

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-209Б-24

Студент: Буланов М.С.

Преподаватель: Тихонов Ф.А

Оценка: \_\_\_\_\_

Дата: 4.11.25

Москва, 2025

## Постановка задачи

Вариант 3.

### Общий метод и алгоритм решения

Использованные системные вызовы:

- pid\_t fork(void); – создает дочерний процесс.
- int pipe(int \*fd); – создание пайпа
- exit(EXIT\_FAILURE) – завершение процесса с кодом
- close() – закрытие одного из концов пайпа
- 

**Написана программа создающая дочерний процесс, в рамках которого происходит замена исполняемого кода, открываются анонимные пайпы. Родительский процесс получает строку , преобразует в массив чисел. Дочерний процесс считает их и выводит результаты в файл.**

### Код программы

#### Код программы в папке с докладом

### Протокол работы программы

Тестирование:

```
root@bb4442151ed3:/workspaces/Operating_Systems/lab1/build# ./parent
```

Введите имя файла для дочернего процесса: ww

Дочерний процесс создан. Вводите команды (числа через пробел):

```
> 16 4 2 2
```

Дочерний процесс: Дочерний процесс готов. Файл: ww

```
> 2
```

Дочерний процесс: OK: Command processed successfully

```
> 8 0
```

Дочерний процесс: ERROR: Need at least 2 numbers

```
>
```

```
> 2
```

```
root@bb4442151ed3:/workspaces/Operating_Systems/lab1/build# cat ww
```

```
== Начало работы дочернего процесса ==
```

Входные данные: 16 4 2 2

Результаты деления:

```
16 / 4 = 4.00
```

```
16 / 2 = 8.00
```

```
16 / 2 = 8.00
```

```
---
```

Ошибка: нужно как минимум 2 числа

Входные данные: 8 0

## Результаты деления:

ОШИБКА: деление на ноль (8 / 0)

- - -

==== Дочерний процесс завершен ===

**Strace:**

```
munmap(0x7fb9a85c2000, 33103)          = 0
pipe([3, 4])                          = 0
pipe([5, 6])                          = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
brk(NULL)                            = 0x1f5b000
brk(0x1f7c000)                      = 0x1f7c000
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
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"..., 77Введите имя файла для дочернего процесса: ) = 77
read(0, ww
"ww\n", 1024)                      = 3
clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fb9a85c1810) = 5205
close(3)                            = 0
close(6)                            = 0
write(1, "\320\224\320\276\321\207\320\265\321\200\320\275\320\270\320\271
\320\277\321\200\320\276\321\206\320\265\321\201\321\201 "..., 114Дочерний процесс создан.
Вводите команды (числа через пробел):
) = 114
write(1, "> ", 2>)                  = 2
read(0, 8 2 1
"8 2 1\n", 1024)                  = 6
write(4, "8 2 1\n", 6)                = 6
wait4(5205, 0x7ffd22bdf6c, WNOHANG, NULL) = 0
select(6, [5], NULL, NULL, {tv_sec=0, tv_usec=100000}) = 1 (in [5], left {tv_sec=0,
tv_usec=99998})
read(5, "\320\224\320\276\321\207\320\265\321\200\320\275\320\270\320\271
\320\277\321\200\320\276\321\206\320\265\321\201\321\201 "..., 1023) = 92
write(1, "\320\224\320\276\321\207\320\265\321\200\320\275\320\270\320\271
\320\277\321\200\320\276\321\206\320\265\321\201\321\201:"..., 125Дочерний процесс: Дочерний
процесс готов. Файл: ww
OK: Command processed successfully
) = 125
write(1, "> ", 2>)                  = 2
read(0, 2 0
```

```
"2 0 \n", 1024) = 5
write(4, "2 0 \n", 5) = 5
wait4(5205, 0x7ffd22bdf6c, WNOHANG, NULL) = 0
select(6, [5], NULL, NULL, {tv_sec=0, tv_usec=100000}) = 1 (in [5], left {tv_sec=0,
tv_usec=99998})
read(5, "ERROR: Division by zero detected"..., 1023) = 33
write(1, "\320\224\320\276\321\207\320\265\321\200\320\275\320\270\320\271
\320\277\321\200\320\276\321\206\320\265\321\201\321\201:"..., 66Дочерний процесс: ERROR:
Division by zero detected
) = 66
write(1, "\320\227\320\260\320\262\320\265\321\200\321\210\320\265\320\275\320\270\320\265
\321\200\320\260\320\261\320\276\321\202\321"..., 73Завершение работы из-за деления на ноль
) = 73
close(4) = 0
close(5) = 0
wait4(5205, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 5205
--- SIGCHLD {si_signo=SIGHLD, si_code=CLD_EXITED, si_pid=5205, si_uid=0, si_status=0,
si_utime=0, si_stime=0} ---
write(1,
"\320\240\320\276\320\264\320\270\321\202\320\265\320\273\321\214\321\201\320\272\320\270\320\2
71 \320\277\321\200\320\276\321"..., 57Родительский процесс завершен
) = 57
exit_group(0) = ?
+++ exited with 0 +++
```

## Вывод

Неудобно обрабатывать ошибки. Поскольку программа написана на си, неудобно обрабатывать данные, лучше использовать с++. Сам по себе синтаксис создания процесса и пайпа несложный.