

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-215Б-23

Студент: Авраменко Д.А

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_

Дата: 02.10.24

Москва, 2024

# Постановка задачи

## Вариант 19.

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 пересылает 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.

Вариант 19) Правило фильтрации: с вероятностью 80% строки отправляются в pipe1, иначе в pipe2. Дочерние процессы удаляют все гласные из строк.

## Общий метод и алгоритм решения

Использованные системные вызовы:

- `pid_t fork(void);` – создает дочерний процесс.
- `int pipe(int *fd);` – создает пайпы для взаимодействия разных процессов
- `int dup2(int __fd, int __fd2);` - дублирует fd на fd2, закрыв fd2 и открыв его в том же файле.
- `int execl(const char* __path, const char* __arg, ...);` - заменяет текущий процесс новым. (l - list, аргументы передаются через запятую)
- `int close(int __fd);` - закрывает переданный fd
- `__pid_t waitpid(__pid_t __pid, int* __stat_loc, int __options);` - ждет завершения процесса

Алгоритм решения

### Родитель

1. Инициализируем две пайпы для двух будущих процессов
2. Инициализируем генератор случайных чисел
3. Вводим имена для будущих файлов
4. Создаем первый дочерний процесс
  - 4.1 Закрываем вторую пайпу с концами
  - 4.2 Закрываем первую пайпу для записи
  - 4.3 Дублируем чтение первой пайпы на стандартный ввод (чтобы ребенок мог читать из стандартного ввода)
  - 4.4 Закрываем чтение с первой пайпы (она нам больше не нужна, так как ее копия есть на стандартном вводе)
  - 4.5 Запускаем ребенка, передавая ему имя первого файла
  - 4.6 Хэндлим ошибку запуска
5. Создаем второй дочерний процесс
  - 5.1 Закрываем первую пайпу с концами
  - 5.2 Закрываем вторую пайпу для записи
  - 5.3 Дублируем чтение второй пайпы на стандартный ввод
  - 5.4 Закрываем чтение с второй пайпы

- 5.5 Запускаем ребенка, передавая ему имя второго файла
- 5.6 Хэндлим ошибку запуска
6. В родительском процессе закрываем обе пайпы на чтение
7. В бесконечном цикле обрабатываем строки (прервать можно Ctrl+D)
  - 7.1 Роллим рандом
  - 7.2 В зависимости от получившегося значения пишем в нужную пайпу полученную строку
8. Закрываем пайпы для записи
9. Ждем завершения дочерних процессов
10. Готово!

## Дети 1 и 2

1. Пытаемся создать (открыть для записи) файл с переданным именем
2. В бесконечном цикле считываем из стандартного ввода строки (из стандартного т.к мы дублировали в него запись пайпы в родителе)
3. Для каждой строки удаляем все гласные (на самом деле просто их пропускаем, что позволяет это сделать в той же строке, экономя память)

## Код программы

### parent.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <unistd.h>
4  #include <string.h>
5  #include <time.h>
6  #include <sys/wait.h>
7  #include <fcntl.h>
8
9  #define MAX_LINE 1000
10 #define PROB_PIPE1 0.8
11
12 int main() {
13     int pipe1[2], pipe2[2];
14     pid_t child1, child2;
15     char filename1[256], filename2[256];
16
17     // Создаем каналы
18     if (pipe(pipe1) == -1 || pipe(pipe2) == -1) {
19         perror("Ошибка создания канала");
20         exit(1);
21     }
22
23     // Инициализируем генератор случайных чисел
24     srand(time(NULL));
25
26     // Получаем имена файлов от пользователя
27     printf("Введите имя файла для child1: ");
28     scanf("%255s", filename1);
29     printf("Введите имя файла для child2: ");
30     scanf("%255s", filename2);
31
32     // Создаем первый дочерний процесс
33     child1 = fork();
34     if (child1 == -1) {
35         perror("Ошибка создания первого дочернего процесса");
```

