

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Институт №8 “Компьютерные науки и прикладная математика”
Кафедра №806 “Вычислительная математика и программирование”

Лабораторная работа №5 по курсу
«Операционные системы»

Группа: М8О-209БВ-24

Студент: Артонкин В. Н.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 19.12.25

Москва, 2025

Постановка задачи

Составить и отладить программу на языке Си, осуществляющую трассировку системных вызовов и анализ работы программ, разработанных в предыдущих четырёх лабораторных работах, с использованием утилиты strace (для Linux).

Изучить механизм системных вызовов операционной системы, научиться анализировать взаимодействие пользовательских программ с ядром ОС, понимать, как реализованы на системном уровне функции, использованные в предыдущих лабораторных работах.

Общий метод и алгоритм решения

Утилита **strace** предназначена для трассировки системных вызовов, выполняемых процессами в операционной системе Linux. Системные вызовы являются основным механизмом взаимодействия пользовательских программ с ядром ОС. Согласно классическим представлениям теории операционных систем, изложенным в книгах Э. Таненбаума и У. Вахалии, доступ к ресурсам компьютера осуществляется через системные вызовы, обеспечивающие переход из пользовательского режима в режим ядра.

Утилита strace позволяет наблюдать этот процесс на практике, отображая информацию о каждом системном вызове, включая его имя, переданные аргументы, возвращаемое значение и возможные ошибки. Работа strace основана на механизме, который позволяет одному процессу отслеживать выполнение другого. При каждом системном вызове выполнение трассируемой программы временно приостанавливается, после чего strace выводит соответствующую информацию.

Использование утилиты strace позволяет анализировать поведение программ, выявлять причины ошибок, изучать взаимодействие процессов с ядром операционной системы и наглядно исследовать принципы работы ОС в рамках лабораторных работ.

При работе с утилитой strace могут использоваться следующие **основные флаги**:

Флаг **-o <файл>** используется для сохранения вывода трассировки в указанный файл, что удобно для последующего анализа и оформления отчётов.

Флаг **-f** позволяет отслеживать системные вызовы дочерних процессов, создаваемых в ходе выполнения программы.

Флаги **-t** и **-tt** добавляют временные метки к системным вызовам, позволяя анализировать временные характеристики работы программы.

Флаг `-s <число>` задаёт максимальную длину выводимых строк.

Флаг -p <pid> применяется для подключения к уже запущенному процессу по его идентификатору.

Флаг `-e trace=<набор>` ограничивает вывод только указанными системными вызовами или их группами, что уменьшает объём выводимой информации.

Флаг -с выводит статистику по количеству и времени выполнения системных вызовов.

Таким образом, утилита strace является эффективным средством изучения системных вызовов и механизмов взаимодействия пользовательских программ с ядром операционной системы, что позволяет закрепить теоретические знания, полученные в курсе предмета «Операционные системы».

Выводы strace

Лабораторная работа №1

```
mmap(0x7fc030d9c000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c9000) = 0x7fc030d9c000

mmap(0x7fc030da2000, 13952, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fc030da2000

close(3) = 0

arch_prctl(ARCH_SET_FS, 0x7fc030da7540) = 0

mprotect(0x7fc030d9c000, 16384, PROT_READ) = 0

mprotect(0x55b953760000, 4096, PROT_READ) = 0

mprotect(0x7fc030dd8000, 4096, PROT_READ) = 0

munmap(0x7fc030da8000, 21784) = 0

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0

brk(NULL) = 0x55b979fae000

brk(0x55b979fcf000) = 0x55b979fcf000

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\270\320\274\321\217 1-\320\263\320\276 \321\204\320"..., 40) = 40

fstat(0, {st_mode=S_IFREG|0644, st_size=104, ...}) = 0

read(0, "fedya.txt\nvasya.txt\n1234567890\n1"..., 4096) = 104

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\270\320\274\321\217 2-\320\263\320\276 \321\204\320"..., 40) = 40

pipe([3, 4]) = 0

pipe([5, 6]) = 0

clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fc030da7810) = 17062

clone(child_stack=NULL,
flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7fc030da7810) = 17063

close(3) = 0

close(5) = 0

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\320\272\321\203 \320\270\320\273"..., 63) = 63

write(4, "1234567890\n", 11) = 11

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\320\272\321\203 \320\270\320\273"..., 63) = 63

write(6, "12345678901234567890\n", 21) = 21

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\320\272\321\203 \320\270\320\273"..., 63) = 63
```

```

write(6, "argentina manit negra\n", 22) = 22
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\320\272\321\203 \320\270\320\273"..., 63) = 63
write(6, "mama mila ramu\n", 15) = 15
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\320\272\321\203 \320\270\320\273"..., 63) = 63
write(4, "abc123deg\n", 10) = 10
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\320\272\321\203 \320\270\320\273"..., 63) = 63
close(4) = 0
close(6) = 0
wait4(-1, NULL, 0, NULL) = 17062
--- SIGCHLD {si_signo=SIGHLD, si_code=CLD_EXITED, si_pid=17062, si_uid=1000,
si_status=0, si_utime=0, si_stime=1} ---
wait4(-1, NULL, 0, NULL) = 17063
--- SIGCHLD {si_signo=SIGHLD, si_code=CLD_EXITED, si_pid=17063, si_uid=1000,
si_status=0, si_utime=0, si_stime=1} ---
exit_group(0) = ?
+++ exited with 0 +++

