

**Московский авиационный институт
(национальный исследовательский университет)**

**Факультет информационных технологий и прикладной
математики**

Кафедра вычислительной математики и программирования

Лабораторная работа №1 по курсу «Операционные сети»

Тема: Диагностика работы программного обеспечения

Студент: Д. С. Ляшун
Преподаватель: Е. С. Миронов
Группа: М8О-207Б
Дата:
Оценка:
Подпись:

Москва, 2021

1 Постановка задачи

Цель работы: Приобретение практических знаний системных вызовов операционных сетей.

Задание: Продемонстрировать с помощью утилиты Linux strace использование ключевых системных вызовов программами из лабораторных работ 2-8, дать им объяснение.

2 Лабораторная работа №2

В ходе работы программы сперва вызывается функция `execv`, которая запускает файл программы на исполнение. Клонирование процесса происходит с помощью вызова `clone`, а смена его образа выполнения – также с помощью функции `execv`. В ходе работы программы также происходит переопределение файловых дескрипторов с помощью `dup2`, получение `id` процесса с помощью `getpid`. Печать и вывод в консоль с использованием функций `scanf` и `printf` реализуется соответственно с помощью системных вызовов `read` и `write`, где в аргументах указывается буфер, откуда читаются или записываются данные, их размер в байтах, а также дескриптор файла, откуда читаются или записываются данные (0 - стандартный поток ввода, 1 - стандартный поток вывода).

```
execve("./main",["./main"],0x7ffc6b6de238 /* 49 vars */) = 0
brk(NULL)                                     = 0x55bccdaff000
arch_prctl(0x3001 /* ARCH_??? */,0x7ffc84ff6ff0) = -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=81326,...) = 0
mmap(NULL,81326,PROT_READ,MAP_PRIVATE,3,0) = 0x7f35ab47a000
close(3)                                       = 0
close(3)                                       = 0
arch_prctl(ARCH_SET_FS,0x7f35ab479540) = 0
....
fstat(1,st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...) = 0
brk(NULL)                                     = 0x55bccdaff000
brk(0x55bccdb20000)                           = 0x55bccdb20000
fstat(0,st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...) = 0
write(1,"Enter file name: ",17Enter file name: ) = 17
read(0,tests1.txt
"tests1.txt
n",1024)                                       = 11
openat(AT_FDCWD,"tests1.txt",O_RDONLY) = 3
pipe([4,5])                                   = 0
clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|
SIGCHLD,child_tidptr=0x7f35ab479810) = 6817
getpid()                                       = 6816
write(1,"[6816] It's parent process of 68"... ,35[6816] It's parent process
of 6817
```

```

) = 35
close(5) = 0
read(4,strace: Process 6817 attached
<unfinished ...>
[pid 6817] getpid() = 6817
[pid 6817] write(1,"[6817] It's child process
n",26[pid 6817] It's child process
) = 26
[pid 6817] close(4) = 0
[pid 6817] dup2(5,1) = 1
[pid 6817] dup2(3,0) = 0
[pid 6817] execve("child",["child"],0x7ffc84ff70d8 /* 49 vars */) = 0
[pid 6817] brk(NULL) = 0x5590d4b19000
[pid 6817] arch_prctl(0x3001 /* ARCH_??? */ ,0x7ffd31d22540) = -1 EINVAL
(Недопустимый аргумент)
[pid 6817] access("/etc/ld.so.preload",R_OK) = -1 ENOENT
(Нет такого файла или каталога)
[pid 6817] openat(AT_FDCWD,"/etc/ld.so.cache",O_RDONLY|O_CLOEXEC) = 4
[pid 6817] fstat(4,st_mode=S_IFREG|0644,st_size=81326,...) = 0
[pid 6817] mmap(NULL,81326,PROT_READ,MAP_PRIVATE,4,0) = 0x7f66e9cbe000
[pid 6817] close(4) = 0
[pid 6817] openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY
|O_CLOEXEC) = 4
[pid 6817] read(4,"77ELF>60q
...",832) = 832
[pid 6817] pread64(4,"@@@
...",784,64) = 784
[pid 6817] pread64(4,"OGNU00
",32,848) = 32
[pid 6817] pread64(4,"4GNU6377?320070
704d45n55Y77
t34"... ,68,880) = 68
[pid 6817] fstat(4,st_mode=S_IFREG|0755,st_size=2029224,...) = 0
[pid 6817] mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,
-1,0) = 0x7f66e9cbc000
[pid 6817] pread64(4,"@@@
...",784,64) = 784
[pid 6817] pread64(4,"OGNU00",
32,848) = 32
[pid 6817] pread64(4,"4GNU6377?320070704d45n55Y77
t34"... ,68,880) = 68