```

36         exit(1);
37     } else if (child1 == 0) {
38         // Код для child1
39         close(pipe1[1]);
40         dup2(pipe1[0], STDIN_FILENO);
41         close(pipe1[0]);
42         close(pipe2[0]);
43         close(pipe2[1]);
44         execl("./child1", "child1", filename1, NULL);
45         perror("Ошибка execl для child1");
46         exit(1);
47     }
48
49     // Создаем второй дочерний процесс
50     child2 = fork();
51     if (child2 == -1) {
52         perror("Ошибка создания второго дочернего процесса");
53         exit(1);
54     } else if (child2 == 0) {
55         // Код для child2
56         close(pipe2[1]);
57         dup2(pipe2[0], STDIN_FILENO);
58         close(pipe2[0]);
59         close(pipe1[0]);
60         close(pipe1[1]);
61         execl("./child2", "child2", filename2, NULL);
62         perror("Ошибка execl для child2");
63         exit(1);
64     }
65
66     // Код родительского процесса
67     close(pipe1[0]);
68     close(pipe2[0]);
69
70     char line[MAX_LINE];
71     printf("Введите строки (Ctrl+D для завершения):\n");
72     while (fgets(line, MAX_LINE, stdin) != NULL) {
73         if (((double)rand() / RAND_MAX) < PROB_PIPE1) {
74             write(pipe1[1], line, strlen(line));
75         } else {
76             write(pipe2[1], line, strlen(line));
77         }
78     }
79
80     close(pipe1[1]);
81     close(pipe2[1]);
82
83     // Ожидаем завершения дочерних процессов
84     waitpid(child1, NULL, 0);
85     waitpid(child2, NULL, 0);
86
87     printf("Все процессы завершены.\n");
88     return 0;
89 }

```

### child1.c и child2.c

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <ctype.h>
5
6  #define MAX_LINE 1000
7
8  int is_vowel(char c) {
9      c = tolower(c);

```

```

10     return (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
11 }
12
13 void remove_vowels(char *str) {
14     int i, j;
15     for (i = 0, j = 0; str[i]; i++) {
16         if (!is_vowel(str[i])) {
17             str[j++] = str[i];
18         }
19     }
20     str[j] = '\0';
21 }
22
23 int main(int argc, char *argv[]) {
24     if (argc != 2) {
25         fprintf(stderr, "Использование: %s <имя_файла>\n", argv[0]);
26         exit(1);
27     }
28
29     FILE *file = fopen(argv[1], "w");
30     if (file == NULL) {
31         perror("Ошибка открытия файла");
32         exit(1);
33     }
34
35     char line[MAX_LINE];
36     while (fgets(line, MAX_LINE, stdin) != NULL) {
37         remove_vowels(line);
38         fprintf(file, "%s", line);
39     }
40
41     fclose(file);
42     return 0;
43 }

```

## Протокол работы программы

### Тестим:

```

1  root@d1282f6d624c:/IdeaProjects/MAI_OS_Labs/build# ./Lab1_exe
2  Введите имя файла для child1: biba
3  Введите имя файла для child2: boba
4  Введите строки (Ctrl+D для завершения):
5  asasasasasaasasasa
6  bobobobobobobobob
7  iiiiiiiiii
8  llllllllllll
9  ufo
10 ihatec
11 Все процессы завершены.
12 root@d1282f6d624c:/IdeaProjects/MAI_OS_Labs/build# cat < biba
13
14 bbbbbbbb
15 llllllllllll
16 f
17 htc
18 root@d1282f6d624c:/IdeaProjects/MAI_OS_Labs/build# cat < boba
19 ssssssss
20

