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-214БВ-24

Студент: Горбачев Ф.М..

Преподаватель: Бахарев В.Д.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 03.10.25

Москва, 2025

**Постановка задачи**

**Вариант 4.**

Родительский процесс создает дочерний процесс. Первой строчкой пользователь в консоль родительского процесса пишет имя файла, которое будет передано при создании дочернего процесса. Родительский и дочерний процесс должны быть представлены разными программами. Родительский процесс передает команды пользователя через pipe1, который связан со стандартным входным потоком дочернего процесса. Дочерний процесс при необходимости передает данные в родительский процесс через pipe2. Результаты своей работы дочерний процесс пишет в созданный им файл. Допускается просто открыть файл и писать туда, не перенаправляя стандартный поток вывода.

Пользователь вводит команды вида: «число число число<endline>». Далее эти числа передаются от родительского процесса в дочерний. Дочерний процесс производит деление первого числа, на последующие, а результат выводит в файл. Если происходит деление на 0, то тогда дочерний и родительский процесс завершают свою работу. Проверка деления на 0 должна осуществляться на стороне дочернего процесса. Числа имеют тип float. Количество чисел может быть произвольным.

**Общий метод и алгоритм решения**

**Использованные системные вызовы:**

* **pid\_t fork(void)** – создает дочерний процесс.
* **int pipe(int \*fd)** – создает неименованный канал для межпроцессорного взаимодействия.
* **ssize\_t write(int fd, void \*buf, size\_t count**)– записывает данные из буфера в файловый дескриптор.
* **ssize\_t read(int fd, void \*buf, size\_t count)** – читает данные из файлового дескриптора в буфер.
* **int open(const char \*pathname, int flags, mode\_t mode)** – открывает\создает файл.
* **int close(int fd)** – закрывает файл.
* **int dup2(int oldfd, int newfd)** – переназначение файлового дескриптора
* **int execl(const char \*path, const char \*arg0, ..., NULL)** – запуск другой программы, замещая текущий процесс.
* **pid\_t wait(int \*status)** – ожидание завершения дочернего процесса.
* **void exit (int status)** – завершения выполнения процесса и возвращение статуса.

**Алгоритм решения:**

Создается родительский процесс, запрашивающий имя файла и открывающий его на чтение. Содержимое файла отправляет дочернему процессу, который выполняет следующие задачи: обработка строк, парсинг чисел, операция деления чисел. Если встречается деление на ноль, дочерний процесс выводит сообщение об ошибке, уведомляя родителя через пайп. В остальных случаях результат деления отправляется родителю через второй пайп, который выводит его на экран. Таким образом, родитель и дочерний процесс обмениваются данными и выводят результаты построчно, обеспечивая корректное завершение при ошибках.

**Код программы**

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание **parent.c**

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описаниеchild.c**

**Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Протокол работы программы**

**Тесты:** vboxuser@Ubuntu1:~/MAI\_OS/MAI\_OS/lab\_1/src$ ./parent

Enter output filename: input.txt

Enter numbers separated by spaces (or 'exit' to quit):

> 10 5 2

Calculation completed successfully

> 450 9 10 5

Calculation completed successfully

> 347 13

Calculation completed successfully

> 1024 2 2 2 4

Calculation completed successfully

> 65 13 5

Calculation completed successfully

> exit

Parent process terminated**.**

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

vboxuser@Ubuntu1:~/MAI\_OS/MAI\_OS/lab\_1/src$ ./parent

Enter output filename: input\_error.txt

Enter numbers separated by spaces (or 'exit' to quit):

> 12a 3 4

Error: invalid input format

> aaa aa b c

Error: not enough numbers (need at least 2)

> 10 a

Error: not enough numbers (need at least 2)

>

> 45 0

Error: division by zero detected. Terminating...