```

Лабораторная работа №2

```

execve("./sort", [ "./sort"], 0x7ffc45ddcd50 /* 32 vars */) = 0
brk(NULL) = 0x5570a88a4000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=16648, ...}) = 0
mmap(NULL, 16648, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f926e217000
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\0\3\0>\0\1\0\0\0 1\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=149520, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f926e215000
mmap(NULL, 136304, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f926e1f3000
mmap(0x7f926e1f9000, 65536, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x6000) = 0x7f926e1f9000

```



```
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\264\320\273\320\270\320\275\321\203 \320\274\320\260\321\201"..., 72) = 72
read(0, "16\n", 1024) = 3
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\272\320\276\320\273\320\270\321\207\320\265\321\201\321\202\320"..., 60) = 60
read(0, "4\n", 1024) = 2
write(1, "\320\240\320\265\320\266\320\270\320\274
\320\262\320\262\320\276\320\264\320\260: 1 \342\200\224 \320\262\321"..., 69) = 69
read(0, "2\n", 1024) = 2
write(1,
"\320\241\320\263\320\265\320\275\320\265\321\200\320\270\321\200\320\276\320\262\320
\260\320\275\320\275\321\213\320\271 \320"..., 45) = 45
write(1, "402 867 249 118 568 674 188 678"..., 63) = 63
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f926d81b000
mprotect(0x7f926d81c000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f926e01afb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SE
TTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[14957],
tls=0x7f926e01b700, child_tidptr=0x7f926e01b9d0) = 14957
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f926d01a000
mprotect(0x7f926d01b000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f926d819fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SE
TTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[14958],
tls=0x7f926d81a700, child_tidptr=0x7f926d81a9d0) = 14958
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f926c819000
mprotect(0x7f926c81a000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f926d018fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SE
TTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[14959],
tls=0x7f926d019700, child_tidptr=0x7f926d0199d0) = 14959
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f926c018000
mprotect(0x7f926c019000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f926c817fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SE
TTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[14960],
tls=0x7f926c818700, child_tidptr=0x7f926c8189d0) = 14960
futex(0x7f926e01b9d0, FUTEX_WAIT, 14957, NULL) = 0
futex(0x7f926d0199d0, FUTEX_WAIT, 14959, NULL) = 0
write(1, "\n", 1) = 1
```

```
write(1,
"\320\236\321\202\321\201\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320
\260\320\275\320\275\321\213\320\271 \320"..., 45) = 45

write(1, "78 96 118 188 208 249 402 568 59"..., 63) = 63

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

write(1, "\320\222\321\200\320\265\320\274\321\217
\321\201\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\272\320\270:"...
, 64) = 64

lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)

exit_group(0) = ?

+++ exited with 0 +++
```