```

## Трейсинг:

```
1  root@d1282f6d624c:/IdeaProjects/MAI_OS_Labs/build# strace -f ./Lab1_exe
2  execve("./Lab1_exe", ["/Lab1_exe"], 0x7ffebcb7f928 /* 21 vars */) = 0
3  brk(NULL) = 0x1719000
4  mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f741ca89000
5  access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or
directory)
6  openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
7  newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=25258, ...}, AT_EMPTY_PATH)
= 0
8  mmap(NULL, 25258, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f741ca82000
9  close(3) = 0
10 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
11 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20t\2\0\0\0\0"...,
832) = 832
12 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
13 newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1922136, ...},
AT_EMPTY_PATH) = 0
14 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
15 mmap(NULL, 1970000, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f741c8a1000
16 mmap(0x7f741c8c7000, 1396736, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000) = 0x7f741c8c7000
17 mmap(0x7f741ca1c000, 339968, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x17b000) = 0x7f741ca1c000
18 mmap(0x7f741ca6f000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1ce000) = 0x7f741ca6f000
19 mmap(0x7f741ca75000, 53072, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f741ca75000
20 close(3) = 0
21 mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f741c89e000
22 arch_prctl(ARCH_SET_FS, 0x7f741c89e740) = 0
23 set_tid_address(0x7f741c89ea10) = 39369
24 set_robust_list(0x7f741c89ea20, 24) = 0
25 rseq(0x7f741c89f060, 0x20, 0, 0x53053053) = 0
26 mprotect(0x7f741ca6f000, 16384, PROT_READ) = 0
27 mprotect(0x403000, 4096, PROT_READ) = 0
28 mprotect(0x7f741cabb000, 8192, PROT_READ) = 0
29 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
30 munmap(0x7f741ca82000, 25258) = 0
31 pipe2([3, 4], 0) = 0
32 pipe2([5, 6], 0) = 0
33 newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...},
AT_EMPTY_PATH) = 0
34 getrandom("\x18\x8e\x0f\x4a\xed\xc0\x07\x23", 8, GRND_NONBLOCK) = 8
35 brk(NULL) = 0x1719000
36 brk(0x173a000) = 0x173a000
37 newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...},
AT_EMPTY_PATH) = 0
38 write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\270\320\274\321\217 \321\204\320\260\320\271\320\273\320\260"..., 48Введите
имя файла для child1: ) = 48
39 read(0, biba
"biba\n", 1024) = 5
41 write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\270\320\274\321\217 \321\204\320\260\320\271\320\273\320\260"..., 48Введите
имя файла для child2: ) = 48
42 read(0, boba
"boba\n", 1024) = 5
```

```

44 clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace: Process 39420
attached
45 , child_tidptr=0x7f741c89ea10) = 39420
46 [pid 39420] set_robust_list(0x7f741c89ea20, 24) = 0
47 [pid 39369] clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD <unfinished ...>
48 [pid 39420] close(4strace: Process 39421 attached
49 <unfinished ...>
50 [pid 39369] <... clone resumed>, child_tidptr=0x7f741c89ea10) = 39421
51 [pid 39420] <... close resumed>) = 0
52 [pid 39421] set_robust_list(0x7f741c89ea20, 24 <unfinished ...>
53 [pid 39369] close(3 <unfinished ...>
54 [pid 39420] dup2(3, 0 <unfinished ...>
55 [pid 39369] <... close resumed>) = 0
56 [pid 39421] <... set_robust_list resumed>) = 0
57 [pid 39369] close(5 <unfinished ...>
58 [pid 39420] <... dup2 resumed>) = 0
59 [pid 39369] <... close resumed>) = 0
60 [pid 39421] close(6 <unfinished ...>
61 [pid 39420] close(3 <unfinished ...>
62 [pid 39369] write(1,
"\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\320\272\320\270 (Ctr"...
, 66 <unfinished ...>
63 Введите строки (Ctrl+D для завершения):
64 [pid 39421] <... close resumed>) = 0
65 [pid 39420] <... close resumed>) = 0
66 [pid 39369] <... write resumed>) = 66
67 [pid 39421] dup2(5, 0 <unfinished ...>
68 [pid 39369] write(4, "\n", 1 <unfinished ...>
69 [pid 39420] close(5 <unfinished ...>
70 [pid 39369] <... write resumed>) = 1
71 [pid 39421] <... dup2 resumed>) = 0
72 [pid 39420] <... close resumed>) = 0
73 [pid 39369] read(0, <unfinished ...>
74 [pid 39421] close(5 <unfinished ...>
75 [pid 39420] close(6 <unfinished ...>
76 [pid 39421] <... close resumed>) = 0
77 [pid 39420] <... close resumed>) = 0
78 [pid 39421] close(3) = 0
79 [pid 39420] execve("./child1", ["child1", "biba"], 0x7fff3e46ed78 /* 21 vars
*/ <unfinished ...>
80 [pid 39421] close(4) = 0
81 [pid 39421] execve("./child2", ["child2", "boba"], 0x7fff3e46ed78 /* 21 vars
*/) = 0
82 [pid 39420] <... execve resumed>) = 0
83 [pid 39421] brk(NULL <unfinished ...>
84 [pid 39420] brk(NULL <unfinished ...>
85 [pid 39421] <... brk resumed>) = 0x1957000
86 [pid 39420] <... brk resumed>) = 0x16ea000
87 [pid 39421] mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
88 [pid 39420] mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
89 [pid 39421] <... mmap resumed>) = 0x7f5e83cb8000
90 [pid 39420] <... mmap resumed>) = 0x7f63f4ef6000
91 [pid 39421] access("/etc/ld.so.preload", R_OK <unfinished ...>
92 [pid 39420] access("/etc/ld.so.preload", R_OK <unfinished ...>
93 [pid 39421] <... access resumed>) = -1 ENOENT (No such file or
directory)
94 [pid 39420] <... access resumed>) = -1 ENOENT (No such file or
directory)
95 [pid 39421] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC
<unfinished ...>
96 [pid 39420] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC
<unfinished ...>
97 [pid 39421] <... openat resumed>) = 3

