

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-209БВ-24

Студент: Корепанов Иван Алексеевич

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 08.10.25

Москва, 2025

Постановка задачи

Вариант 10.

В файле записаны команды вида: «число». Дочерний процесс производит проверку этого числа на простоту. Если число составное, то дочерний процесс пишет это число в стандартный поток вывода. Если число отрицательное или простое, то тогда дочерний и родительский процессы завершаются. Количество чисел может быть произвольным

Общий метод и алгоритм решения

Использованные системные вызовы:

- pid_t fork(void); – создает дочерний процесс.
- int pipe(int *fd); – создание неименованного канала для передачи данных между процессами
- int dup2(int oldfd, int newfd); — переназначение файлового дескриптора
- int execl(const char *path, const char *arg, ..., (char *)0); — замещает текущий процесс новой программой.
- int close(int fd); — закрывает файловый дескриптор, освобождая системные ресурсы.

Сначала были созданы необходимые переменные, включая файловые дескрипторы, которые использовались для организации обмена данными между процессами. После этого родительский процесс запрашивал у пользователя имя файла и открывал его для чтения. Затем создавался неименованный канал (pipe), обеспечивающий связь между родительским и дочерним процессами. После вызова fork() создавался дочерний процесс, в котором стандартный поток ввода перенаправлялся на открытый файл, а стандартный поток вывода — в конец записи канала. Далее с помощью системного вызова execl() запускалась отдельная программа дочернего процесса. Дочерний процесс считывал числа из файла, проверял их на простоту и передавал составные числа в канал. Родительский процесс, в свою очередь, читал данные из канала и выводил их на экран. После завершения работы дочернего процесса родитель закрывал все ненужные файловые дескрипторы и завершал выполнение программы.

Код программы

parent.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

#define MAXSIZE 256

int main() {
    int pipefd[2];
    pid_t pid;
    char filename[MAXSIZE];

    if (pipe(pipefd) == -1) {
        perror("pipe");
        exit(EXIT_FAILURE);
    }

    printf("файл:");
    if (scanf("%255s", filename) != 1) {
        fprintf(stderr, "неверное имя файла\n");
        exit(EXIT_FAILURE);
    }

    pid = fork();
    if (pid == -1) {
        perror("fork");
        exit(EXIT_FAILURE);
    }

    if (pid == 0) {
```

```
close(pipefd[0]);

dup2(pipefd[1], STDOUT_FILENO);

close(pipefd[1]);


execl("./child", "child", filename, (char *)NULL);

perror("execl");

exit(EXIT_FAILURE);

} else {

close(pipefd[1]);


char *buffer = (char *)malloc(256);

int buffer_size = 256;

int total_read = 0;

ssize_t count;

while ((count = read(pipefd[0], buffer + total_read, buffer_size - total_read - 1)) > 0) {

total_read += count;

if (total_read >= buffer_size - 1) {

buffer_size *= 2;

char *temp = (char *)realloc(buffer, buffer_size);

if (temp == NULL) {

free(buffer);

perror("realloc");

exit(EXIT_FAILURE);

}

buffer = temp;

}

}

if (count == -1) {

perror("read");

free(buffer);

}
```

```
    exit(EXIT_FAILURE);
```

```
}
```

```
buffer[total_read] = '\0';
```

```
printf("%s", buffer);
```

```
fflush(stdout);
```

```
if (atoi(buffer) <= 0) {
```

```
}
```

```
free(buffer);
```

```
close(pipefd[0]);
```

```
wait(NULL);
```

```
}
```

```
return 0;
```

```
}
```

Протокол работы программы

Здесь нужно показать тесты программы (текст или скриншоты), а затем показать полный вывод утилиты strace (или какой-либо другой утилиты на Windows, если вы выполняете лабы на этой операционной системе).

В strace нужно обязательно выделить, где происходят системные вызовы, которые вы использовали в лабораторной работе (например, где в первой лабораторной работе был вызван fork и другие вызовы). Полный список вызовов, которые нужно будет выделить в выводе strace, будет указан при выдаче лабы в нашем канале.

Тестирование:

```
$ ./main
file1.txt
file2.txt
string1
string2
string3
string4
string5
$ cat < file1.txt
string1
string2
```

```
string3
$ cat < file2.txt
string4
string5
```

Strace:

```
close(3) = 0
arch_prctl(ARCH_SET_FS, 0x7fb731767540) = 0
mprotect(0x7fb73175c000, 12288, PROT_READ) = 0
mprotect(0x5643edb3c000, 4096, PROT_READ) = 0
mprotect(0x7fb7317a8000, 4096, PROT_READ) = 0
munmap(0x7fb731768000, 73833) = 0
brk(NULL) = 0x5643edd4d000
brk(0x5643edd6e000) = 0x5643edd6e000
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
read(0, file1.txt
"file1.txt\n", 1024) = 10
openat(AT_FDCWD, "file1.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
read(0, file2.txt
"file2.txt\n", 1024) = 10
openat(AT_FDCWD, "file2.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 4
pipe([5, 6]) = 0
pipe([7, 8]) = 0
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fb731767810) = 4728
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fb731767810) = 4729
close(3) = 0
close(4) = 0
close(5) = 0
close(7) = 0
read(0, strace: Process 4728 attached
<unfinished ...>
[pid 4728] close(4) = 0
[pid 4728] close(7) = 0
[pid 4728] close(8) = 0
[pid 4728] close(6) = 0
[pid 4728] dup2(5, 0) = 0
[pid 4728] dup2(3, 1) = 1
[pid 4728] close(3)strace: Process 4729 attached
) = 0
[pid 4728] execve("child", NULL, 0x7ffed25bef48 /* 49 vars */ <unfinished ...>
[pid 4729] close(3) = 0
[pid 4729] close(5) = 0
[pid 4729] close(6) = 0
[pid 4729] close(8) = 0
[pid 4729] dup2(7, 0) = 0
[pid 4729] dup2(4, 1) = 1
[pid 4729] close(4) = 0
[pid 4729] execve("child", NULL, 0x7ffed25bef48 /* 49 vars */ <unfinished ...>
[pid 4728] <... execve resumed> = 0
[pid 4728] brk(NULL <unfinished ...>
[pid 4729] <... execve resumed> = 0
```

```
[pid 4729] brk(NULL <unfinished ...>
[pid 4728] <... brk resumed> = 0x55b32f123000
[pid 4728] arch_prctl(0x3001 /* ARCH_???, 0x7ffc0937eb80 <unfinished ...>
[pid 4729] <... brk resumed> = 0x55884ee36000
[pid 4729] arch_prctl(0x3001 /* ARCH_???, 0x7ffd061df910) = -1 EINVAL
(Недопустимый аргумент)
[pid 4728] <... arch_prctl resumed> = -1 EINVAL (Недопустимый аргумент)
[pid 4729] access("/etc/ld.so.preload", R_OK <unfinished ...>
[pid 4728] access("/etc/ld.so.preload", R_OK) = -1 ENOENT (Нет такого файла или
каталога)
[pid 4729] <... access resumed> = -1 ENOENT (Нет такого файла или каталога)
[pid 4728] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>
[pid 4729] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
[pid 4728] <... openat resumed> = 3
[pid 4728] fstat(3, <unfinished ...>
[pid 4729] fstat(3, <unfinished ...>
[pid 4728] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=73833, ...}) = 0
[pid 4729] <... fstat resumed>{st_mode=S_IFREG|0644, st_size=73833, ...}) = 0
[pid 4729] mmap(NULL, 73833, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>
[pid 4728] mmap(NULL, 73833, PROT_READ, MAP_PRIVATE, 3, 0 <unfinished ...>
[pid 4729] <... mmap resumed> = 0x7f9f03ba1000
[pid 4728] <... mmap resumed> = 0x7f8c0c66c000
[pid 4728] close(3 <unfinished ...>
[pid 4729] close(3 <unfinished ...>
[pid 4728] <... close resumed> = 0
[pid 4729] <... close resumed> = 0
[pid 4728] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC
<unfinished ...>
[pid 4729] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) =
3
[pid 4728] <... openat resumed> = 3
[pid 4729] read(3, <unfinished ...>
[pid 4728] read(3, <unfinished ...>
[pid 4729] <... read
resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832
[pid 4728] <... read
resumed>"\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\360q\2\0\0\0\0\0"..., 832) = 832
[pid 4729] pread64(3, <unfinished ...>
[pid 4728] pread64(3, <unfinished ...>
[pid 4729] <... pread64
resumed>"\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\0"..., 784, 64) = 784
[pid 4728] <... pread64
resumed>"\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"..., 784, 64) = 784
[pid 4729] pread64(3, <unfinished ...>
[pid 4728] pread64(3, <unfinished ...>
[pid 4729] <... pread64
resumed>"\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
```

```
[pid 4728] <... pread64
resumed>"\4\0\0\0\2\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0", 32, 848) = 32

[pid 4729] pread64(3, <unfinished ...>
[pid 4728] pread64(3, <unfinished ...>
[pid 4729] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263".
.., 68, 880) = 68

[pid 4728] <... pread64
resumed>"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263".
.., 68, 880) = 68

[pid 4729] fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
[pid 4729] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f9f03b9f000

[pid 4729] 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\0"..., 784, 64) = 784

[pid 4729] pread64(3,
"\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0", 32, 848) = 32