Лабораторная работа №3

```
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f2656b31000
arch_prctl(ARCH_SET_FS, 0x7f2656b31740) = 0
set_tid_address(0x7f2656b31a10) = 8278
set_robust_list(0x7f2656b31a20, 24) = 0
rseq(0x7f2656b32060, 0x20, 0, 0x53053053) = 0
mprotect(0x7f2656d33000, 16384, PROT_READ) = 0
mprotect(0x5577683ca000, 4096, PROT_READ) = 0
mprotect(0x7f2656d81000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f2656d46000, 11939) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
getrandom("\x64\x07\xc4\xf9\x16\x25\xe3\x02", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x557788396000
brk(0x5577883b7000) = 0x5577883b7000
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\270\320\274\321\217 1-\320\263\320\276 \321\204\320"..., 40) = 40
fstat(0, {st_mode=S_IFREG|0644, st_size=104, ...}) = 0
read(0, "file1.txt\nfile2.txt\n1234567890\n1"..., 4096) = 104
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\270\320\274\321\217 2-\320\263\320\276 \321\204\320"..., 40) = 40
openat(AT_FDCWD, "/dev/shm/string_processor", O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC,
0666) = 3
ftruncate(3, 411208) = 0
mmap(NULL, 411208, PROT_READ|PROT_WRITE, MAP_SHARED, 3, 0) = 0x7f2656acc000
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f2656b31a10) = 8279
clone(child_stack=NULL, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f2656b31a10) = 8280
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\321\201\321\202\321\200\320\276\320\272\321\203 \320\270\320\273"..., 63) = 63
wait4(-1, NULL, 0, NULL) = 8279
```

```

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=8279, si_uid=1000,
si_status=0, si_utime=0, si_stime=0} ---

    wait4(-1, NULL, 0, NULL)          = 8280

--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_EXITED, si_pid=8280, si_uid=1000,
si_status=0, si_utime=0, si_stime=0} ---

    munmap(0x7f2656acc000, 411208)      = 0
    unlink("/dev/shm/string_processor")   = 0
    exit_group(0)                      = ?
+++ exited with 0 +++

```

Лабораторная работа №4

prog_static

```

17:26:57.632698 execve("./prog_static", ["./prog_static"], ["SHELL=/bin/bash",
"COLORTERM=truecolor", "TERM_PROGRAM_VERSION=1.107.0",
"HOSTNAME=1044750a6712",
"REMOTE_CONTAINERS_IPC=/tmp/vscode-remote-containers-ipc-50516a1e-05b0-4b1f-86
fe-cf69a6c69dcf.sock", "PWD=/workspaces/OS/lab4/src",
"VSCODE_GIT_ASKPASS_NODE=/vscode/vscode-server/bin/linux-x64/618725e67565b290
ba4da6fe2d29f8fa1d4e3622"..., "HOME=/home/vscode", "LANG=en_US.UTF-8",
"LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd
=40;33:01:or=40;31;0"..., "REMOTE_CONTAINERS=true",
"WAYLAND_DISPLAY=vscode-wayland-dcec0b54-2c91-4eff-9348-f39cd119df95.sock",
"GIT_ASKPASS=/vscode/vscode-server/bin/linux-x64/618725e67565b290ba4da6fe2d29f8fa1
d4e3622/extensions/"..., "VSCODE_GIT_ASKPASS_EXTRA_ARGS=",
"VSCODE_PYTHON_AUTOACTIVATE_GUARD=1", "LESSCLOSE=/usr/bin/lesspipe %s
%s", "TERM=xterm-256color",
"REMOTE_CONTAINERS_SOCKETS=[\"/tmp/.X11-unix/X1\", \"/home/vscode/.gnupg/S.gpg
-agent\"]", "LESSOPEN=| /usr/bin/lesspipe %s", "USER=vscode",
"VSCODE_GIT_IPC_HANDLE=/tmp/user/1000/vscode-git-86ac5bc9b1.sock",
"DISPLAY=:1", "SHLVL=2", "GIT_EDITOR=code --wait", "PROMPT_DIRTRIM=4",
"XDG_RUNTIME_DIR=/tmp/user/1000",
"VSCODE_GIT_ASKPASS_MAIN=/vscode/vscode-server/bin/linux-x64/618725e67565b290
ba4da6fe2d29f8fa1d4e3622"..., "BROWSER=/vscode/vscode-server/bin/linux-x64/618725e67565b290ba4da6fe2d29f8fa1d4e
3622/bin/helpers/bro"..., "PATH=/home/vscode/.vscode-server/data/User/globalStorage/github.copilot-chat/copilotCli:/h
ome/vscode"..., "REMOTE_CONTAINERS_DISPLAY_SOCK=/tmp/.X11-unix/X1",
"OLDPWD=/workspaces/OS/lab4", "TERM_PROGRAM=vscode",

```


17:26:57.651562 mmap(0x7f80fb4e000, 323584, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7f80fb4e000 <0.000117>