```

```

98 [pid 39420] <... openat resumed>          = 3
99 [pid 39421] newfstatat(3, "", <unfinished ...>
100 [pid 39420] newfstatat(3, "", <unfinished ...>
101 [pid 39421] <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=25258,
..., AT_EMPTY_PATH) = 0
102 [pid 39420] <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=25258,
..., AT_EMPTY_PATH) = 0
103 [pid 39421] mmap(NULL, 25258, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>
104 [pid 39420] mmap(NULL, 25258, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>
105 [pid 39421] <... mmap resumed>          = 0x7f5e83cb1000
106 [pid 39420] <... mmap resumed>          = 0x7f63f4eef000
107 [pid 39421] close(3 <unfinished ...>
108 [pid 39420] close(3 <unfinished ...>
109 [pid 39421] <... close resumed>          = 0
110 [pid 39420] <... close resumed>          = 0
111 [pid 39421] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6",
O_RDONLY|O_CLOEXEC <unfinished ...>
112 [pid 39420] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6",
O_RDONLY|O_CLOEXEC <unfinished ...>
113 [pid 39421] <... openat resumed>          = 3
114 [pid 39420] <... openat resumed>          = 3
115 [pid 39421] read(3, <unfinished ...>
116 [pid 39420] read(3, <unfinished ...>
117 [pid 39421] <... read
resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20t\2\0\0\0\0\0"..., 832)
= 832
118 [pid 39420] <... read
resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20t\2\0\0\0\0\0"..., 832)
= 832
119 [pid 39421] pread64(3, <unfinished ...>
120 [pid 39420] pread64(3, <unfinished ...>
121 [pid 39421] <... 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"..., 784,
64) = 784
122 [pid 39420] <... 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"..., 784,
64) = 784
123 [pid 39421] newfstatat(3, "", <unfinished ...>
124 [pid 39420] newfstatat(3, "", <unfinished ...>
125 [pid 39421] <... newfstatat resumed>{st_mode=S_IFREG|0755, st_size=1922136,
..., AT_EMPTY_PATH) = 0
126 [pid 39420] <... newfstatat resumed>{st_mode=S_IFREG|0755, st_size=1922136,
..., AT_EMPTY_PATH) = 0
127 [pid 39421] pread64(3, <unfinished ...>
128 [pid 39420] pread64(3, <unfinished ...>
129 [pid 39421] <... 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"..., 784,
64) = 784
130 [pid 39420] <... 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"..., 784,
64) = 784
131 [pid 39421] mmap(NULL, 1970000, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0
<unfinished ...>
132 [pid 39420] mmap(NULL, 1970000, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0
<unfinished ...>
133 [pid 39421] <... mmap resumed>          = 0x7f5e83ad0000
134 [pid 39420] <... mmap resumed>          = 0x7f63f4d0e000
135 [pid 39421] mmap(0x7f5e83af6000, 1396736, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000 <unfinished ...>
136 [pid 39420] mmap(0x7f63f4d34000, 1396736, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000 <unfinished ...>
137 [pid 39421] <... mmap resumed>          = 0x7f5e83af6000
138 [pid 39421] mmap(0x7f5e83c4b000, 339968, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17b000 <unfinished ...>
139 [pid 39420] <... mmap resumed>          = 0x7f63f4d34000
140 [pid 39421] <... mmap resumed>          = 0x7f5e83c4b000

