

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: М80-206Б-22

Студент: Волков А.Д.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 06.10.2023

Москва, 2023

Постановка задачи

Вариант 14.

Child1 переводит строки в нижний регистр. Child2 убирает все задвоенные пробелы.

Общий метод и алгоритм решения

Использованные системные вызовы:

- `pid_t fork(void)`; – создает дочерний процесс.
- `int pipe(int *fd)`; – создает однонаправленный канал данных, который можно использовать для взаимодействия между процессами.
- `int dup2(int oldfd, int newfd)`; – создаёт копию файлового дескриптора `oldfd`, используя для нового дескриптора самый маленький свободный номер файлового дескриптора.
- `int close(int fd)`; – закрывает файловый дескриптор, который после этого не ссылается ни на один из файлов и может быть использован повторно.
- `ssize_t write(int fd, const void *buf, size_t count)`; – записывает `count` байт из буфера `buf` в файл, на который ссылается файловый дескриптор `fd`.
- `ssize_t read(int fd, void *buf, size_t count)`; – пытается записать `count` байт из файла, на который ссылается файловый дескриптор `fd`, в буфер, адрес которого начинается с `buf`.
- `int execl(const char *path, const char *arg, ...)`; – заменяет текущий образ процесса новым образом процесса.
- `void *malloc(size_t size)`; – распределяет `size` байт и возвращает указатель на распределенную память.

Для выполнения моего варианта лабораторной работы, я решил реализовать структуру статической строки, для удобства дальнейшего написания. Программа принимает эту строку из стандартного ввода в родительский процесс. Далее родительский процесс отправляет эту строку первому дочернему процессу, который ее принимает и обрабатывает в соответствии с условием варианта. Потом первый дочерний процесс отправляет через другой пайп строку второму дочернему процессу, который так же обрабатывает строку и отправляет ее через третий пайп родительскому процессу. Родительский процесс принимает эту строку и выводит ее в стандартный вывод, т.е. уже непосредственно пользователю.

Код программы

string.h

```
#include <stdio.h>
#include <stdlib.h>
```

```
#define DEFAULT 80
```

```
typedef struct {
    char *str;
    int length;
} my_string;
```

```
my_string *create_string();
void read_string(my_string *mstr);
```

```
void print_string(my_string *mstr);
```

mstring.c

```
#include "string.h"
```

```
my_string *create_string() {  
    my_string *tmp = (my_string *) malloc(sizeof(my_string));  
    tmp->str = (char *) malloc(sizeof(char) * DEFAULT);  
    tmp->length = 0; return tmp;  
}
```

```
void read_string(my_string *mstr) {  
    char c;  
    while ((c = getchar()) != 10) {  
        mstr->str[mstr->length] = c; mstr->length++;  
    }  
}
```

```
void print_string(my_string *mstr) {  
    for (int j = 0; j < mstr->length; j++) {  
        printf("%c", mstr->str[j]);  
    }  
    printf("\n");  
}
```