[pid 4729] pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68,
880) = 68

[pid 4729] mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f9f039ad000

[pid 4729] mprotect(0x7f9f039d2000, 1847296, PROT_NONE) = 0
[pid 4729] mmap(0x7f9f039d2000, 1540096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f9f039d2000

[pid 4729] mmap(0x7f9f03b4a000, 303104, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f9f03b4a000

[pid 4729] mmap(0x7f9f03b95000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000 <unfinished ...>

[pid 4728] fstat(3, <unfinished ...>
[pid 4729] <... mmap resumed> = 0x7f9f03b95000
[pid 4728] <... fstat resumed>{st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
[pid 4728] mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
[pid 4729] mmap(0x7f9f03b9b000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0 <unfinished ...>
[pid 4728] <... mmap resumed> = 0x7f8c0c66a000
[pid 4728] pread64(3, <unfinished ...>
[pid 4729] <... mmap resumed> = 0x7f9f03b9b000
[pid 4729] close(3 <unfinished ...>
[pid 4728] <... pread64
resumed>"\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"..., 784, 64) = 784
[pid 4728] pread64(3, <unfinished ...>
```

```
[pid  4729] <... close resumed>          = 0
[pid  4728] <... pread64
resumed>"\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0", 32, 848) = 32
[pid  4729] arch_prctl(ARCH_SET_FS, 0x7f9f03ba0540 <unfinished ...>
[pid  4728] pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"..., 68,
880) = 68
[pid  4728] mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f8c0c478000
[pid  4728] mprotect(0x7f8c0c49d000, 1847296, PROT_NONE) = 0
[pid  4729] <... arch_prctl resumed>      = 0
[pid  4729] mprotect(0x7f9f03b95000, 12288, PROT_READ <unfinished ...>
[pid  4728] mmap(0x7f8c0c49d000, 1540096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f8c0c49d000
[pid  4728] mmap(0x7f8c0c615000, 303104, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f8c0c615000
[pid  4728] mmap(0x7f8c0c660000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f8c0c660000
[pid  4728] mmap(0x7f8c0c666000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f8c0c666000
[pid  4728] close(3)                      = 0
[pid  4728] arch_prctl(ARCH_SET_FS, 0x7f8c0c66b540 <unfinished ...>
[pid  4729] <... mprotect resumed>       = 0
[pid  4728] <... arch_prctl resumed>     = 0
[pid  4728] mprotect(0x7f8c0c660000, 12288, PROT_READ <unfinished ...>
[pid  4729] mprotect(0x55884dfa5000, 4096, PROT_READ) = 0
[pid  4729] mprotect(0x7f9f03be1000, 4096, PROT_READ) = 0
[pid  4729] munmap(0x7f9f03ba1000, 73833) = 0
[pid  4728] <... mprotect resumed>       = 0
[pid  4728] mprotect(0x55b32d937000, 4096, PROT_READ <unfinished ...>
[pid  4729] read(0, <unfinished ...>
[pid  4728] <... mprotect resumed>       = 0
[pid  4728] mprotect(0x7f8c0c6ac000, 4096, PROT_READ) = 0
[pid  4728] munmap(0x7f8c0c66c000, 73833) = 0
[pid  4728] read(0, 123456789123456789123456789
<unfinished ...>
[pid  4726] <... read resumed>"1", 1)   = 1
[pid  4726] read(0, "2", 1)                 = 1
[pid  4726] read(0, "3", 1)                 = 1
[pid  4726] read(0, "4", 1)                 = 1
[pid  4726] read(0, "5", 1)                 = 1
[pid  4726] read(0, "6", 1)                 = 1
[pid  4726] read(0, "7", 1)                 = 1
[pid  4726] read(0, "8", 1)                 = 1
[pid  4726] read(0, "9", 1)                 = 1
[pid  4726] read(0, "1", 1)                 = 1
```

