Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №8 по курсу**

**«Операционные системы»**

**ДИАГНОСТИКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Студент: Куценко Максим Дмитриевич

Группа: М8О–212Б-22

Вариант: 3

Преподаватель: Соколов Андрей Алексеевич

Оценка: \_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_

Москва, 2023.

**Постановка задачи**

## Цель работы

## Приобретение практических навыков диагностики работы программного обеспечения.

## Задание

*При выполнении лабораторных работ по курсу ОС необходимо продемонстрировать ключевые системные вызовы, которые в них используются и то, что их использование соответствует варианту ЛР.*

По итогам выполнения всех лабораторных работ отчет по данной ЛР должен содержать краткую сводку по исследованию написанных программ.

**Общие сведения о программе**

Для диагностики используется лабораторная работа № 3. В ней при помощи Makefile собираются программы — main и divider. В main происходит создание и маппирование файла, а также создание процесса-потомка при помощи fork, чтение из пользовательского ввода, обработка ошибок процесса-потомка и вывод результата. В divider происходит считывание из файла, с которым происходит работа, деление чисел, запись результата в файл и уменьшение размера файла. Пока проходит ввод в родительском файле, процесс-потомок останавливается при помощи сигнала.

Программа компилируется из файла main.c. Также используется заголовочные файлы: "stdlib.h", "stdio.h", "unistd.h", "fcntl.h", "sys/mman.h", "sys/wait.h", "signal.h", "string.h", "errno.h".

В программе используются следующие системные вызовы:

1. **open** — открытие файла.
2. **close —** закрытие файла.
3. **mmap —** маппирование файла.
4. **munmap —** прекращение маппирования.
5. **raise —** передача сигнала вызвавшему команду процессу.
6. **kill —** передача сигнала другому процессу.
7. **fork** — создание дочернего процесса.
8. **exec** — запуск исполняемого файла.
9. **wait** — ожидание окончания работы дочернего процесса родительским. Получает статус выхода.
10. **ftruncate** — установка размера файла.

**Общий метод и алгоритм решения**.

Диагностика программы производится в операционной системе при помощи утилиты strace. Утилита отображает все системные вызовы, сгенерированные программой.

**Результаты работы**

**[Maxim@HONOR-MB15 src]$ strace -o res.txt -f ./main**

171503 execve("./main", ["./main"], 0x7fff426e8088 /\* 88 vars \*/) = 0

171503 brk(NULL) = 0x1f43000

171503 arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffd1af0b5f0) = -1 EINVAL (Недопустимый аргумент)

171503 access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (Нет такого файла или каталога)

171503 openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

171503 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=86775, ...}, AT\_EMPTY\_PATH) = 0

171503 mmap(NULL, 86775, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f0b5be64000

171503 close(3) = 0

171503 openat(AT\_FDCWD, "/lib64/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

171503 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\320\226\2\0\0\0\0\0"..., 832) = 832

171503 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

171503 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

171503 pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\202W\356\220vF\351\260W\31u3\321\344\254\216"..., 68, 928) = 68

171503 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2375624, ...}, AT\_EMPTY\_PATH) = 0

171503 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f0b5be62000

171503 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

171503 mmap(NULL, 2104720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f0b5bc00000

171503 mmap(0x7f0b5bc28000, 1523712, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7f0b5bc28000

171503 mmap(0x7f0b5bd9c000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19c000) = 0x7f0b5bd9c000

171503 mmap(0x7f0b5bdf4000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1f3000) = 0x7f0b5bdf4000

171503 mmap(0x7f0b5bdfa000, 32144, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f0b5bdfa000