parent.c

```
#include <stdio.h>  
#include <stdlib.h>  
#include <unistd.h>  
#include <sys/wait.h>  
#include "string.h"
```

```
pid_t create_process() {  
    pid_t pid = fork();  
    if (-1 == pid) {  
        perror("fork");  
        exit(-1);  
    }  
    return pid;  
}
```

```
void my_dup(int old_fd, int new_fd) {  
    if (dup2(old_fd, new_fd) == -1) {  
        perror("dup2");  
        exit(-1);  
    }  
}
```

```
void my_pipe(int pipe_fd[]) {  
    int err = pipe(pipe_fd);
```

```

        if (-1 == err) {
            perror("pipe");
            exit(-1);
        }
    }
}

int main() {
    int pipe_fd1[2]; int pipe_fd2[2]; int pipe_fd3[2];
    my_pipe(pipe_fd1); my_pipe(pipe_fd2); my_pipe(pipe_fd3);
    pid_t cp1, cp2;
    if ((cp1 = create_process()) == 0) { //child1
        if (close(pipe_fd1[1]) == -1 || close(pipe_fd2[0]) == -1) {return -1;}
        my_dup(pipe_fd1[0], STDIN_FILENO); my_dup(pipe_fd2[1], STDOUT_FILENO);
        execl("../build/child1", "../build/child1", NULL);
        if (close(pipe_fd1[0]) == -1 || close(pipe_fd2[1]) == -1) {return -1;}
    } else if (cp1 > 0 && (cp2 = create_process()) == 0) { //child2
        if (close(pipe_fd2[1]) == -1 || close(pipe_fd3[0]) == -1) {return -1;}
        my_dup(pipe_fd2[0], STDIN_FILENO); my_dup(pipe_fd3[1], STDOUT_FILENO);
        execl("../build/child2", "../build/child2", NULL);
        if (close(pipe_fd2[0]) == -1 || close(pipe_fd3[1]) == -1) {return -1;}
    } else { //parent
        my_string *p_mstr = create_string();
        printf("Enter your string: "); read_string(p_mstr);
        if (close(pipe_fd1[0]) == -1 || close(pipe_fd3[1]) == -1) {return -1;}
        write(pipe_fd1[1], &(p_mstr->length), sizeof(int));
        write(pipe_fd1[1], p_mstr->str, sizeof(char) * p_mstr->length);
        read(pipe_fd3[0], &(p_mstr->length), sizeof(int));
        read(pipe_fd3[0], p_mstr->str, sizeof(char) * p_mstr->length);
        if (close(pipe_fd1[1]) == -1 || close(pipe_fd3[0]) == -1) {return -1;}
        printf("Result: "); print_string(p_mstr);
    }
    return 0;
}

```

child1.c

```

#include <unistd.h>
#include "string.h"

```

```

void tolowerc(my_string *mstr) {
    for (int i = 0; i < mstr->length; i++) {
        if (mstr->str[i] >= 'A' && mstr->str[i] <= 'Z') {
            mstr->str[i] += 32;
        }
    }
}

```

```

int main() {
    my_string *c1_mstr = create_string();
    read(STDIN_FILENO, &(c1_mstr->length), sizeof(int));
    read(STDIN_FILENO, c1_mstr->str, sizeof(char) * c1_mstr->length);
}

```

```

    tolower(c1_mstr);
    write(STDOUT_FILENO, &(amp;c1_mstr->length), sizeof(int));
    write(STDOUT_FILENO, c1_mstr->str, sizeof(char) * c1_mstr->length);
    close(STDIN_FILENO); close(STDOUT_FILENO);
    return 0;
}
child2.c
#include <unistd.h>
#include "string.h"

void delete_double_spaces(my_string *mstr) {
    int len = mstr->length;
    for (int i = 0; i < len; i++) {
        if (mstr->str[i] == 32 && mstr->str[i + 1] == 32) {
            mstr->str[i] = 0;
        }
    }
}

int main() {
    my_string *c2_mstr = create_string();
    read(STDIN_FILENO, &(amp;c2_mstr->length), sizeof(int));
    read(STDIN_FILENO, c2_mstr->str, sizeof(char) * c2_mstr->length);
    delete_double_spaces(c2_mstr);
    write(STDOUT_FILENO, &(amp;c2_mstr->length), sizeof(int));
    write(STDOUT_FILENO, c2_mstr->str, sizeof(char) * c2_mstr->length);
    close(STDIN_FILENO); close(STDOUT_FILENO);
    return 0;
}

```

Протокол работы программы

Тестирование:

```

$ ./parent
Enter your string: AAAabc   DDFfaA gg  TR
Result: aaabc ddffaa gg tr
$ ./parent
Enter your string:
Result:
$ ./parent
Enter your string: ABCD
Result: abcd
$ ./parent
Enter your string: aabbccdd
Result: aabbccdd
$ ./parent
Enter your string: ABcd
Result: abcd

```

```
$ ./parent
Enter your string: ABCD efgh IGKL mnop
Result: abcd efgh igkl mnop
```

