

Лабораторная работа №2 по курсу "Операционные системы"

Студент группы: М80-207Б-21, Крючков Артемий Владимирович

Контакты: artemkr2003@mail.ru

Работа выполнена: 17.09.2022

Преподаватель: Миронов Евгений Сергеевич

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 №1

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 с 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.

Вариант 4

4 вариант) 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

Методы и алгоритмы решения

```
#include <bits/stdc++.h>
#include <unistd.h>

#include "../includes/helper.h"

#define OUTPUT 0
#define INPUT 1

int main()
{
    int pipefd[2];
    if (pipe(pipefd) == -1) {
        std::cerr << "ERR pipe\n";
        return 1;
    }

    printf("Name of file: ");
    std::string file_name;
    if (!getline(std::cin, file_name)) {
        std::cerr << "ERR create file\n";
        return 1;
    }

    switch (auto pid = fork()) {
    case -1:
        std::cerr << "ERR fork\n";
        break;
    case 0:
        execlp("./child", std::to_string(pipefd[OUTPUT]).c_str(),
std::to_string(pipefd[INPUT]).c_str(), file_name.c_str(),
std::to_string(pid).c_str());
        break;
    default:
        float x;
        auto flag = true;
        char c;

        while (c != EOF) {
            std::cin >> x;

            if (flag) {
                packet_value<float> packet(x, value_info::START_OF_SEQUENCE);
                write(pipefd[INPUT], &packet, sizeof(packet));
                flag = false;
            } else {
                c = getchar();
                if (c == '\n') {
                    packet_value<float> packet(x, value_info::END_OF_SEQUENCE);
                    write(pipefd[INPUT], &packet, sizeof(packet));
                }
            }
        }
    }
}
```

```

        flag = true;
    } else {
        packet_value<float> packet(x,
value_info::MIDDLE_OF_SEQUENCE);
        write(pipefd[INPUT], &packet, sizeof(packet));
    }
}

break;
}

close(pipefd[OUTPUT]);
close(pipefd[INPUT]);
return 0;
}

```

```

#include <bits/stdc++.h>
#include <fcntl.h>
#include <unistd.h>
#include <signal.h>

#include "../includes/helper.h"

#define OUTPUT 0
#define INPUT 1

int main(int argc, char const* argv[])
{
    int pipefd[2];
    pipefd[OUTPUT] = std::stoi(argv[0]);
    pipefd[INPUT] = std::stoi(argv[1]);

    int file = open(argv[2], O_RDWR | O_APPEND | O_CREAT, S_IRUSR | S_IRGRP |
S_IROTH);
    if (file == -1) {
        std::cerr << "ERR open file\n";
        return 1;
    }

    packet_value<float> first_packet{};

    while (true) {
        packet_value<float> packet(pipefd[OUTPUT]);
        // read(pipefd[OUTPUT], &packet, sizeof(packet));

        // std::cout << packet.info << std::endl;
        // std::cout << packet.value << std::endl;

        switch(packet.info) {
            case START_OF_SEQUENCE:
                first_packet = packet;

```

```

        // std::cout << "start" << std::endl;
        break;
    case MIDDLE_OF_SEQUENCE:
        if (packet.value == 0) {
            goto kill_parent_process;
        }

        first_packet.value /= packet.value;
        // std::cout << "mid" << std::endl;
        break;
    case END_OF_SEQUENCE:
        if (packet.value == 0) {
            goto kill_parent_process;
        }

        first_packet.value /= packet.value;
        std::cout << "ANS = " << first_packet.value << std::endl;

        std::string buf = std::to_string(first_packet.value) + "\n";

        write(file, buf.c_str(), strlen(buf.c_str()));

        // std::cout << "end" << std::endl;
        break;
    }

}

goto end;

kill_parent_process:
    kill(std::stoi(argv[3]), SIGKILL);
end:
    close(pipefd[OUTPUT]);
    close(pipefd[INPUT]);
    close(file);
    return 0;
}

```

Выполнение программы

```

Name of file: file1
1 2 3 4
ANS = 0.0416667
222 2 312 123 13
ANS = 0.000222495