```



```

141 [pid 39420] mmap(0x7f63f4e89000, 339968, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17b000 <unfinished ...>
142 [pid 39421] mmap(0x7f5e83c9e000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1ce000 <unfinished ...>
143 [pid 39420] <... mmap resumed>) = 0x7f63f4e89000
144 [pid 39421] <... mmap resumed>) = 0x7f5e83c9e000
145 [pid 39420] mmap(0x7f63f4edc000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1ce000 <unfinished ...>
146 [pid 39421] mmap(0x7f5e83ca4000, 53072, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>
147 [pid 39420] <... mmap resumed>) = 0x7f63f4edc000
148 [pid 39421] <... mmap resumed>) = 0x7f5e83ca4000
149 [pid 39420] mmap(0x7f63f4ee2000, 53072, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>
150 [pid 39421] close(3 <unfinished ...>
151 [pid 39420] <... mmap resumed>) = 0x7f63f4ee2000
152 [pid 39421] <... close resumed>) = 0
153 [pid 39420] close(3 <unfinished ...>
154 [pid 39421] mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
155 [pid 39420] <... close resumed>) = 0
156 [pid 39421] <... mmap resumed>) = 0x7f5e83acd000
157 [pid 39420] mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>
158 [pid 39421] arch_prctl(ARCH_SET_FS, 0x7f5e83acd740 <unfinished ...>
159 [pid 39420] <... mmap resumed>) = 0x7f63f4d0b000
160 [pid 39421] <... arch_prctl resumed>) = 0
161 [pid 39420] arch_prctl(ARCH_SET_FS, 0x7f63f4d0b740 <unfinished ...>
162 [pid 39421] set_tid_address(0x7f5e83acda10 <unfinished ...>
163 [pid 39420] <... arch_prctl resumed>) = 0
164 [pid 39421] <... set_tid_address resumed>) = 39421
165 [pid 39420] set_tid_address(0x7f63f4d0ba10 <unfinished ...>
166 [pid 39421] set_robust_list(0x7f5e83acda20, 24 <unfinished ...>
167 [pid 39420] <... set_tid_address resumed>) = 39420
168 [pid 39421] <... set_robust_list resumed>) = 0
169 [pid 39420] set_robust_list(0x7f63f4d0ba20, 24 <unfinished ...>
170 [pid 39421] rseq(0x7f5e83ace060, 0x20, 0, 0x53053053 <unfinished ...>
171 [pid 39420] <... set_robust_list resumed>) = 0
172 [pid 39421] <... rseq resumed>) = 0
173 [pid 39420] rseq(0x7f63f4d0c060, 0x20, 0, 0x53053053) = 0
174 [pid 39421] mprotect(0x7f5e83c9e000, 16384, PROT_READ) = 0
175 [pid 39420] mprotect(0x7f63f4edc000, 16384, PROT_READ <unfinished ...>
176 [pid 39421] mprotect(0x403000, 4096, PROT_READ <unfinished ...>
177 [pid 39420] <... mprotect resumed>) = 0
178 [pid 39421] <... mprotect resumed>) = 0
179 [pid 39420] mprotect(0x403000, 4096, PROT_READ <unfinished ...>
180 [pid 39421] mprotect(0x7f5e83cea000, 8192, PROT_READ <unfinished ...>
181 [pid 39420] <... mprotect resumed>) = 0
182 [pid 39421] <... mprotect resumed>) = 0
183 [pid 39420] mprotect(0x7f63f4f28000, 8192, PROT_READ <unfinished ...>
184 [pid 39421] prlimit64(0, RLIMIT_STACK, NULL, <unfinished ...>
185 [pid 39420] <... mprotect resumed>) = 0
186 [pid 39421] <... prlimit64 resumed>{rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
187 [pid 39420] prlimit64(0, RLIMIT_STACK, NULL, <unfinished ...>
188 [pid 39421] munmap(0x7f5e83cb1000, 25258 <unfinished ...>
189 [pid 39420] <... prlimit64 resumed>{rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
190 [pid 39421] <... munmap resumed>) = 0
191 [pid 39420] munmap(0x7f63f4eef000, 25258 <unfinished ...>
192 [pid 39421] getrandom(<unfinished ...>
193 [pid 39420] <... munmap resumed>) = 0
194 [pid 39421] <... getrandom resumed>"\x96\x93\xe4\x42\x06\x7f\x20\xf0", 8,
GRND_NONBLOCK) = 8
195 [pid 39420] getrandom(<unfinished ...>
196 [pid 39421] brk(NULL <unfinished ...>