17:26:57.652418 mmap(0x7f80fb9d000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7f80fb9d000 <0.000098>

17:26:57.653266 mmap(0x7f80fbba3000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f80fbba3000 <0.000095>

17:26:57.654248 close(3) = 0 <0.000090>

17:26:57.655217 mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f80fb99b000 <0.000071>

17:26:57.655997 arch_prctl(ARCH_SET_FS, 0x7f80fb99b740) = 0 <0.000069>

17:26:57.656986 set_tid_address(0x7f80fb99ba10) = 31769 <0.000319>

17:26:57.658178 set_robust_list(0x7f80fb99ba20, 24) = 0 <0.000138>

17:26:57.659391 rseq(0x7f80fb99c060, 0x20, 0, 0x53053053) = 0 <0.000075>

17:26:57.660797 mprotect(0x7f80fb9d000, 16384, PROT_READ) = 0 <0.000075>

17:26:57.661515 mprotect(0x5616c6120000, 4096, PROT_READ) = 0 <0.000077>

17:26:57.662326 mprotect(0x7f80fbbeb000, 8192, PROT_READ) = 0 <0.000111>

17:26:57.663223 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0 <0.000097>

17:26:57.664100 munmap(0x7f80fbbe0000, 11939) = 0 <0.000126>

17:26:57.665111 fstat(0, {st_dev=makedev(0, 0x9c), st_ino=4, st_mode=S_IFCHR|0620, st_nlink=1,
st_uid=1000, st_gid=5, st_blksize=1024, st_blocks=0, st_rdev=makedev(0x88, 0x1),
st_atime=1765733217 /* 2025-12-14T17:26:57.534358669+0000 */, st_atime_nsec=534358669,
st_mtime=1765733217 /* 2025-12-14T17:26:57.534358669+0000 */, st_mtime_nsec=534358669,
st_ctime=1765718984 /* 2025-12-14T13:29:44.534358669+0000 */, st_ctime_nsec=534358669}) = 0
<0.000081>

17:26:57.665979 getrandom("\x77\x4a\x8b\xd7\x34\x94\xd8\x9a", 8, GRND_NONBLOCK) = 8
<0.000101>

17:26:57.666876 brk(NULL) = 0x5616f1ea3000 <0.000062>

17:26:57.667551 brk(0x5616f1ec4000) = 0x5616f1ec4000 <0.000073>

17:26:57.668219 read(0, "1 2 3\n", 1024) = 6 <2.037120>

17:26:59.706542 fstat(1, {st_dev=makedev(0, 0x9c), st_ino=4, st_mode=S_IFCHR|0620, st_nlink=1,
st_uid=1000, st_gid=5, st_blksize=1024, st_blocks=0, st_rdev=makedev(0x88, 0x1),
st_atime=1765733217 /* 2025-12-14T17:26:57.534358669+0000 */, st_atime_nsec=534358669,
st_mtime=1765733217 /* 2025-12-14T17:26:57.534358669+0000 */, st_mtime_nsec=534358669,
st_ctime=1765718984 /* 2025-12-14T13:29:44.534358669+0000 */, st_ctime_nsec=534358669}) = 0
<0.000083>

```
17:26:59.707439 write(1, "6.000000\n", 9) = 9 <0.000118>
17:26:59.708324 read(0, "2 3 5 6 3\n", 1024) = 10 <11.568834>
17:27:11.278274 write(1, "3 5 6\n", 6) = 6 <0.000113>
17:27:11.279118 read(0, "q\n", 1024) = 2 <3.471041>
17:27:14.751225 lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek) <0.000076>
17:27:14.752018 exit_group(0) = ?
17:27:14.753801 +++ exited with 0 +++
```

prog_dynamic

```
17:27:23.409279 execve("./prog_dynamic", ["/prog_dynamic"], 0x7ffc141898b8 /* 35 vars */) = 0
<0.004962>
17:27:23.416432 brk(NULL) = 0x55b6ab6cf000 <0.000071>
17:27:23.417289 mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f41f8e44000 <0.000071>
17:27:23.417986 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
<0.000086>
17:27:23.419018 openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v4/libc.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) <0.000480>
17:27:23.420778 openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v3/libc.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) <0.000370>
17:27:23.421852 openat(AT_FDCWD, "./glibc-hwcaps/x86-64-v2/libc.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory) <0.000401>
17:27:23.422956 openat(AT_FDCWD, "./libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory) <0.000359>
17:27:23.423912 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
<0.000068>
17:27:23.424608 fstat(3, {st_mode=S_IFREG|0644, st_size=11939, ...}) = 0 <0.000107>
17:27:23.425413 mmap(NULL, 11939, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f41f8e41000
<0.000087>
17:27:23.426255 close(3) = 0 <0.000080>
17:27:23.427103 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC)
= 3 <0.000120>
```