Error: division by zero

Parent process terminated.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Strace:**

vboxuser@Ubuntu1:~/MAI\_OS/MAI\_OS/lab\_1/src$ strace ./parent

execve("./parent", ["./parent"], 0x7ffcc2074da0 /\* 65 vars \*/) = 0

brk(NULL) = 0x62a1344f7000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x771a3927c000

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=56079, ...}) = 0

mmap(NULL, 56079, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x771a3926e000

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) = 0x771a39000000

mmap(0x771a39028000, 1605632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x771a39028000

mmap(0x771a391b0000, 323584, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1b0000) = 0x771a391b0000

mmap(0x771a391ff000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1fe000) = 0x771a391ff000

mmap(0x771a39205000, 52624, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x771a39205000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x771a3926b000

arch\_prctl(ARCH\_SET\_FS, 0x771a3926b740) = 0

set\_tid\_address(0x771a3926ba10) = 7089

set\_robust\_list(0x771a3926ba20, 24) = 0

rseq(0x771a3926c060, 0x20, 0, 0x53053053) = 0

mprotect(0x771a391ff000, 16384, PROT\_READ) = 0

mprotect(0x62a0f8e69000, 4096, PROT\_READ) = 0

mprotect(0x771a392ba000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x771a3926e000, 56079) = 0

write(1, "Enter output filename: ", 23Enter output filename: ) = 23

read(0, input\_strace.txt

"input\_strace.txt\n", 1023) = 17

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

pipe2([5, 6], 0) = 0

clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x771a3926ba10) = 7131

close(3) = 0

close(6) = 0

write(1, "Enter numbers separated by space"..., 55Enter numbers separated by spaces (or 'exit' to quit):

) = 55

write(1, "> ", 2> ) = 2

read(0, 1000 4 5 6

"1000 4 5 6\n", 4095) = 11

write(4, "1000 4 5 6\n", 11) = 11

pselect6(6, [5], NULL, NULL, {tv\_sec=0, tv\_nsec=100000000}, NULL) = 1 (in [5], left {tv\_sec=0, tv\_nsec=99998489})

read(5, "Calculation completed successful"..., 255) = 35

write(2, "Calculation completed successful"..., 35Calculation completed successfully

) = 35

write(1, "> ", 2> ) = 2

read(0, 345 98 234

"345 98 234\n", 4095) = 11

write(4, "345 98 234\n", 11) = 11

pselect6(6, [5], NULL, NULL, {tv\_sec=0, tv\_nsec=100000000}, NULL) = 1 (in [5], left {tv\_sec=0, tv\_nsec=99998667})

read(5, "Calculation completed successful"..., 255) = 35

write(2, "Calculation completed successful"..., 35Calculation completed successfully

) = 35

write(1, "> ", 2> ) = 2

read(0, 50a a d

"50a a d\n", 4095) = 8

write(4, "50a a d\n", 8) = 8

pselect6(6, [5], NULL, NULL, {tv\_sec=0, tv\_nsec=100000000}, NULL) = 1 (in [5], left {tv\_sec=0, tv\_nsec=99998749})

read(5, "Error: invalid input format\n", 255) = 28

write(2, "Error: invalid input format\n", 28Error: invalid input format

) = 28

write(1, "> ", 2> ) = 2

read(0, 56 7 8 9 5 4

"56 7 8 9 5 4\n", 4095) = 13

write(4, "56 7 8 9 5 4\n", 13) = 13

pselect6(6, [5], NULL, NULL, {tv\_sec=0, tv\_nsec=100000000}, NULL) = 1 (in [5], left {tv\_sec=0, tv\_nsec=99689242})

read(5, "Calculation completed successful"..., 255) = 35

write(2, "Calculation completed successful"..., 35Calculation completed successfully

) = 35

write(1, "> ", 2> ) = 2

read(0, exit

"exit\n", 4095) = 5

close(4) = 0

close(5) = 0

wait4(-1, [{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 7131

--- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=7131, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

write(1, "Parent process terminated.\n", 27Parent process terminated.

) = 27

exit\_group(0) = ?

+++ exited with 0 +++

**Вывод**

В ходе лабораторной работы были изучены и применены основные системные вызовы для работы с процессами и межпроцессного взаимодействия в Linux. Реализована программа, демонстрирующая создание процессов, организацию каналов связи и перенаправление стандартных потоков. Для обработки входных данных были разработаны собственные алгоритмы разбора строк и преобразования чисел, учитывающие различные пробелы и разделители. Также обеспечено корректное закрытие файловых дескрипторов для предотвращения утечек.