
21768 22:32:18.650880 munmap(0x7f1dea5e9000,29840 <unfinished ...>
21767 22:32:18.650918 munmap(0x7f8eb9270000,29840 <unfinished ...>
21768 22:32:18.651035 <... munmap resumed>) = 0
21767 22:32:18.651121 <... munmap resumed>) = 0
21768 22:32:18.651223 getrandom( <unfinished ...>
21767 22:32:18.651295 getrandom( <unfinished ...>
21768 22:32:18.651330 <... getrandom
resumed>"\x12\x36\x24\xde\x87\xb8\x50\x76",8,GRND_NONBLOCK) = 8
21767 22:32:18.651397 <... getrandom
resumed>"\xba\x8d\x44\x92\x84\x39\xce\x7f",8,GRND_NONBLOCK) = 8
21768 22:32:18.651438 brk(NULL <unfinished ...>
21767 22:32:18.651482 brk(NULL <unfinished ...>
21768 22:32:18.651537 <... brk resumed>) = 0x55fb5508b000
21767 22:32:18.651573 <... brk resumed>) = 0x55d3e77d4000
21768 22:32:18.651629 brk(0x55fb550ac000 <unfinished ...>
21767 22:32:18.651666 brk(0x55d3e77f5000 <unfinished ...>
21768 22:32:18.651722 <... brk resumed>) = 0x55fb550ac000
21767 22:32:18.651758 <... brk resumed>) = 0x55d3e77f5000
21768 22:32:18.651794 newfstatat(0,"", <unfinished ...>
21767 22:32:18.651869 newfstatat(0,"", <unfinished ...>
21768 22:32:18.651915 <... newfstatat
resumed>{st_mode=S_IFIFO|0600,st_size=0,...},AT_EMPTY_PATH) = 0
21767 22:32:18.651959 <... newfstatat
resumed>{st_mode=S_IFIFO|0600,st_size=0,...},AT_EMPTY_PATH) = 0
21768 22:32:18.652001 read(0, <unfinished ...>
21767 22:32:18.652035 read(0, <unfinished ...>
21657 22:32:20.254720 <... read resumed>"text\n",1024) = 5
21657 22:32:20.254943 write(6,"text\n",5) = 5
21767 22:32:20.255286 <... read resumed>"text\n",4096) = 5
21657 22:32:20.255432 read(0, <unfinished ...>
21767 22:32:20.255552
newfstatat(1,"",{st_mode=S_IFREG|0644,st_size=0,...},AT_EMPTY_PATH) = 0
21767 22:32:20.255993 write(1,"txet\n",5) = 5
21767 22:32:20.256521 read(0, <unfinished ...>
21657 22:32:21.039197 <... read resumed>"asd\n",1024) = 4
21657 22:32:21.039483 write(8,"asd\n",4) = 4
21768 22:32:21.039846 <... read resumed>"asd\n",4096) = 4
21657 22:32:21.040039 read(0, <unfinished ...>
21768 22:32:21.040161
newfstatat(1,"",{st_mode=S_IFREG|0644,st_size=0,...},AT_EMPTY_PATH) = 0
21768 22:32:21.040572 write(1,"dsa\n",4) = 4
21768 22:32:21.040900 read(0, <unfinished ...>
21657 22:32:22.982080 <... read resumed>"123das\n",1024) = 7
21657 22:32:22.982344 write(6,"123das\n",7) = 7
```

```
21767 22:32:22.982792 <... read resumed>"123das\n",4096) = 7
21657 22:32:22.982944 read(0, <unfinished ...>
21767 22:32:22.983079 write(1,"sad321\n",7) = 7
21767 22:32:22.983472 read(0, <unfinished ...>
21657 22:32:24.349829 <... read resumed>"Test\n",1024) = 5
21657 22:32:24.350050 write(8,"Test\n",5) = 5
21768 22:32:24.350380 <... read resumed>"Test\n",4096) = 5
21657 22:32:24.350583 read(0, <unfinished ...>
21768 22:32:24.350694 write(1,"tseT\n",5) = 5
21768 22:32:24.350945 read(0, <unfinished ...>
21657 22:32:25.017900 <... read resumed>"",1024) = 0
21657 22:32:25.018254 close(6) = 0
21767 22:32:25.018718 <... read resumed>"",4096) = 0
21657 22:32:25.018801 close(8) = 0
21768 22:32:25.018995 <... read resumed>"",4096) = 0
21767 22:32:25.019044 exit_group(0 <unfinished ...>
21657 22:32:25.019092 wait4(21767, <unfinished ...>
21768 22:32:25.019166 exit_group(0 <unfinished ...>
21767 22:32:25.019220 <... exit_group resumed>) = ?
21768 22:32:25.019266 <... exit_group resumed>) = ?
21767 22:32:25.019494 +++ exited with 0 ===+
21768 22:32:25.019590 +++ exited with 0 ===+
21657 22:32:25.019612 <... wait4 resumed>[{WIFEXITED(s) &&
WEXITSTATUS(s) == 0}],0,NULL) = 21767
21657 22:32:25.019753 ---SIGCHLD
{si_signo=SIGCHLD,si_code=CLD_EXITED,si_pid=21767,si_uid=1000,si_status=0,si_utime=0,
---+
21657 22:32:25.019975 wait4(21768,[{WIFEXITED(s) && WEXITSTATUS(s)
== 0}],0,NULL) = 21768
21657 22:32:25.020441 exit_group(0) = ?
21657 22:32:25.020889 +++ exited with 0 ===+
```

Листинг 5: *Strace логи*