171503 close(3) = 0

171503 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f0b5be60000

171503 arch\_prctl(ARCH\_SET\_FS, 0x7f0b5be63600) = 0

171503 set\_tid\_address(0x7f0b5be638d0) = 171503

171503 set\_robust\_list(0x7f0b5be638e0, 24) = 0

171503 rseq(0x7f0b5be63fa0, 0x20, 0, 0x53053053) = 0

171503 mprotect(0x7f0b5bdf4000, 16384, PROT\_READ) = 0

171503 mprotect(0x403000, 4096, PROT\_READ) = 0

171503 mprotect(0x7f0b5beaf000, 8192, PROT\_READ) = 0

171503 prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

171503 munmap(0x7f0b5be64000, 86775) = 0

171503 openat(AT\_FDCWD, "file.txt", O\_RDWR|O\_CREAT|O\_TRUNC, 0777) = 3

171503 fallocate(3, 0, 0, 4096) = 0

171503 mmap(NULL, 4096, PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f0b5beae000

171503 close(3) = 0

171503 clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f0b5be638d0) = 171504

171504 set\_robust\_list(0x7f0b5be638e0, 24) = 0

171503 wait4(171504, <unfinished ...>

171504 gettid() = 171504

171504 getpid() = 171504

171504 tgkill(171504, 171504, SIGSTOP) = 0

171504 --- SIGSTOP {si\_signo=SIGSTOP, si\_code=SI\_TKILL, si\_pid=171504, si\_uid=1000} ---

171504 --- stopped by SIGSTOP ---

171503 <... wait4 resumed>NULL, WSTOPPED, NULL) = 171504

171503 --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_STOPPED, si\_pid=171504, si\_uid=1000, si\_status=SIGSTOP, si\_utime=0, si\_stime=0} ---

171503 newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x2), ...}, AT\_EMPTY\_PATH) = 0

171503 getrandom("\xc6\x15\x0b\x87\xbb\x80\x7d\x14", 8, GRND\_NONBLOCK) = 8

171503 brk(NULL) = 0x1f43000

171503 brk(0x1f64000) = 0x1f64000

171503 read(0, "100 1 2 3\n", 1024) = 10

171503 munmap(0x7f0b5beae000, 4096) = 0

171503 kill(171504, SIGCONT) = 0

171504 --- SIGCONT {si\_signo=SIGCONT, si\_code=SI\_USER, si\_pid=171503, si\_uid=1000} ---

171503 --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_CONTINUED, si\_pid=171504, si\_uid=1000, si\_status=SIGCONT, si\_utime=0, si\_stime=0} ---

171504 execve("divider", ["divider", "file.txt"], NULL <unfinished ...>

171503 wait4(-1, <unfinished ...>

171504 <... execve resumed>) = 0

171504 brk(NULL) = 0x1d03000

171504 arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffc75ade890) = -1 EINVAL (Недопустимый аргумент)

171504 access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (Нет такого файла или каталога)

171504 openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

171504 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=86775, ...}, AT\_EMPTY\_PATH) = 0

171504 mmap(NULL, 86775, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7feb605ef000

171504 close(3) = 0

171504 openat(AT\_FDCWD, "/lib64/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

171504 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\320\226\2\0\0\0\0\0"..., 832) = 832

171504 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

171504 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

171504 pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\202W\356\220vF\351\260W\31u3\321\344\254\216"..., 68, 928) = 68

171504 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2375624, ...}, AT\_EMPTY\_PATH) = 0

171504 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7feb605ed000

171504 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

171504 mmap(NULL, 2104720, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7feb60200000

171504 mmap(0x7feb60228000, 1523712, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7feb60228000

171504 mmap(0x7feb6039c000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19c000) = 0x7feb6039c000

171504 mmap(0x7feb603f4000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1f3000) = 0x7feb603f4000

171504 mmap(0x7feb603fa000, 32144, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7feb603fa000

171504 close(3) = 0

171504 mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7feb605ea000

171504 arch\_prctl(ARCH\_SET\_FS, 0x7feb605ea740) = 0

171504 set\_tid\_address(0x7feb605eaa10) = 171504

171504 set\_robust\_list(0x7feb605eaa20, 24) = 0

171504 rseq(0x7feb605eb0e0, 0x20, 0, 0x53053053) = 0

171504 mprotect(0x7feb603f4000, 16384, PROT\_READ) = 0

171504 mprotect(0x403000, 4096, PROT\_READ) = 0

171504 mprotect(0x7feb6063a000, 8192, PROT\_READ) = 0

171504 prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

171504 munmap(0x7feb605ef000, 86775) = 0

171504 openat(AT\_FDCWD, "file.txt", O\_RDWR) = 3

171504 mmap(NULL, 4096, PROT\_READ|PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7feb60639000

171504 ftruncate(3, 2) = 0

171504 exit\_group(0) = ?