```

Strace

```

[crewch@pc build]$ strace ./main
execve("./main", [ "./main" ], 0x7ffe282ecac0 /* 84 vars */) = 0
brk(NULL)                                = 0x55e3b7f2d000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd8038b8f0) = -1 EINVAL (Недопустимый
аргумент)
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=158379, ...}, AT_EMPTY_PATH) =
0
mmap(NULL, 158379, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f453e336000
close(3)                                  = 0
openat(AT_FDCWD, "/usr/lib/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"...,
832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=19198496, ...}, AT_EMPTY_PATH)
= 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f453e334000
mmap(NULL, 2320384, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f453e0fd000
mmap(0x7f453e196000, 1138688, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x99000) = 0x7f453e196000
mmap(0x7f453e2ac000, 487424, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1af000) = 0x7f453e2ac000
mmap(0x7f453e323000, 57344, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x225000) = 0x7f453e323000
mmap(0x7f453e331000, 10240, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f453e331000
close(3)                                  = 0
openat(AT_FDCWD, "/usr/lib/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"...,
832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=944600, ...}, AT_EMPTY_PATH) =
0
mmap(NULL, 946368, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f453e015000
mmap(0x7f453e023000, 499712, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7f453e023000
mmap(0x7f453e09d000, 385024, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x88000) = 0x7f453e09d000
mmap(0x7f453e0fb000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe5000) = 0x7f453e0fb000
close(3)                                  = 0
openat(AT_FDCWD, "/usr/lib/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"...,
832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=571848, ...}, AT_EMPTY_PATH) =
0
mmap(NULL, 127304, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f453dff5000
mmap(0x7f453dff8000, 94208, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f453dff8000

```

```

mmap(0x7f453e00f000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1a000) = 0x7f453e00f000
mmap(0x7f453e013000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x7f453e013000
close(3) = 0
openat(AT_FDCWD, "/usr/lib/libc.so.6", 0_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\0P4\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
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1953472, ...}, 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, 1994384, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f453de0e000
mmap(0x7f453de30000, 1421312, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = 0x7f453de30000
mmap(0x7f453df8b000, 356352, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x17d000) = 0x7f453df8b000
mmap(0x7f453dfe2000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d4000) = 0x7f453dfe2000
mmap(0x7f453dfe8000, 52880, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f453dfe8000
close(3) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f453de0c000
arch_prctl(ARCH_SET_FS, 0x7f453de0d200) = 0
set_tid_address(0x7f453de0d4d0) = 30005
set_robust_list(0x7f453de0d4e0, 24) = 0
rseq(0x7f453de0db20, 0x20, 0, 0x53053053) = 0
mprotect(0x7f453dfe2000, 16384, PROT_READ) = 0
mprotect(0x7f453e013000, 4096, PROT_READ) = 0
mprotect(0x7f453e0fb000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f453de0a000
mprotect(0x7f453e323000, 53248, PROT_READ) = 0
mprotect(0x55e3b6afa000, 4096, PROT_READ) = 0
mprotect(0x7f453e38e000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f453e336000, 158379) = 0
getrandom("\xbax8c\x9b\x3d\x35\x4a\x07\x54", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x55e3b7f2d000
brk(0x55e3b7f4e000) = 0x55e3b7f4e000
futex(0x7f453e3316bc, FUTEX_WAKE_PRIVATE, 2147483647) = 0
pipe2([3, 4], 0) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x3), ...},
AT_EMPTY_PATH) = 0
write(1, "Name of file: ", 14Name of file: ) = 14
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x3), ...},
AT_EMPTY_PATH) = 0

```

```

read(0, file2
"file2\n", 1024)                = 6
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f453de0d4d0) = 30043
read(0, 1 2 3 4 5 6
"1 2 3 4 5 6\n", 1024)          = 12
write(4, "\0\0\200?\0\0\0\0", 8)    = 8
write(4, "\0\0\0@\1\0\0\0", 8)      = 8
write(4, "\0\0@@\1\0\0\0", 8)       = 8
write(4, "\0\0\200@\1\0\0\0", 8)    = 8
write(4, "\0\0\240@\1\0\0\0", 8)    = 8
write(4, "\0\0\300@\2\0\0\0", 8)    = 8
read(0, ANS = 0.00138889
"", 1024)                        = 0
write(4, "\0\0\300@\0\0\0\0", 8)    = 8
write(4, "\0\0\300@\1\0\0\0", 8)    = 8
close(3)                         = 0
close(4)                         = 0
exit_group(0)                    = ?
+++ exited with 0 +++

```