```

```

197 [pid 39420] <... getrandom resumed>"\xe0\x7c\x11\xa7\x85\x1a\xff\x02", 8,
GRND_NONBLOCK) = 8
198 [pid 39421] <... brk resumed>) = 0x1957000
199 [pid 39420] brk(NULL <unfinished ...>
200 [pid 39421] brk(0x1978000 <unfinished ...>
201 [pid 39420] <... brk resumed>) = 0x16ea000
202 [pid 39421] <... brk resumed>) = 0x1978000
203 [pid 39420] brk(0x170b000 <unfinished ...>
204 [pid 39421] openat(AT_FDCWD, "boba", O_WRONLY|O_CREAT|O_TRUNC, 0666
<unfinished ...>
205 [pid 39420] <... brk resumed>) = 0x170b000
206 [pid 39420] openat(AT_FDCWD, "biba", O_WRONLY|O_CREAT|O_TRUNC, 0666
<unfinished ...>
207 [pid 39421] <... openat resumed>) = 3
208 [pid 39421] newfstatat(0, "", {st_mode=S_IFIFO|0600, st_size=0, ...},
AT_EMPTY_PATH) = 0
209 [pid 39421] read(0, <unfinished ...>
210 [pid 39420] <... openat resumed>) = 3
211 [pid 39420] newfstatat(0, "", {st_mode=S_IFIFO|0600, st_size=0, ...},
AT_EMPTY_PATH) = 0
212 [pid 39420] read(0, "\n", 4096) = 1
213 [pid 39420] newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=0, ...},
AT_EMPTY_PATH) = 0
214 [pid 39420] read(0, sasasasasasasasasas
215 <unfinished ...>
216 [pid 39369] <... read resumed>"sasasasasasasasasas\n", 1024) = 20
217 [pid 39369] write(4, "sasasasasasasasasas\n", 20) = 20
218 [pid 39420] <... read resumed>"sasasasasasasasasas\n", 4096) = 20
219 [pid 39369] read(0, <unfinished ...>
220 [pid 39420] read(0, ababababababaaba
221 <unfinished ...>
222 [pid 39369] <... read resumed>"ababababababaaba\n", 1024) = 17
223 [pid 39369] write(4, "ababababababaaba\n", 17) = 17
224 [pid 39420] <... read resumed>"ababababababaaba\n", 4096) = 17
225 [pid 39369] read(0, <unfinished ...>
226 [pid 39420] read(0, lolololo
227 <unfinished ...>
228 [pid 39369] <... read resumed>"lolololo\n", 1024) = 9
229 [pid 39369] write(4, "lolololo\n", 9) = 9
230 [pid 39420] <... read resumed>"lolololo\n", 4096) = 9
231 [pid 39369] read(0, <unfinished ...>
232 [pid 39420] read(0, kkkkkkkkk
233 <unfinished ...>
234 [pid 39369] <... read resumed>"kkkkkkkkk\n", 1024) = 9
235 [pid 39369] write(4, "kkkkkkkkk\n", 9) = 9
236 [pid 39420] <... read resumed>"kkkkkkkkk\n", 4096) = 9
237 [pid 39369] read(0, <unfinished ...>
238 [pid 39420] read(0, <unfinished ...>
239 [pid 39369] <... read resumed>"", 1024) = 0
240 [pid 39369] close(4) = 0
241 [pid 39420] <... read resumed>"", 4096) = 0
242 [pid 39369] close(6) = 0
243 [pid 39421] <... read resumed>"", 4096) = 0
244 [pid 39369] wait4(39420, <unfinished ...>
245 [pid 39420] write(3, "\nssssssssss\nbbbbbbb\nllll\nkkkkkkkk"..., 34
<unfinished ...>
246 [pid 39421] close(3 <unfinished ...>
247 [pid 39420] <... write resumed>) = 34
248 [pid 39420] close(3 <unfinished ...>
249 [pid 39421] <... close resumed>) = 0
250 [pid 39421] exit_group(0) = ?
251 [pid 39420] <... close resumed>) = 0
252 [pid 39420] exit_group(0) = ?
253 [pid 39421] +++ exited with 0 +++
254 [pid 39369] <... wait4 resumed>NULL, 0, NULL) = ? ERESTARTSYS (To be
restarted if SA_RESTART is set)
255 [pid 39420] +++ exited with 0 +++

```

```
256 --- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=39421, si_uid=0,
si_status=0, si_utime=0, si_stime=0} ---
257 wait4(39420, NULL, 0, NULL)          = 39420
258 wait4(39421, NULL, 0, NULL)          = 39421
259 write(1, "\320\222\321\201\320\265
\320\277\321\200\320\276\321\206\320\265\321\201\321\201\321\213
\320\267\320\260\320\262\320\265"..., 44Все процессы завершены.
260 ) = 44
261 exit_group(0)                        = ?
262 +++ exited with 0 +++
```

## Вывод

Очень интересно было поработать с процессами и пайпами. Так же понял, что нужно быть очень внимательным с этими же пайпами и закрывать все ненужные, во избежание непредвиденных ситуаций. Самым сложным было держать в голове для какого процесса я сейчас пишу код и как он может повлиять на другие процессы, которые будут этот код выполнять.