```
[pid 4726] read(0, "2", 1)          = 1
[pid 4726] read(0, "3", 1)          = 1
[pid 4726] read(0, "4", 1)          = 1
[pid 4726] read(0, "5", 1)          = 1
[pid 4726] read(0, "6", 1)          = 1
[pid 4726] read(0, "7", 1)          = 1
[pid 4726] read(0, "8", 1)          = 1
[pid 4726] read(0, "9", 1)          = 1
[pid 4726] read(0, "1", 1)          = 1
[pid 4726] read(0, "2", 1)          = 1
[pid 4726] read(0, "3", 1)          = 1
[pid 4726] read(0, "4", 1)          = 1
[pid 4726] read(0, "5", 1)          = 1
[pid 4726] read(0, "6", 1)          = 1
[pid 4726] read(0, "7", 1)          = 1
[pid 4726] read(0, "8", 1)          = 1
[pid 4726] read(0, "9", 1)          = 1
[pid 4726] read(0, "\n", 1)          = 1
[pid 4726] write(8, "\33\0\0\0", 4) = 4
[pid 4726] write(8, "123456789123456789123456789", 27) = 27
[pid 4726] read(0, <unfinished ...>
[pid 4729] <... read resumed>"\33\0\0\0", 4) = 4
[pid 4729] read(0, "123456789123456789123456789", 27) = 27
[pid 4729] fstat(1, {st_mode=S_IFREG|0664, st_size=0, ...}) = 0
[pid 4729] brk(NULL)                = 0x55884ee36000
[pid 4729] brk(0x55884ee57000)      = 0x55884ee57000
[pid 4729] write(1, "987654321987654321987654321\n", 28) = 28
[pid 4729] read(0, okay
<unfinished ...>
[pid 4726] <... read resumed>"o", 1) = 1
[pid 4726] read(0, "k", 1)          = 1
[pid 4726] read(0, "a", 1)          = 1
[pid 4726] read(0, "y", 1)          = 1
[pid 4726] read(0, "\n", 1)          = 1
[pid 4726] write(8, "\4\0\0\0", 4) = 4
[pid 4726] write(8, "okay", 4)       = 4
[pid 4726] read(0, <unfinished ...>
[pid 4729] <... read resumed>"\4\0\0\0", 4) = 4
[pid 4729] read(0, "okay", 4)       = 4
[pid 4729] write(1, "yako\n", 5)     = 5
[pid 4729] read(0, fedor
<unfinished ...>
[pid 4726] <... read resumed>"f", 1) = 1
[pid 4726] read(0, "e", 1)          = 1
[pid 4726] read(0, "d", 1)          = 1
[pid 4726] read(0, "o", 1)          = 1
[pid 4726] read(0, "r", 1)          = 1
[pid 4726] read(0, "\n", 1)          = 1
[pid 4726] write(6, "\5\0\0\0", 4)   = 4
```

```
[pid 4726] write(6, "fedor", 5)          = 5
[pid 4726] read(0, <unfinished ...>
[pid 4728] <... read resumed>"\5\0\0\0", 4) = 4
[pid 4728] read(0, "fedor", 5)          = 5
[pid 4728] fstat(1, {st_mode=S_IFREG|0664, st_size=0, ...}) = 0
[pid 4728] brk(NULL)                  = 0x55b32f123000
[pid 4728] brk(0x55b32f144000)        = 0x55b32f144000
[pid 4728] write(1, "rodef\n", 6)       = 6
[pid 4728] read(0, rodeo
<unfinished ...>
[pid 4726] <... read resumed>"r", 1)   = 1
[pid 4726] read(0, "o", 1)              = 1
[pid 4726] read(0, "d", 1)              = 1
[pid 4726] read(0, "e", 1)              = 1
[pid 4726] read(0, "o", 1)              = 1
[pid 4726] read(0, "\n", 1)             = 1
[pid 4726] write(6, "\5\0\0\0", 4)      = 4
[pid 4726] write(6, "rodeo", 5)         = 5
[pid 4726] read(0, <unfinished ...>
[pid 4728] <... read resumed>"\5\0\0\0", 4) = 4
[pid 4728] read(0, "rodeo", 5)          = 5
[pid 4728] write(1, "oedor\n", 6)        = 6
[pid 4728] read(0, hihihi
<unfinished ...>
[pid 4726] <... read resumed>"h", 1)   = 1
[pid 4726] read(0, "i", 1)              = 1
[pid 4726] read(0, "h", 1)              = 1
[pid 4726] read(0, "i", 1)              = 1
[pid 4726] read(0, "h", 1)              = 1
[pid 4726] read(0, "i", 1)              = 1
[pid 4726] read(0, "\n", 1)             = 1
[pid 4726] write(8, "\6\0\0\0", 4)      = 4
[pid 4726] write(8, "hihihi", 6)        = 6
[pid 4726] read(0, <unfinished ...>
[pid 4729] <... read resumed>"\6\0\0\0", 4) = 4
[pid 4729] read(0, "hihihi", 6)         = 6
[pid 4729] write(1, "ihihih\n", 7)       = 7
[pid 4729] read(0, <unfinished ...>
[pid 4726] <... read resumed>"", 1)    = 0
[pid 4726] close(6)                   = 0
[pid 4726] close(8)                   = 0
[pid 4726] exit_group(0)              = ?
[pid 4726] +++ exited with 0 ===+
[pid 4728] <... read resumed>"", 4)     = 0
[pid 4728] exit_group(0)              = ?
[pid 4728] +++ exited with 0 ===+
<... read resumed>"", 4)                 = 0
exit_group(0)                          = ?
+++ exited with 0 ===+
```

Вывод

Вывод 3-4 предложения. Также опишите проблемы, с которыми столкнулись при выполнении лабораторной работы (если они были), пожелания и т.д.