Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №1 по курсу**

**«Операционные системы»**

**Использование утилиты strace**

Студент: Никулин Кристиан Ильич

Группа: М8О-208Б-21

Вариант: -

Преподаватель: Соколов Андрей Алексеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**Постановка задачи**

## Цель работы

Приобретение практических навыков диагностики работы программного обеспечения.

## Задание

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.

**Демонстрация работы с утилитой strace**

execve("./lab3", ["./lab3", "100", "10"], 0x7ffdb4739150 /\* 51 vars \*/) = 0

brk(NULL) = 0x56275d873000

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffe27a2d990) = -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

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=75998, ...}) = 0

mmap(NULL, 75998, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7f7637760000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libpthread.so.0", 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\220q\0\0\0\0\0\0"..., 832) = 832

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0{E6\364\34\332\245\210\204\10\350-\0106\343="..., 68, 824) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=157224, ...}) = 0

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f763775e000

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0{E6\364\34\332\245\210\204\10\350-\0106\343="..., 68, 824) = 68

mmap(NULL, 140408, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f763773b000

mmap(0x7f7637741000, 69632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x6000) = 0x7f7637741000

mmap(0x7f7637752000, 24576, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17000) = 0x7f7637752000

mmap(0x7f7637758000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1c000) = 0x7f7637758000

mmap(0x7f763775a000, 13432, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f763775a000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/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\0\300A\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\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\30x\346\264ur\f|Q\226\236i\253-'o"..., 68, 880) = 68

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2029592, ...}) = 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

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\0", 32, 848) = 32

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\30x\346\264ur\f|Q\226\236i\253-'o"..., 68, 880) = 68

mmap(NULL, 2037344, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7f7637549000

mmap(0x7f763756b000, 1540096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x22000) = 0x7f763756b000

mmap(0x7f76376e3000, 319488, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x19a000) = 0x7f76376e3000

mmap(0x7f7637731000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1e7000) = 0x7f7637731000

mmap(0x7f7637737000, 13920, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7f7637737000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7f7637546000

arch\_prctl(ARCH\_SET\_FS, 0x7f7637546740) = 0

mprotect(0x7f7637731000, 16384, PROT\_READ) = 0

mprotect(0x7f7637758000, 4096, PROT\_READ) = 0

mprotect(0x56275c60c000, 4096, PROT\_READ) = 0

mprotect(0x7f76377a0000, 4096, PROT\_READ) = 0

munmap(0x7f7637760000, 75998) = 0

set\_tid\_address(0x7f7637546a10) = 5324

set\_robust\_list(0x7f7637546a20, 24) = 0

rt\_sigaction(SIGRTMIN, {sa\_handler=0x7f7637741bf0, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_SIGINFO, sa\_restorer=0x7f763774f420}, NULL, 8) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7f7637741c90, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7f763774f420}, NULL, 8) = 0

rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0

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

brk(NULL) = 0x56275d873000

brk(0x56275d894000) = 0x56275d894000

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x3), ...}) = 0

write(1, "10 rounds for each thread\n\n", 27) = 27

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f7636d45000

mprotect(0x7f7636d46000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f7637544fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[0], tls=0x7f7637545700, child\_tidptr=0x7f76375459d0) = 5325

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f7636544000

mprotect(0x7f7636545000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f7636d43fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[5326], tls=0x7f7636d44700, child\_tidptr=0x7f7636d449d0) = 5326

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f7635d43000

mprotect(0x7f7635d44000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f7636542fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[5327], tls=0x7f7636543700, child\_tidptr=0x7f76365439d0) = 5327

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f7635542000

mprotect(0x7f7635543000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f7635d41fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[5328], tls=0x7f7635d42700, child\_tidptr=0x7f7635d429d0) = 5328

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f7634d41000

mprotect(0x7f7634d42000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f7635540fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[5329], tls=0x7f7635541700, child\_tidptr=0x7f76355419d0) = 5329

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f7634540000

mprotect(0x7f7634541000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f7634d3ffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[5330], tls=0x7f7634d40700, child\_tidptr=0x7f7634d409d0) = 5330

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f7633d3f000

mprotect(0x7f7633d40000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f763453efb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[5331], tls=0x7f763453f700, child\_tidptr=0x7f763453f9d0) = 5331

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f763353e000

mprotect(0x7f763353f000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f7633d3dfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[5332], tls=0x7f7633d3e700, child\_tidptr=0x7f7633d3e9d0) = 5332

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f7632d3d000

mprotect(0x7f7632d3e000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f763353cfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[5333], tls=0x7f763353d700, child\_tidptr=0x7f763353d9d0) = 5333

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7f763253c000

mprotect(0x7f763253d000, 8388608, PROT\_READ|PROT\_WRITE) = 0

clone(child\_stack=0x7f7632d3bfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[5334], tls=0x7f7632d3c700, child\_tidptr=0x7f7632d3c9d0) = 5334

munmap(0x7f7636d45000, 8392704) = 0

munmap(0x7f7636544000, 8392704) = 0

munmap(0x7f7635d43000, 8392704) = 0

munmap(0x7f7635542000, 8392704) = 0

munmap(0x7f7634d41000, 8392704) = 0

futex(0x7f7632d3c9d0, FUTEX\_WAIT, 5334, NULL) = 0

munmap(0x7f7634540000, 8392704) = 0

write(1, "\n", 1) = 1

write(1, "Monte-Carlo chance 19.91000\n", 28) = 28

write(1, "Real chance 0.05882\n", 27) = 27

exit\_group(0) = ?

+++ exited with 0 +++

Что делают системные вызовы:  
execve(…) - запускает программу  
openat(…) - открывает файл, имеет в качестве возвращаемого значения файловый дескриптор  
fstat(…) - собирает информацию из файла  
mmap(…) - отображает файл на память  
mprotect(…) - контролирует доступ к памяти  
close(…) - закрывает файловый дескриптор  
read(…) - считывает из файлового дескриптора  
munmap(…) - освобождает память, отведенную для отображения файла  
write(…) - пишет в консоль

pread64(…) – читает из файлового дескриптора

**Вывод**

В данной лабораторной работе я научился диагностировать программы с помощью strace и узнал о работе системных вызовов.