```

```

[pid 6817] mmap(NULL,2036952,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,4,0)
= 0x7f66e9aca000
[pid 6817] mprotect(0x7f66e9aef000,1847296,PROT_NONE) = 0
[pid 6817] mmap(0x7f66e9aef000,1540096,PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,4,0x25000) = 0x7f66e9aef000
[pid 6817] mmap(0x7f66e9c67000,303104,PROT_READ,MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE,4,0x19d000) = 0x7f66e9c67000
[pid 6817] mmap(0x7f66e9cb2000,24576,PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,4,0x1e7000) = 0x7f66e9cb2000
[pid 6817] mmap(0x7f66e9cb8000,13528,PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1,0) = 0x7f66e9cb8000
[pid 6817] close(4) = 0
[pid 6817] arch_prctl(ARCH_SET_FS,0x7f66e9cbd540) = 0
[pid 6817] mprotect(0x7f66e9cb2000,12288,PROT_READ) = 0
[pid 6817] mprotect(0x5590d47d4000,4096,PROT_READ) = 0
[pid 6817] mprotect(0x7f66e9cff000,4096,PROT_READ) = 0
[pid 6817] munmap(0x7f66e9cbe000,81326) = 0
[pid 6817] fstat(0,st_mode=S_IFREG|0664,st_size=14,...) = 0
[pid 6817] brk(NULL) = 0x5590d4b19000
[pid 6817] brk(0x5590d4b3a000) = 0x5590d4b3a000
[pid 6817] read(0,"8
n10
n12
n2
n-10
n",4096) = 14
[pid 6817] write(1,"0",4 <unfinished ...>
[pid 6816] <... read resumed>"0",4) = 4
[pid 6817] <... write resumed> = 4
[pid 6816] write(1,"8 is composit number
n",218 is composit number
) = 21
[pid 6816] read(4, <unfinished ...>
[pid 6817] write(1,"
n",4 <unfinished ...>
[pid 6816] <... read resumed>"
n",4) = 4
[pid 6817] <... write resumed> = 4
[pid 6816] write(1,"10 is composit number
n",2210 is composit number
) = 22

```

```

[pid 6816] read(4, <unfinished ...>
[pid 6817] write(1,"
f",4 <unfinished ...>
[pid 6816] <... read resumed>"
f",4) = 4
[pid 6817] <... write resumed>          = 4
[pid 6816] write(1,"12 is composit number
n",2212 is composit number
) = 22
[pid 6816] read(4, <unfinished ...>
[pid 6817] lseek(0,-5,SEEK_CUR)          = 9
[pid 6817] exit_group(0)                  = ?
[pid 6816] <... read resumed>","",4)      = 0
[pid 6816] lseek(0,-1,SEEK_CUR)          = -1 ESPIPE (Недопустимая операция смещения)
[pid 6816] exit_group(0)                  = ?
[pid 6816] +++ exited with 0 +++
+++ exited with 0 +++

```

3 Лабораторная работа №3

В ходе работы программы видно, что для проведения замеров времени работы алгоритма вызывается функция `clock`, которая делает системный вызов функции `clock_gettime`, записывающей текущие показания часов выполнения данного процесса в наносекундах, разность полученных значений и будет являться временем работы.

