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-214Б-23

Студент: Гусев С.В.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 28.10.24

Москва, 2024

Постановка задачи

Вариант 2.

Пользователь вводит команды вида: «число число число». Далее эти числа передаются от родительского процесса в дочерний. Дочерний процесс считает их сумму и выводит её в файл. Числа имеют тип float. Количество чисел может быть произвольным.

Общий метод и алгоритм решения

Использованные системные вызовы:

- int channel[2];
  pipe(channel); – создает два канала связи.
- const pid_t child = fork(); – создает дочерний процесс.
- pid_t pid = getpid(); – получает номер текущего процесса.
- dup2(STDIN_FILENO, channel[STDIN_FILENO]); – перенаправляет стандартный ввод на дескриптор канала связи.
- int32_t status = execv(path, args); – ● заменяет код новым программным кодом, указанным в path.
- wait(&child_status); – родительский процесс ждет завершения дочернего процесса.

Решение:

1. Обрабатываю путь переданный через аргументы командной строки.
2. С помощью функций написанных выше связываю родительский процесс с дочерним(передаю обработанный ввод).
3. В дочернем процессе считываю строку (используя read()) и проверяю находящиеся там символы на соответствие вводу указанному в задании («число число число»), параллельно заменяя знаки пробела на ‘\0’(дабы потом удобно было переводить в числа типа float).
4. Циклом прохожу по строке и складываю числа (переводя их в тип float с помощью atof).
5. Ответ вывожу в канал связи с родительским процессом (использую write()).

Код программы

server.c

```c
#include <stdint.h>
#include <stdbool.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>
#include <stdio.h>


static char CLIENT_PROGRAM_NAME[] = "client";

int main(int argc, char **argv) {
    if (argc == 1) {
        char msg[1024];
        uint32_t len = snprintf(msg, sizeof(msg) - 1, "usage: %s filename\n", argv[0]);
        write(STDERR_FILENO, msg, len);
```

```
        exit(EXIT_SUCCESS);
    }

    char progpath[1024];
    {
        ssize_t len = readlink("/proc/self/exe", progpath, sizeof(progpath) - 1);
        if (len == -1) {
            const char msg[] = "error: failed to read full program path\n";
            write(STDERR_FILENO, msg, sizeof(msg));
            exit(EXIT_FAILURE);
        }

        while (progpath[len] != '/')
            --len;

        progpath[len] = '\0';
    }

    char buf[4096];
    ssize_t bytes;

    int channel[2];
    if (pipe(channel) == -1) {
        const char msg[] = "error: failed to create pipe\n";
        write(STDERR_FILENO, msg, sizeof(msg));
        exit(EXIT_FAILURE);
    }

    const pid_t child = fork();

    switch (child) {
        case -1: {
            const char msg[] = "error: failed to spawn new process\n";
            write(STDERR_FILENO, msg, sizeof(msg));
            exit(EXIT_FAILURE);
        } break;

        case 0: {
            pid_t pid = getpid();
//            dup2(STDIN_FILENO, channel[STDIN_FILENO]);
            dup2(channel[STDIN_FILENO], STDIN_FILENO);
//            close(channel[STDOUT_FILENO]);

            {
                char msg[64];
                const int32_t length = snprintf(msg, sizeof(msg), "%d: I'm a child\n",
pid);
                write(STDOUT_FILENO, msg, length);
            }

            {
                char path[1024];
                snprintf(path, sizeof(path) - 1, "%s/%s", progpath, CLIENT_PROGRAM_NAME);
                char *const args[] = {CLIENT_PROGRAM_NAME, argv[1], NULL};
                int32_t status = execv(path, args);
                if (status == -1) {
```

```
                    const char msg[] = "error: failed to exec into new executable
image\n";
                    write(STDERR_FILENO, msg, sizeof(msg));
                    exit(EXIT_FAILURE);
                }
            }
        } break;

        default: {
            pid_t pid = getpid();

            {
                char msg[64];
                const int32_t length = snprintf(msg, sizeof(msg), "%d: I'm a parent, my
child has PID %d\n", pid, child);
                write(STDOUT_FILENO, msg, length);
            }

            while ((bytes = read(STDIN_FILENO, buf, sizeof(buf) - 1)) > 0) {
                if (bytes == 1 && buf[0] == '\n') {
                    exit(EXIT_SUCCESS);
                }
                write(channel[STDOUT_FILENO], buf, bytes);
            }

            int child_status;
            wait(&child_status);

            if (child_status != EXIT_SUCCESS) {
                const char msg[] = "error: child exited with error\n";
                write(STDERR_FILENO, msg, sizeof(msg));
                exit(child_status);
            }
        } break;
    }
}
```

