

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, которое 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 введенный 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
3 3 5 3 5
1 0 0 0
1 2 2 3
1 2 1 2 2 2

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\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\360\224\0H\34\36\241\250w"
..., 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\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\0\0\0\0>\0\1\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\0\0\0\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.