В ходе запуска потоков видно, что это производится путем вызова системных функций: `mmap`, где указывается, что под каждый поток выделяется примерно 8 Мбайт памяти; `mprotect`, контролирующий доступ к области памяти, в нём указывается, что в выделенной памяти разрешается чтение и запись данных для создаваемого потока; и `clone`, который производит клонирование потоков с указанным образом выполнения (адреса исполняемой функции), возвращая `id` потока. Ожидания выполнения того или иного потока выполняется с помощью вызова `pthread_join`, в котором происходит системный вызов `futex`, использующийся для ожидания основным потоком изменения значения адреса указанной памяти и также пробуждения ожидающих выполнения потоков на указанном адресе.

```
execve("./prog",["./prog","2"],0x7ffc5899c3d0 /* 49 vars */) = 0
brk(NULL) = 0x55729e03d000
arch_prctl(0x3001 /* ARCH_??? */,0x7ffc7119f980) = -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(0,st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...) = 0
write(1,"Input numbrer to check: ",24Input numbrer to check: ) = 24
read(0,100
"100",1024) = 4
clock_gettime(CLOCK_PROCESS_CPUTIME_ID,tv_sec=0,tv_nsec=2831771) = 0
mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x7ff3d7f0e000
mprotect(0x7ff3d7f0f000,8388608,PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7ff3d870dfb0,flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|
CLONE_CHILD_CLEARPID,parent_tid=[3115],tls=0x7ff3d870e700,
child_tidptr=0x7ff3d870e9d0) = 3115
mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x7ff3d770d000
mprotect(0x7ff3d770e000,8388608,PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7ff3d7f0cfb0,flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
```

```

CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|
CLONE_CHILD_CLEARTID,parent_tid=[3116],tls=0x7ff3d7f0d700,
child_tidptr=0x7ff3d7f0d9d0) = 3116
futex(0x7ff3d870e9d0,FUTEX_WAIT,3115,NULLstrace: Process 3116 attached
<unfinished ...>
[pid 3116] set_robust_list(0x7ff3d7f0d9e0,24) = 0
[pid 3116] madvise(0x7ff3d770d000,8368128,MADV_DONTNEEDstrace: Process 3115
attached
<unfinished ...>
[pid 3115] set_robust_list(0x7ff3d870e9e0,24) = 0
[pid 3115] madvise(0x7ff3d7f0e000,8368128,MADV_DONTNEED) = 0
[pid 3116] <... madvise resumed>          = 0
[pid 3116] exit(0 <unfinished ...>
[pid 3115] exit(0 <unfinished ...>
[pid 3116] <... exit resumed>             = ?
[pid 3115] <... exit resumed>             = ?
[pid 3116] +++ exited with 0 +++
[pid 3115] +++ exited with 0 +++
<... futex resumed>                      = 0
clone(child_stack=0x7ff3d870dfb0,flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|
CLONE_CHILD_CLEARTID,parent_tid=[3117],tls=0x7ff3d870e700,
child_tidptr=0x7ff3d870e9d0) = 3117
mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x7ff3d6f0c000
mprotect(0x7ff3d6f0d000,8388608,PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7ff3d770bfb0,flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|
CLONE_CHILD_CLEARTID,parent_tid=[3118],tls=0x7ff3d770c700,
child_tidptr=0x7ff3d770c9d0) = 3118
futex(0x7ff3d870e9d0,FUTEX_WAIT,3117,NULLstrace: Process 3118 attached
<unfinished ...>
[pid 3118] set_robust_list(0x7ff3d770c9e0,24) = 0
[pid 3118] madvise(0x7ff3d6f0c000,8368128,MADV_DONTNEED) = 0
[pid 3118] exit(0)                        = ?
[pid 3118] +++ exited with 0 +++
strace: Process 3117 attached
[pid 3117] set_robust_list(0x7ff3d870e9e0,24) = 0
[pid 3117] madvise(0x7ff3d7f0e000,8368128,MADV_DONTNEED) = 0
[pid 3117] exit(0)                        = ?
[pid 3113] <... futex resumed>            = 0

```



```
[pid 3113] clock_gettime(CLOCK_PROCESS_CPUTIME_ID,tv_sec=0,tv_nsec=3889737)
= 0
[pid 3113] write(1,"Time of execution 1.058000 ms.
n",31Time of execution 1.058000 ms.
) = 31
[pid 3113] write(1,"100 is composite number.
n",25100 is composite number.
) = 25
[pid 3113] lseek(0,-1,SEEK_CUR) = -1
ESPIPE (Недопустимая операция смещения)
[pid 3113] exit_group(0) = ?
[pid 3117] +++ exited with 0 +++
+++ exited with 0 +++
```

4 Лабораторная работа №4

В ходе работы программы видно, что при открытии входного файла на чтение, а также при создании общего файла происходит вызов функции `openat`, где указывается имя файла (с добавлением информации о том, что он находится в текущем рабочем каталоге – ключ `AT_FDCWD`), режим работы, и возвращается дескриптор файла.

Создание дочернего процесса происходит с помощью системного вызова `clone`, далее происходит переопределение файловых дескрипторов стандартного потока ввода и файла с входными данными с помощью `dup2`, затем заменяется образ процесса вызовом `execv` с указанием имени программы `child` и её аргументом с именем общего файла.