Strace:

```
$ strace -f ./parent
execve("./parent", [ "./parent" ], 0x7ffdc17997a8 /* 76 vars */) = 0
brk(NULL)                                = 0x560b5e4e3000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fffd100cb90) = -1 EINVAL (Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f870e21a000
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=64135, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 64135, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f870e20a000
close(3)                                 = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
832 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0"... , 832) =
= 784 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)
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,
896) "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\360\352,\223\340."... , 68,
68
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2216304, ...}, 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, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f870de00000
mmap(0x7f870de28000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x28000) = 0x7f870de28000
mmap(0x7f870dfbd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1bd000) = 0x7f870dfbd000
mmap(0x7f870e015000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x214000) = 0x7f870e015000
mmap(0x7f870e01b000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f870e01b000
close(3)                                 = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f870e207000
arch_prctl(ARCH_SET_FS, 0x7f870e207740) = 0
set_tid_address(0x7f870e207a10)         = 8437
set_robust_list(0x7f870e207a20, 24)     = 0
rseq(0x7f870e2080e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f870e015000, 16384, PROT_READ) = 0
mprotect(0x560b5d8ee000, 4096, PROT_READ) = 0
mprotect(0x7f870e254000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
```

```

munmap(0x7f870e20a000, 64135)          = 0
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 8438 attached
, child_tidptr=0x7f870e207a10) = 8438
[pid 8438] set_robust_list(0x7f870e207a20, 24 <unfinished ...>
[pid 8437] clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD <unfinished ...>
strace: Process 8439 attached
[pid 8438] <... set_robust_list resumed>) = 0
[pid 8438] close(4 <unfinished ...>
[pid 8437] <... clone resumed>, child_tidptr=0x7f870e207a10) = 8439
[pid 8439] set_robust_list(0x7f870e207a20, 24 <unfinished ...>
[pid 8438] <... close resumed>)          = 0
[pid 8437] getrandom( <unfinished ...>
[pid 8439] <... set_robust_list resumed>) = 0
[pid 8437] <... getrandom resumed>"\x23\x51\xd7\x1d\x60\x00\x11\x73", 8,
GRND_NONBLOCK) = 8
[pid 8438] close(5 <unfinished ...>
[pid 8439] close(6 <unfinished ...>
[pid 8437] brk(NULL <unfinished ...>
[pid 8439] <... close resumed>)          = 0
[pid 8438] <... close resumed>)          = 0
[pid 8437] <... brk resumed>)            = 0x560b5e4e3000
[pid 8439] close(7 <unfinished ...>
[pid 8438] dup2(3, 0 <unfinished ...>
[pid 8437] brk(0x560b5e504000 <unfinished ...>
[pid 8439] <... close resumed>)          = 0
[pid 8437] <... brk resumed>)            = 0x560b5e504000
[pid 8438] <... dup2 resumed>)            = 0
[pid 8439] dup2(5, 0 <unfinished ...>
[pid 8437] newfstatat(1, "", <unfinished ...>
[pid 8439] <... dup2 resumed>)            = 0
[pid 8437] <... newfstatat resumed>{st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0),
...}, AT_EMPTY_PATH) = 0
[pid 8438] dup2(6, 1 <unfinished ...>
[pid 8437] newfstatat(0, "", <unfinished ...>
[pid 8439] dup2(8, 1 <unfinished ...>
[pid 8437] <... newfstatat resumed>{st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0),
...}, AT_EMPTY_PATH) = 0
[pid 8439] <... dup2 resumed>)            = 1
[pid 8438] <... dup2 resumed>)            = 1
[pid 8437] write(1, "Enter your string: ", 19 <unfinished ...>
Enter your string: [pid 8439] execve("../build/child2", ["../build/child2"],
0x7fffd100cd68 /* 76 vars */ <unfinished ...>
[pid 8437] <... write resumed>)            = 19
[pid 8438] execve("../build/child1", ["../build/child1"], 0x7fffd100cd68 /* 76 vars */