17:27:23.429859 fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0 <0.000065>

17:27:23.431504 mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0)
= 0x7f41f8c2f000 <0.000100>

17:27:23.432377 mmap(0x7f41f8c57000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f41f8c57000 <0.000144>

17:27:23.433129 mmap(0x7f41f8ddf000, 323584, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7f41f8ddf000 <0.000083>

17:27:23.433811 mmap(0x7f41f8e2e000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7f41f8e2e000 <0.000080>

17:27:23.434624 mmap(0x7f41f8e34000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS -1 0) = 0x7f41f8e34000 <0 000497>

17:27:23 435904 close(3) = 0 <0 000064>

17:27:23.436698 mmap(NULL, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS -1 0) = 0x7f41f8c2c000 <0 0000074>

17:27:23.437372 arch_prctl(ARCH_SET_ES, 0x7f41f8c2c740) = 0 <0 000060>

17:27:23.438017 set_tid_address(0x7f41f8c2ca10) ≡ 31957 <0 000056>

17:27:23.438636 set robust list(0x7f41f8c2ca20 24) = 0 <0 0000064>

17:27:23.439642 rseq(0x7f41f8c2d060, 0x20, 0, 0x53053053) = 0 < 0.000139>

17:27:23 440761 mprotect(0x7f41f8e2e000, 16384, PROT_READ) = 0 <0 000096>

17:27:23 441607 mprotect(0x55b6764e6000, 4096, PROT_READ) = 0 <0 000084>

17:27:23.465025 mmap(NULL, 16408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f41f8c22000 <0.000078>

17:27:23.465806 mmap(0x7f41f8c23000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7f41f8c23000 <0.000109>

17:27:23.466673 mmap(0x7f41f8c24000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f41f8c24000 <0.000099>

17:27:23.467476 mmap(0x7f41f8c25000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f41f8c25000 <0.000068>

17:27:23.469707 close(3) = 0 <0.000069>

17:27:23.470278 mprotect(0x7f41f8c25000, 4096, PROT_READ) = 0 <0.000067>

17:27:23.470841 fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
<0.000060>

17:27:23.471323 read(0, "1 2 3\n", 1024) = 6 <3.222835>

17:27:23.965502 fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
<0.000062>

17:27:23.966326 write(1, "6.000000\n", 9) = 9 <0.000145>

17:27:23.967195 read(0, "2 3 5 6 3\n", 1024) = 10 <9.926320>

17:27:33.894593 write(1, "3 5 6\n", 6) = 6 <0.000088>

17:27:33.895506 read(0, "0\n", 1024) = 2 <4.626970>

17:27:38.523633 read(0, "1 2 3\n", 1024) = 6 <2.362873>

17:27:40.887683 write(1, "3.000000\n", 9) = 9 <0.000095>

17:27:40.888532 read(0, "2 3 5 6 3\n", 1024) = 10 <6.684968>

17:27:47.574525 write(1, "3 5 6\n", 6) = 6 <0.000092>

17:27:47.575390 read(0, "q\n", 1024) = 2 <2.453838>

17:27:50.030268 munmap(0x7f41f8c27000, 16400) = 0 <0.000680>

17:27:50.031689 munmap(0x7f41f8c22000, 16408) = 0 <0.000633>

17:27:50.033081 lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek) <0.000088>

17:27:50.033922 exit_group(0) = ?

17:27:50.035516 +++ exited with 0 +++

Вывод

В ходе выполнения лабораторных работ с использованием утилиты **strace** были изучены основные механизмы взаимодействия пользовательских программ с ядром операционной системы Linux. На практике были проанализированы системные вызовы, связанные с созданием и завершением процессов, межпроцессным взаимодействием, перенаправлением потоков ввода-вывода, управлением памятью и обработкой сигналов. Использование утилиты **strace** позволило наглядно проследить работу системных вызовов и закрепить теоретические знания, полученные в рамках курса «Операционные системы»..