client.c

```
#include <stdint.h>

#include <stdbool.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <ctype.h>

#include <stdio.h>


int main(int argc, char **argv) {
```

```
    char buf[4096];
    ssize_t bytes;
    char ans[4096];

    pid_t pid = getpid();

    int32_t file = open(argv[1], O_WRONLY | O_CREAT | O_TRUNC | O_APPEND, 0600);
    if (file == -1) {
        const char msg[] = "error: failed to open requested file\n";
        write(STDERR_FILENO, msg, sizeof(msg));
        exit(EXIT_FAILURE);
    }

    {
        char msg[128];
        int32_t len = snprintf(msg, sizeof(msg) - 1,
                               "%d: Start typing lines of text. Press 'Ctrl-D' or
'Enter' with no input to exit\n", pid);
        write(STDOUT_FILENO, msg, len);
    }

    while ((bytes = read(STDIN_FILENO, buf, sizeof(buf)))) {
        float sum = 0;
        if (bytes < 0) {
            const char msg[] = "error: failed to read from stdin\n";
            write(STDERR_FILENO, msg, sizeof(msg));
            exit(EXIT_FAILURE);
        } else if (buf[0] == '\n') {
            break;
        }

        {
            char msg[32];
```

```
    int32_t len = snprintf(msg, sizeof(msg) - 1,
                           "Sum of your numbers: ");

    int32_t written = write(file, msg, len);
    if (written != len) {
        const char msg[] = "error: failed to write to file\n";
        write(STDERR_FILENO, msg, sizeof(msg));
        exit(EXIT_FAILURE);
    }
}

{
    buf[bytes] = '\0';
    int point_cnt = 0;
    int numb_cnt = 1;
    for (int i = 0; i < bytes - 1; ++i) {
        if (isdigit(buf[i]) || (buf[i] == '.' && !point_cnt)) {
            if (buf[i] == '.') point_cnt++;
            continue;
        }
        if (buf[i] == ' ') {
            point_cnt = 0;
            buf[i] = '\0';
            continue;
        }
        const char msg[] = "error: value is not a number\n";
        write(STDERR_FILENO, msg, sizeof(msg));
        exit(EXIT_FAILURE);
    }
}

{
    char *ptr = buf;
```

```
                float numb = 0;
                sum += atof(ptr);
                for(int i = 0; i < bytes - 1; ++i) {
                    if (buf[i] == '\0' && bytes > i + 1) {
                        numb = atof(ptr + i + 1);
                        sum += numb;
                    }
                }

                size_t ansLen = snprintf(ans, sizeof(ans), "%.5f\n", sum);
                int32_t written = write(file, ans, ansLen);
                if (written != ansLen) {
                    const char msg[] = "error: failed to write to file\n";
                    write(STDERR_FILENO, msg, sizeof(msg));
                    exit(EXIT_FAILURE);
                }
            }
        }
        const char term = '\0';
        write(file, &term, sizeof(term));

        close(file);
}
```

Протокол работы программы

Тестирование:

```
$ ./server filename.txt
633: I'm a parent, my child has PID 634
634: I'm a child
634: Start typing lines of text. Press 'Ctrl-D' or 'Enter' with no input to exit
1.2 1.3 1.5
1.2345 53.124 8911.132
0.00134 0.12209 0.00252 0.01919 1.12602
```

```
1234 3567 2378.541 5678 7890 6.5678 45.45678

181

0.678 567.672


$ cat filename.txt

Sum of your numbers: 4.00000

Sum of your numbers: 8965.49023

Sum of your numbers: 1.27116

Sum of your numbers: 20799.56641

Sum of your numbers: 181.00000

Sum of your numbers: 568.34998
```

