Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №1 по курсу**

**«Операционные системы»**

Студент: Соколов Арсений Игоревич

Группа: М8О-207Б-21

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Демонстрация работы strace
4. Вывод

**Репозиторий**

https://github.com/valerasaray/os

**Постановка задачи**

При выполнении последующих лабораторных работ 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 должен содержать краткую

сводку по исследованию последующих ЛР.

**access**

access - это системный вызов в операционной системе Linux, который проверяет, существует ли файл с указанным путем и есть ли у текущего процесса права на его чтение/запись/исполнение. Агументы, переданные в access: путь к файлу и режим доступа (READ, WRITE, EXECUTE) для проверки.

**openat**

openat - это системный вызов в операционной системе Linux, который открывает файл или директорию по указанному пути относительно заданного каталога файловой системы, который определяется его файловым дескриптором. аргументы, переданные в openat: файловый дескриптор каталога, путь к файлу или директории, режим открытия файла (чтение, запись, создание), а также дополнительные флаги и права доступа.

**getrandom**

getrandom - это системный вызов в операционной системе Linux, который используется для получения случайных из источника случайности ядра операционной системы. аргументы, переданные в getrandom: указатель на буфер для записи случайных данных, количество запрашиваемых байтов и флаги, указывающие, какие дополнительные параметры использовать для генерации случайных данных. Возвращаемое значение getrandom указывает на количество собранных случайных байтов.

**Демонстрация работы strace (2 ЛР)**

strace ./main

execve("./main", ["./main"], 0x7fff18aa1ed0 /\* 68 vars \*/) = 0

brk(NULL) = 0x55d2b37c3000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7fffbd4323f0) = -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=144687, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 144687, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f9877b44000

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\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=19211048, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f9877b42000

mmap(NULL, 2324480, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f9877800000

mmap(0x7f987789a000, 1138688, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x9a000) = 0x7f987789a000

mmap(0x7f98779b0000, 487424, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1b0000) = 0x7f98779b0000

mmap(0x7f9877a27000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x226000) = 0x7f9877a27000

mmap(0x7f9877a35000, 10240, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f9877a35000

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\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=944536, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 946368, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f9877a5a000

mmap(0x7f9877a68000, 499712, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe000) = 0x7f9877a68000

mmap(0x7f9877ae2000, 385024, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x88000) = 0x7f9877ae2000

mmap(0x7f9877b40000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xe5000) = 0x7f9877b40000

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\0\0"..., 832) = 832

newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=596864, ...}, AT\_EMPTY\_PATH) = 0

mmap(NULL, 127304, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f9877a3a000

mmap(0x7f9877a3d000, 94208, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x3000) = 0x7f9877a3d000

mmap(0x7f9877a54000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1a000) = 0x7f9877a54000

mmap(0x7f9877a58000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d000) = 0x7f9877a58000

close(3) = 0

openat(AT\_FDCWD, "/usr/lib/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\0P9\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=1953112, ...}, 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, 1994096, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f9877619000

mmap(0x7f987763b000, 1417216, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7f987763b000

mmap(0x7f9877795000, 360448, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17c000) = 0x7f9877795000

mmap(0x7f98777ed000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1d4000) = 0x7f98777ed000

mmap(0x7f98777f3000, 52592, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f98777f3000

close(3) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f9877a38000

arch\_prctl(ARCH\_SET\_FS, 0x7f9877a39200) = 0

set\_tid\_address(0x7f9877a394d0) = 5599

set\_robust\_list(0x7f9877a394e0, 24) = 0

rseq(0x7f9877a39b20, 0x20, 0, 0x53053053) = 0

mprotect(0x7f98777ed000, 16384, PROT\_READ) = 0

mprotect(0x7f9877a58000, 4096, PROT\_READ) = 0

mprotect(0x7f9877b40000, 4096, PROT\_READ) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f9877617000

mprotect(0x7f9877a27000, 53248, PROT\_READ) = 0

mprotect(0x55d2b1e8f000, 4096, PROT\_READ) = 0

mprotect(0x7f9877b99000, 8192, PROT\_READ) = 0

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

munmap(0x7f9877b44000, 144687) = 0

getrandom("\x85\x4e\x8a\x88\xb2\x12\xd1\x3b", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55d2b37c3000

brk(0x55d2b37e4000) = 0x55d2b37e4000

futex(0x7f9877a356bc, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0

newfstatat(1, "", {st\_mode=S\_IFCHR|0600, st\_rdev=makedev(0x88, 0x2), ...}, AT\_EMPTY\_PATH) = 0

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\270\320\274\321\217 \320\264\320\273\321\217 \320\277\320"..., 75Введите имя для первого дочернего файла: ) = 75

newfstatat(0, "", {st\_mode=S\_IFCHR|0600, st\_rdev=makedev(0x88, 0x2), ...}, AT\_EMPTY\_PATH) = 0

read(0, 0x55d2b37d52c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0,

**Вывод**

Благодаря данной лабораторной работе, я приобрел навыки, необходимые для работы с strace, а также изучил системные вызовы.