171504 +++ exited with 0 +++

171503 <... wait4 resumed>[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 171504

171503 --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=171504, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

171503 openat(AT\_FDCWD, "file.txt", O\_RDONLY) = 3

171503 mmap(NULL, 4096, PROT\_READ, MAP\_SHARED, 3, 0) = 0x7f0b5beae000

171503 newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x2), ...}, AT\_EMPTY\_PATH) = 0

171503 write(1, "Result: 16\n", 11) = 11

171503 exit\_group(0) = ?

171503 +++ exited with 0 +++

**Пояснение**

171503 execve("./main", ["./main"], 0x7fff426e8088 /\* 88 vars \*/) = 0

**Исполнение файла main в текущем процессе.**

171503 openat(AT\_FDCWD, "file.txt", O\_RDWR|O\_CREAT|O\_TRUNC, 0777) = 3

171503 fallocate(3, 0, 0, 4096) = 0

171503 mmap(NULL, 4096, PROT\_WRITE, MAP\_SHARED, 3, 0) = 0x7f0b5beae000

171503 close(3) = 0

**Создание и маппирование файла.**

171503 clone(child\_stack=NULL, flags=CLONE\_CHILD\_CLEARTID|CLONE\_CHILD\_SETTID|SIGCHLD, child\_tidptr=0x7f0b5be638d0) = 171504

**Создание дочернего процесса через клонирование родительского.**

171503 wait4(171504, <unfinished ...>

171504 --- SIGSTOP {si\_signo=SIGSTOP, si\_code=SI\_TKILL, si\_pid=171504, si\_uid=1000} ---

171504 --- stopped by SIGSTOP ---

171503 <... wait4 resumed>NULL, WSTOPPED, NULL) = 171504

**Родительский процесс ждёт, пока дочерний сам себя остановит, после чего продолжается**

171503 read(0, "100 1 2 3\n", 1024) = 10

171503 munmap(0x7f0b5beae000, 4096) = 0

**Читаем данные из ввода и прекращаем маппировать файл**

171503 kill(171504, SIGCONT) = 0

171504 --- SIGCONT {si\_signo=SIGCONT, si\_code=SI\_USER, si\_pid=171503, si\_uid=1000} ---

171503 --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_CONTINUED, si\_pid=171504, si\_uid=1000, si\_status=SIGCONT, si\_utime=0, si\_stime=0} ---

171504 execve("divider", ["divider", "file.txt"], NULL <unfinished ...>

171503 wait4(-1, <unfinished ...>

**Отправляем сигнал на продолжение в дочерний процесс после окончания его работы и в родительском процессе ждём его завершения**

171504 exit\_group(0) = ?

171504 +++ exited with 0 +++

171503 <... wait4 resumed>[{WIFEXITED(s) && WEXITSTATUS(s) == 0}], 0, NULL) = 171504

171503 --- SIGCHLD {si\_signo=SIGCHLD, si\_code=CLD\_EXITED, si\_pid=171504, si\_uid=1000, si\_status=0, si\_utime=0, si\_stime=0} ---

**Дочерний процесс завершается, после чего родительский процесс продолжает свою работу**

171503 exit\_group(0) = ?

171503 +++ exited with 0 +++

**Работа завершена**

**Вывод**

Было диагностировано выполнение программы из лабораторной работы 3. В результате диагностики было определено, что происходят именно те вызовы, которые и требуются.

В процессе работы познакомился с программой strace, изучил некоторые её параметры, например слежение за дочерними процессами и запись результатов в файл.