Strace:

```
$ strace ./server file

execve("./server", ["./server", "file"], 0x7ffd37db8138 /* 28 vars */) = 0

brk(NULL)                               = 0x556a7935b000

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7ff87f1d5000

access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=20415, ...}) = 0

mmap(NULL, 20415, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7ff87f1d0000

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\0\220\243\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

fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff87efbe000

mmap(0x7ff87efe6000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x28000) = 0x7ff87efe6000
```

```
     mmap(0x7ff87f16e000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x7ff87f16e000

     mmap(0x7ff87f1bd000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1fe000) = 0x7ff87f1bd000

     mmap(0x7ff87f1c3000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7ff87f1c3000

     close(3)                                = 0

     mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7ff87efbb000

     arch_prctl(ARCH_SET_FS, 0x7ff87efbb740) = 0

     set_tid_address(0x7ff87efbba10)         = 51243

     set_robust_list(0x7ff87efbba20, 24)     = 0

     rseq(0x7ff87efbc060, 0x20, 0, 0x53053053) = 0

     mprotect(0x7ff87f1bd000, 16384, PROT_READ) = 0

     mprotect(0x556a5ca84000, 4096, PROT_READ) = 0

     mprotect(0x7ff87f20d000, 8192, PROT_READ) = 0

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

     munmap(0x7ff87f1d0000, 20415)           = 0

     readlink("/proc/self/exe", "/mnt/c/Users/\320\232\321\200\321\217/CLionProject"...,
1023) = 49

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

     clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7ff87efbba10) = 51244

     51244: I'm a child

     getpid()                                = 51243

     write(1, "51243: I'm a parent, my child ha"..., 4451243: I'm a parent, my child has PID
51244

     ) = 44

     wait4(-1, 51244: Start typing lines of text. Press 'Ctrl-D' or 'Enter' with no input to
exit

     1.2 1.3 1.5

     1.2345 53.124 8911.132

     0.00134 0.12209 0.00252 0.01919 1.12602

     1234 3567 2378.541 5678 7890 6.5678 45.45678

     181

     0.678 567.672
```

```
[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 51244

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=51244, si_uid=1000,
si_status=0, si_utime=0, si_stime=0} ---

exit_group(0)                           = ?

+++ exited with 0 +++

inkawy@DESKTOP-KBU2DKS:/mnt/c/Users/Кря/CLionProjects/OSy/Lab1$ strace -f ./server
file

execve("./server", ["./server", "file"], 0x7fffc1d771c0 /* 28 vars */) = 0

brk(NULL)                               = 0x55d5b8354000

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fb6f9b6f000

access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=20415, ...}) = 0

mmap(NULL, 20415, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fb6f9b6a000

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\0\220\243\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

fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 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, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fb6f9958000

mmap(0x7fb6f9980000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x28000) = 0x7fb6f9980000

mmap(0x7fb6f9b08000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x7fb6f9b08000

mmap(0x7fb6f9b57000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1fe000) = 0x7fb6f9b57000

mmap(0x7fb6f9b5d000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7fb6f9b5d000

close(3)                                = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fb6f9955000
```

```
arch_prctl(ARCH_SET_FS, 0x7fb6f9955740) = 0

set_tid_address(0x7fb6f9955a10)         = 51470

set_robust_list(0x7fb6f9955a20, 24)     = 0

rseq(0x7fb6f9956060, 0x20, 0, 0x53053053) = 0

mprotect(0x7fb6f9b57000, 16384, PROT_READ) = 0

mprotect(0x55d59060f000, 4096, PROT_READ) = 0

mprotect(0x7fb6f9ba7000, 8192, PROT_READ) = 0

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

munmap(0x7fb6f9b6a000, 20415)           = 0

readlink("/proc/self/exe", "/mnt/c/Users/\320\232\321\200\321\217/CLionProject"...,
1023) = 49

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

clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLDstrace:
Process 51471 attached

, child_tidptr=0x7fb6f9955a10) = 51471

[pid 51471] set_robust_list(0x7fb6f9955a20, 24 <unfinished ...>

[pid 51470] getpid( <unfinished ...>

[pid 51471] <... set_robust_list resumed>) = 0

[pid 51470] <... getpid resumed>)       = 51470

[pid 51470] write(1, "51470: I'm a parent, my child ha"..., 44 <unfinished ...>