```

```

<unfinished ...>
[pid 8437] read(0, <unfinished ...>
[pid 8439] <... execve resumed>)          = 0
[pid 8439] brk(NULL)                      = 0x55dfbfdaf000
[pid 8439] arch_prctl(0x3001 /* ARCH_??? */ , 0x7ffe5ec3a8d0) = -1 EINVAL
(Недопустимый
аргумент)
[pid 8438] <... execve resumed>)          = 0
[pid 8439] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
[pid 8438] brk(NULL <unfinished ...>
[pid 8439] <... mmap resumed>)            = 0x7f7ad39a3000
[pid 8438] <... brk resumed>)              = 0x5574db1a0000
[pid 8439] access("/etc/ld.so.preload", R_OK) = -1 ENOENT (Нет такого файла или
каталога)
[pid 8438] arch_prctl(0x3001 /* ARCH_??? */ , 0x7ffd28fe1320 <unfinished ...>
[pid 8439] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>
[pid 8438] <... arch_prctl resumed>)      = -1 EINVAL (Недопустимый аргумент)
[pid 8439] <... openat resumed>)          = 6
[pid 8439] newfstatat(6, "", <unfinished ...>
[pid 8438] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
[pid 8439] <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=64135, ...},
AT_EMPTY_PATH) = 0
[pid 8438] <... mmap resumed>)            = 0x7fac604f4000
[pid 8439] mmap(NULL, 64135, PROT_READ, MAP_PRIVATE, 6, 0 <unfinished ...>
[pid 8438] access("/etc/ld.so.preload", R_OK <unfinished ...>
[pid 8439] <... mmap resumed>)            = 0x7f7ad3993000
[pid 8438] <... access resumed>)          = -1 ENOENT (Нет такого файла или каталога)
[pid 8439] close(6 <unfinished ...>
[pid 8438] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>
[pid 8439] <... close resumed>)           = 0
[pid 8438] <... openat resumed>)          = 4
[pid 8439] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC
<unfinished ...>
[pid 8438] newfstatat(4, "", <unfinished ...>
[pid 8439] <... openat resumed>)          = 6
[pid 8438] <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=64135, ...},
AT_EMPTY_PATH) = 0
[pid 8439] read(6, <unfinished ...>
[pid 8438] mmap(NULL, 64135, PROT_READ, MAP_PRIVATE, 4, 0 <unfinished ...>
[pid 8439] <... read
832 resumed>"\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\0"... , 832) =
[pid 8438] <... mmap resumed>)            = 0x7fac604e4000
[pid 8439] pread64(6, <unfinished ...>
[pid 8438] close(4 <unfinished ...>
[pid 8439] <... pread64
784 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) =
[pid 8438] <... close resumed>)           = 0

```



```

[pid 8439] pread64(6, "\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

[pid 8438] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC
<unfinished ...>
[pid 8439] pread64(6, <unfinished ...>
[pid 8438] <... openat resumed>) = 4
[pid 8439] <... pread64
68, resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\360\352,\223\340."...,
896) = 68
[pid 8438] read(4, <unfinished ...>
[pid 8439] newfstatat(6, "", <unfinished ...>
[pid 8438] <... read
832 resumed>"\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) =
[pid 8439] <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=2216304, ...},
AT_EMPTY_PATH) = 0
[pid 8438] pread64(4, <unfinished ...>
[pid 8439] pread64(6, <unfinished ...>
[pid 8438] <... pread64
784 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) =
[pid 8439] <... pread64
784 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) =
[pid 8438] pread64(4, <unfinished ...>
...> [pid 8439] mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 6, 0 <unfinished
[pid 8438] <... 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
[pid 8439] <... mmap resumed>) = 0x7f7ad3600000
[pid 8438] pread64(4, <unfinished ...>
[pid 8439] mmap(0x7f7ad3628000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 0x28000 <unfinished ...>
[pid 8438] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\360\352,\223\340."..., 68,
896) = 68
[pid 8439] <... mmap resumed>) = 0x7f7ad3628000
[pid 8438] newfstatat(4, "", <unfinished ...>
[pid 8439] mmap(0x7f7ad37bd000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 0x1bd000 <unfinished ...>
[pid 8438] <... newfstatat resumed>{st_mode=S_IFREG|0644, st_size=2216304, ...},
AT_EMPTY_PATH) = 0
[pid 8439] <... mmap resumed>) = 0x7f7ad37bd000
[pid 8438] pread64(4, <unfinished ...>
[pid 8439] mmap(0x7f7ad3815000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 6, 0x214000 <unfinished ...>
[pid 8438] <... pread64
784 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) =
[pid 8439] <... mmap resumed>) = 0x7f7ad3815000

