Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №2 по курсу**

**«Операционные системы»**

**Тема работы**

**«Работа с процессами и взаимодействие между ними»**

Студент: Лютоев Илья Александрович

Группа: М8О-207Б-21

Вариант: 6

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

## Задание

Составить и отладить программу на языке Си, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. В результате работы программа (основной процесс) должен создать для решение задачи один или несколько дочерних процессов. Взаимодействие между процессами осуществляется через системные сигналы/события и/или каналы (pipe). Необходимо обрабатывать системные ошибки, которые могут возникнуть в результате работы.

### Группа вариантов №2

Родительский процесс создает дочерний процесс. Первой строчкой пользователь в консоль родительского процесса вводит имя файла, оторое будет использовано для открытия файла с таким именем на чтение. Стандартный поток ввода дочернего процесса переопределяется открытым файлом. Дочерний процесс читает команды из стандартного потока ввода. Стандартный поток вывода дочернего процесса перенаправляется в pipe1. Родительский процесс читает из pipe1 и прочитанное выводит в свой стандартный поток вывода. Родительский и дочерний процесс должны быть представлены разными программами.

### Вариант 6

В файле записаны команды вида: «число число число». Дочерний процесс считает их сумму и выводит результат в стандартный поток вывода. Числа имеют тип int. Количество чисел может быть произвольным. img

### Методы и алгоритмы решения

Откроем файл введённый пользователем и создаём pipe. Используем fork() и execlp() для создания дочернего процесса child, в него мы передаём наш pipe и файл. После этого с помощью dup2() перенаправляем файл в поток ввода, а поток вывода в переданный pipe, после чего считываем данные из файла и обрабатываем их. После чего в parent мы перенаправляем pipe в поток вывода.

## Выполнение программы

lyutoev@fedora ~/workshop/os/lab2/build: ./main   
../tests/1  
6  
12  
21  
0  
3  
73  
10  
26  
46  
lyutoev@fedora ~/workshop/os/lab2/build: cat ../tests/1  
1 2 3  
3 4 5  
6 7 8  
0 0 0  
1 1 1  
33 5 35  
10 0 0  
1 22 3  
12 12 22

## Демонстрация системных вызовов с помощью strace

execve("./main", ["./main"], 0x7ffc27c5c740 /\* 55 vars \*/) = 0  
brk(NULL) = 0x19ea000  
arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffe91e84720) = -1 EINVAL (Недопустимый аргумент)  
mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe19555b000  
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=94507, ...}, AT\_EMPTY\_PATH) = 0  
mmap(NULL, 94507, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fe195543000  
close(3) = 0  
openat(AT\_FDCWD, "/lib64/libstdc++.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\0\0\0\0\0\0\0\0"..., 832) = 832  
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\315\212\206\262\230\233\36O\224\0H\34\36\241\250\v"..., 36, 800) = 36  
newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2360040, ...}, AT\_EMPTY\_PATH) = 0  
mmap(NULL, 2308096, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe19530f000  
mprotect(0x7fe1953a9000, 1609728, PROT\_NONE) = 0  
mmap(0x7fe1953a9000, 1118208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7fe1953a9000  
mmap(0x7fe1954ba000, 487424, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ab000) = 0x7fe1954ba000  
mmap(0x7fe195532000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x222000) = 0x7fe195532000  
mmap(0x7fe195540000, 10240, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe195540000  
close(3) = 0  
openat(AT\_FDCWD, "/lib64/libm.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\0\0\0\0\0\0\0\0"..., 832) = 832  
newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=919824, ...}, AT\_EMPTY\_PATH) = 0  
mmap(NULL, 905480, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe195231000  
mmap(0x7fe19523f000, 466944, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7fe19523f000  
mmap(0x7fe1952b1000, 376832, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x80000) = 0x7fe1952b1000  
mmap(0x7fe19530d000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xdb000) = 0x7fe19530d000  
close(3) = 0  
openat(AT\_FDCWD, "/lib64/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3  
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832  
newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=128512, ...}, AT\_EMPTY\_PATH) = 0  
mmap(NULL, 127272, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe195211000  
mmap(0x7fe195214000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7fe195214000  
mmap(0x7fe19522b000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7fe19522b000  
mmap(0x7fe19522f000, 4096, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7fe19522f000  
mmap(0x7fe195230000, 296, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe195230000  
close(3) = 0  
openat(AT\_FDCWD, "/lib64/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\20\227\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  
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\0"..., 80, 848) = 80  
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0o\\\345\24\251\347\365\36\2G\245'\303\244\36\331"..., 68, 928) = 68  
newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2375216, ...}, 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, 2104720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe19500f000  
mmap(0x7fe195037000, 1523712, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7fe195037000  
mmap(0x7fe1951ab000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19c000) = 0x7fe1951ab000  
mmap(0x7fe195203000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1f3000) = 0x7fe195203000  
mmap(0x7fe195209000, 32144, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe195209000  
close(3) = 0  
mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe19500d000  
arch\_prctl(ARCH\_SET\_FS, 0x7fe19500e200) = 0  
set\_tid\_address(0x7fe19500e4d0) = 70006  
set\_robust\_list(0x7fe19500e4e0, 24) = 0  
rseq(0x7fe19500eba0, 0x20, 0, 0x53053053) = 0  
mprotect(0x7fe195203000, 16384, PROT\_READ) = 0  
mprotect(0x7fe19522f000, 4096, PROT\_READ) = 0  
mprotect(0x7fe19530d000, 4096, PROT\_READ) = 0  
mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe19500b000  
mprotect(0x7fe195532000, 53248, PROT\_READ) = 0  
mprotect(0x404000, 4096, PROT\_READ) = 0  
mprotect(0x7fe195592000, 8192, PROT\_READ) = 0  
prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0  
munmap(0x7fe195543000, 94507) = 0  
getrandom("\x94\xda\xfb\xb4\xec\xdb\x34\x09", 8, GRND\_NONBLOCK) = 8  
brk(NULL) = 0x19ea000  
brk(0x1a0b000) = 0x1a0b000  
futex(0x7fe1955406bc, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0  
newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x3), ...}, AT\_EMPTY\_PATH) = 0  
read(0, ../tests/1  
"../tests/1\n", 1024) = 11  
openat(AT\_FDCWD, "../tests/1", O\_RDONLY) = 3  
pipe2([4, 5], 0) = 0  
clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7fe19500e4d0) = 70026  
dup2(1, 4) = 4  
close(5) = 0  
close(4) = 0  
close(3) = 0  
exit\_group(0) = ?  
+++ exited with 0 +++  
6   
12  
21  
0  
3  
73  
10  
26  
46

## Вывод

В ходе выполнения работы я узнал о работе с системными вызовами, процессами, PIPEs и dup.