

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:

```
root@bb4442151ed3:/workspaces/Operating_Systems/lab1/build# strace ./parent
execve("./parent", ["./parent"], 0x7ffe2097760 /* 27 vars */) = 0
brk(NULL) = 0xc46000
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=33103, ...}) = 0
mmap(NULL, 33103, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fdbf5ff4000
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\0\0\0\0\0\0\0\0\0\0@>\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1901536, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdbf5ff2000
mmap(NULL, 1914496, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fdbf5e1e000
mmap(0x7fdbf5e40000, 1413120, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = 0x7fdbf5e40000
mmap(0x7fdbf5f99000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x17b000) = 0x7fdbf5f99000
mmap(0x7fdbf5fe8000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c9000) = 0x7fdbf5fe8000
mmap(0x7fdbf5fee000, 13952, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fdbf5fee000
close(3) = 0
arch_prctl(ARCH_SET_FS, 0x7fdbf5ff3540) = 0
mprotect(0x7fdbf5fe8000, 16384, PROT_READ) = 0
mprotect(0x403000, 4096, PROT_READ) = 0
mprotect(0x7fdbf6027000, 4096, PROT_READ) = 0
munmap(0x7fdbf5ff4000, 33103) = 0
pipe([3, 4]) = 0
pipe([5, 6]) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
brk(NULL) = 0xc46000
brk(0xc67000) = 0xc67000
```

```
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 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=0x7fdbf5ff3810) = 4182
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 4 2
"8 4 2\n", 1024) = 6
write(4, "8 4 2", 5) = 5
write(4, "\n", 1) = 1
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) = 57
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:"..., 90Дочерний процесс: Дочерний
процесс готов. Файл: ww
) = 90
wait4(4182, 0x7ffcba3eb3fc, WNOHANG, NULL) = 0
write(1, ">", 2>) = 2
read(0, 2 0
"2 0\n", 1024) = 4
write(4, "2 0", 3) = 3
write(4, "\n", 1) = 1
read(5, "OK: Command processed successful"..., 1023) = 35
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"..., 68Дочерний процесс: OK:
Command processed successfully
) = 68
wait4(4182, 0x7ffcba3eb3fc, WNOHANG, NULL) = 0
```

```
write(1, "> ", 2) = 2
read(0, 0xc466b0, 1024) = ? ERESTARTSYS (To be restarted if SA_RESTART is set)
--- SIGCHLD {si_signo=SIGHLD, si_code=CLD_EXITED, si_pid=4182, si_uid=0, si_status=0,
si_utime=0, si_stime=0} ---
read(0, 2
"2\n", 1024) = 2
write(4, "2", 1) = -1 EPIPE (Broken pipe)
--- SIGPIPE {si_signo=SIGPIPE, si_code=SI_USER, si_pid=4148, si_uid=0} ---
+++ killed by SIGPIPE +++
root@bb4442151ed3:/workspaces/Operating_Systems/lab1/build#
```

## Вывод

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