```

```

...> [pid 8438] mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 4, 0 <unfinished
[pid 8439] mmap(0x7f7ad381b000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 8438] <... mmap resumed>) = 0x7fac60200000
[pid 8439] <... mmap resumed>) = 0x7f7ad381b000
[pid 8438] mmap(0x7fac60228000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x28000 <unfinished ...>
[pid 8439] close(6) = 0
[pid 8438] <... mmap resumed>) = 0x7fac60228000
[pid 8439] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
[pid 8438] mmap(0x7fac603bd000, 360448, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x1bd000 <unfinished ...>
[pid 8439] <... mmap resumed>) = 0x7f7ad3990000
[pid 8438] <... mmap resumed>) = 0x7fac603bd000
[pid 8438] mmap(0x7fac60415000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x214000 <unfinished ...>
[pid 8439] arch_prctl(ARCH_SET_FS, 0x7f7ad3990740) = 0
[pid 8439] set_tid_address(0x7f7ad3990a10) = 8439
[pid 8439] set_robust_list(0x7f7ad3990a20, 24) = 0
[pid 8439] rseq(0x7f7ad39910e0, 0x20, 0, 0x53053053) = 0
[pid 8438] <... mmap resumed>) = 0x7fac60415000
[pid 8438] mmap(0x7fac6041b000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 8439] mprotect(0x7f7ad3815000, 16384, PROT_READ <unfinished ...>
[pid 8438] <... mmap resumed>) = 0x7fac6041b000
[pid 8439] <... mprotect resumed>) = 0
[pid 8439] mprotect(0x55dfbf3df000, 4096, PROT_READ <unfinished ...>
[pid 8438] close(4 <unfinished ...>
[pid 8439] <... mprotect resumed>) = 0
[pid 8438] <... close resumed>) = 0
[pid 8439] mprotect(0x7f7ad39dd000, 8192, PROT_READ <unfinished ...>
[pid 8438] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
[pid 8439] <... mprotect resumed>) = 0
[pid 8438] <... mmap resumed>) = 0x7fac604e1000
[pid 8439] prlimit64(0, RLIMIT_STACK, NULL, <unfinished ...>
[pid 8438] arch_prctl(ARCH_SET_FS, 0x7fac604e1740 <unfinished ...>
[pid 8439] <... prlimit64 resumed>{rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
[pid 8438] <... arch_prctl resumed>) = 0
[pid 8439] munmap(0x7f7ad3993000, 64135 <unfinished ...>
[pid 8438] set_tid_address(0x7fac604e1a10) = 8438
[pid 8439] <... munmap resumed>) = 0
[pid 8438] set_robust_list(0x7fac604e1a20, 24) = 0
[pid 8439] getrandom( <unfinished ...>
[pid 8438] rseq(0x7fac604e20e0, 0x20, 0, 0x53053053 <unfinished ...>
[pid 8439] <... getrandom resumed>"\xa5\x4a\xe1\x27\x39\xbb\x9a\x51", 8,
GRND_NONBLOCK) = 8
[pid 8439] brk(NULL <unfinished ...>