51470: I'm a parent, my child has PID 51471

[pid 51471] getpid( <unfinished ...>

[pid 51470] <... write resumed>)        = 44

[pid 51471] <... getpid resumed>)       = 51471

[pid 51470] wait4(-1,  <unfinished ...>

[pid 51471] dup2(0, 3)                  = 3

[pid 51471] close(4)                    = 0

[pid 51471] write(1, "51471: I'm a child\n", 1951471: I'm a child

) = 19

[pid 51471]
execve("/mnt/c/Users/\320\232\321\200\321\217/CLionProjects/OSy/Lab1/client", ["client",
"file"], 0x7ffd0c770070 /* 28 vars */) = 0

[pid 51471] brk(NULL)                   = 0x560f28737000

[pid 51471] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f9eec5da000
```

```
[pid 51471] access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)

[pid 51471] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 4

[pid 51471] fstat(4, {st_mode=S_IFREG|0644, st_size=20415, ...}) = 0

[pid 51471] mmap(NULL, 20415, PROT_READ, MAP_PRIVATE, 4, 0) = 0x7f9eec5d5000

[pid 51471] close(4)                    = 0

[pid 51471] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 4

[pid 51471] read(4,
"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"..., 832) = 832

[pid 51471] pread64(4,
"\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 51471] fstat(4, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

[pid 51471] pread64(4,
"\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 51471] mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 4, 0) =
0x7f9eec3c3000

[pid 51471] mmap(0x7f9eec3eb000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x28000) = 0x7f9eec3eb000

[pid 51471] mmap(0x7f9eec573000, 323584, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x1b0000) = 0x7f9eec573000

[pid 51471] mmap(0x7f9eec5c2000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x1fe000) = 0x7f9eec5c2000

[pid 51471] mmap(0x7f9eec5c8000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f9eec5c8000

[pid 51471] close(4)                    = 0

[pid 51471] mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f9eec3c0000

[pid 51471] arch_prctl(ARCH_SET_FS, 0x7f9eec3c0740) = 0

[pid 51471] set_tid_address(0x7f9eec3c0a10) = 51471

[pid 51471] set_robust_list(0x7f9eec3c0a20, 24) = 0

[pid 51471] rseq(0x7f9eec3c1060, 0x20, 0, 0x53053053) = 0

[pid 51471] mprotect(0x7f9eec5c2000, 16384, PROT_READ) = 0

[pid 51471] mprotect(0x560efbffc000, 4096, PROT_READ) = 0

[pid 51471] mprotect(0x7f9eec612000, 8192, PROT_READ) = 0

[pid 51471] prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0

[pid 51471] munmap(0x7f9eec5d5000, 20415) = 0

[pid 51471] getpid()                    = 51471
```

```
[pid 51471] openat(AT_FDCWD, "file", O_WRONLY|O_CREAT|O_TRUNC|O_APPEND, 0600) = 4

[pid 51471] write(1, "51471: Start typing lines of tex"..., 8351471: Start typing lines
of text. Press 'Ctrl-D' or 'Enter' with no input to exit

) = 83

[pid 51471] read(0, 1.2 1.3 1.5

1.23"1.2 1.3 1.5\n", 4096) = 12

45 53.124 8911.1[pid 51471] write(4, "Sum of your numbers: ", 2132

0.00134 0.12209 0.00252 0.019) = 21

19 1.12602

1234 [pid 51471] write(4, "4.00000\n", 83567 2378.541 56)    = 8

78 7890 [pid 51471] read(0, 6.5678 4"1.2345 53.124 8911.132\n", 4096) = 23

5.45678

[pid 51471] write(4, "Sum of your numbers: ", 21181

0.678 567.67) = 21

[pid 51471] write(4, "8965.49023\n", 11) = 11

[pid 51471] read(0, "0.00134 0.12209 0.00252 0.01919 "..., 4096) = 40

[pid 51471] write(4, "Sum of your numbers: ", 21) = 21

[pid 51471] write(4, "1.27116\n", 82)    = 8

[pid 51471] read(0, "1234 3567 2378.541 5678 7890 6.5"..., 4096) = 45

[pid 51471] write(4, "Sum of your numbers: ", 21) = 21

[pid 51471] write(4, "20799.56641\n", 12) = 12

[pid 51471] read(0, "181\n", 4096)      = 4

[pid 51471] write(4, "Sum of your numbers: ", 21) = 21

[pid 51471] write(4, "181.00000\n", 10) = 10

[pid 51471] read(0,

"0.678 567.672\n", 4096) = 14

[pid 51471] write(4, "Sum of your numbers: ", 21) = 21

[pid 51471] write(4, "568.34998\n", 10) = 10

[pid 51471] read(0,

"\n", 4096)         = 1

[pid 51471] write(4, "\0", 1)          = 1

[pid 51471] close(4)                   = 0

[pid 51471] exit_group(0)              = ?
```

```
[pid 51471] +++ exited with 0 +++

<... wait4 resumed>[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 51471

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=51471, si_uid=1000,
si_status=0, si_utime=0, si_stime=0} ---

exit_group(0)                           = ?

+++ exited with 0 +++
```

Вывод

В результате выполнения лабораторной работы удалось познакомиться с системными вызовами (такими как pipe(), fork(), dup2(), execv(), wait()) и реализовать программу сложения нескольких чисел записанных в строку через пробел. Проблем при выполнении работы не возникло.