```

```

[pid 8438] <... rseq resumed>                = 0
[pid 8439] <... brk resumed>                  = 0x55dfbfdaf000
[pid 8439] brk(0x55dfbfdd0000)                = 0x55dfbfdd0000
[pid 8439] read(0, <unfinished ...>
[pid 8438] mprotect(0x7fac60415000, 16384, PROT_READ) = 0
[pid 8438] mprotect(0x5574da111000, 4096, PROT_READ) = 0
[pid 8438] mprotect(0x7fac6052e000, 8192, PROT_READ) = 0
[pid 8438] prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
    rlim_max=RLIM64_INFINITY}) = 0
[pid 8438] munmap(0x7fac604e4000, 64135) = 0
[pid 8438] getrandom("\x0c\xb3\x61\xbf\xf8\x65\xd9\xfc", 8, GRND_NONBLOCK) = 8
[pid 8438] brk(NULL)                        = 0x5574db1a0000
[pid 8438] brk(0x5574db1c1000)              = 0x5574db1c1000
[pid 8438] read(0, AbcD fghI
    <unfinished ...>
[pid 8437] <... read resumed>"AbcD fghI\n", 1024) = 11
[pid 8437] close(3)                        = 0
[pid 8437] close(8)                        = 0
[pid 8437] write(4, "\n\0\0\0", 4)        = 4
[pid 8438] <... read resumed>"\n\0\0\0", 4) = 4
[pid 8437] write(4, "AbcD fghI", 10 <unfinished ...>
[pid 8438] read(0, <unfinished ...>
[pid 8437] <... write resumed>              = 10
[pid 8438] <... read resumed>"AbcD fghI", 10) = 10
[pid 8437] read(7, <unfinished ...>
[pid 8438] write(1, "\n\0\0\0", 4 <unfinished ...>
[pid 8439] <... read resumed>"\n\0\0\0", 4) = 4
[pid 8438] <... write resumed>              = 4
[pid 8439] read(0, <unfinished ...>
[pid 8438] write(1, "abcd fghi", 10) = 10
[pid 8439] <... read resumed>"abcd fghi", 10) = 10
[pid 8439] write(1, "\n\0\0\0", 4 <unfinished ...>
[pid 8438] close(0 <unfinished ...>
[pid 8437] <... read resumed>"\n\0\0\0", 4) = 4
[pid 8439] <... write resumed>              = 4
[pid 8438] <... close resumed>              = 0
[pid 8437] read(7, <unfinished ...>
[pid 8439] write(1, "abcd\0 fghi", 10 <unfinished ...>
[pid 8438] close(1 <unfinished ...>
[pid 8437] <... read resumed>"abcd\0 fghi", 10) = 10
[pid 8439] <... write resumed>              = 10
[pid 8437] close(4 <unfinished ...>
[pid 8439] close(0 <unfinished ...>
[pid 8438] <... close resumed>              = 0
[pid 8437] <... close resumed>              = 0
[pid 8439] <... close resumed>              = 0
[pid 8439] close(1 <unfinished ...>
[pid 8438] exit_group(0 <unfinished ...>
[pid 8437] close(7 <unfinished ...>

```

```
[pid 8439] <... close resumed>          = 0
[pid 8438] <... exit_group resumed>     = ?
[pid 8437] <... close resumed>          = 0
[pid 8437] write(1, "Result: abcd\0 fghi\n", 19 <unfinished ...>
[pid 8439] exit_group(0Result: abcd fghi
<unfinished ...>
[pid 8437] <... write resumed>          = 19
[pid 8439] <... exit_group resumed>     = ?
[pid 8437] exit_group(0)                = ?
[pid 8438] +++ exited with 0 +++
[pid 8439] +++ exited with 0 +++
+++ exited with 0 +++
```

Вывод

В ходе выполнения лабораторной работы я начал понимать как работают процессы в Linux, их иерархию, общение и т.д. Также вспомнил как работать со строками и памятью, после летнего перерыва. Были три самые большие сложности во время выполнения лабораторной работы:

1. Незнание того как работают процессы
2. Незнание как манипулировать процессами через призму ЯП Си
3. Сложности с передачей структур через каналы общения процессов!!!