Переопределение реакции программы на сигналы происходит с помощью `rt_sigaction` с указанием сигнала, действия при получении этого сигнала, а также возможное сохранение старого действия для сигнала. Далее происходит вызов `rt_sigprocmask`, который объединяет полученные набор сигналов с набором блокирующих сигналов текущего процесса. При ожидании получения сигналов вызывается `rt_sigtimedwait`, в котором указывается набор получаемых сигналов, а также место, куда будет записана информация о полученном сигнале (возвращаемое значение функции соответствует номеру полученного сигнала). В свою очередь отправление сигналов процессом осуществляется путем системного вызова `kill`, где указывается номер процесса-получателя, а также сам сигнал.

```
execve("./main",["./main"],0x7fff0e82d7c8 /* 49 vars */) = 0
brk(NULL)                                     = 0x564086053000
arch_prctl(0x3001 /* ARCH_??? */,0x7ffff8520260) = -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=67205,...) = 0
mmap(NULL,67205,PROT_READ,MAP_PRIVATE,3,0) = 0x7f7421d30000
close(3)                                       = 0
...
munmap(0x7f7421d30000,67205)                  = 0
set_tid_address(0x7f7421b0ba10)               = 4084
set_robust_list(0x7f7421b0ba20,24)            = 0
rt_sigaction(SIGRTMIN,sa_handler=0x7f7421d12bf0,sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO,sa_restorer=0x7f7421d203c0,NULL,8) = 0
rt_sigaction(SIGRT_1,sa_handler=0x7f7421d12c90,sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART| SA_SIGINFO,sa_restorer=0x7f7421d203c0,
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
fstat(1,st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...) = 0
brk(NULL) = 0x564086053000
brk(0x564086074000) = 0x564086074000
fstat(0,st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...) = 0
write(1,"Input file name: ",17Input file name: ) = 17
read(0,input.txt
"input.txt",1024) = 10
openat(AT_FDCWD,"input.txt",O_RDONLY) = 3
statfs("/dev/shm/",f_type=TMPFS_MAGIC,f_bsize=4096,f_blocks=393573,f_bfree=393573,
f_bavail=393573,f_files=393573,f_ffree=393572,f_fsid=val=[0,0],f_namelen=255,
f_frsize=4096,f_flags=ST_VALID|ST_NOSUID|ST_NODEV) = 0
futex(0x7f7421d2d390,FUTEX_WAKE_PRIVATE,2147483647) = 0
openat(AT_FDCWD,"/dev/shm/pipe",O_RDWR|O_CREAT|O_NOFOLLOW|O_CLOEXEC,0700) =
4
ftruncate(4,12) = 0
mmap(NULL,12,PROT_READ|PROT_WRITE,MAP_SHARED,4,0) = 0x7f7421d6d000
rt_sigprocmask(SIG_BLOCK,[USR1 USR2],NULL,8) = 0
clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0x7f7421b0ba10) = 4085
getpid() = 4084
write(1,"[4084] It's parent process of 40"... ,35[4084] It's parent process
of 4085
) = 35
rt_sigtimedwait([USR1 USR2],strace: Process 4085 attached
<unfinished ...>
[pid 4085] set_robust_list(0x7f7421b0ba20,24) = 0
[pid 4085] getpid() = 4085
[pid 4085] write(1,"[4085] It's child process
n",26[4085] It's child process
) = 26
[pid 4085] dup2(3,0) = 0
[pid 4085] execve("child",["child","pipe"],0x7ffff8520348 /* 49 vars */) =
0
[pid 4085] brk(NULL) = 0x55ffe637f000
[pid 4085] arch_prctl(0x3001 /* ARCH_??? */,0x7ffd28d29800)
= -1 EINVAL (Недопустимый аргумент)
[pid 4085] access("/etc/ld.so.preload",R_OK)
= -1 ENOENT (Нет такого файла или каталога)
[pid 4085] openat(AT_FDCWD,"/etc/ld.so.cache",O_RDONLY|O_CLOEXEC) = 4

```

```

[pid 4085] fstat(4,st_mode=S_IFREG|0644,st_size=67205,...) = 0
[pid 4085] mmap(NULL,67205,PROT_READ,MAP_PRIVATE,4,0) = 0x7fee42ccc000
[pid 4085] close(4) = 0
[pid 4085] openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libpthread.so.0",
O_RDONLY|O_CLOEXEC) = 4
[pid 4085] read(4,"77ELF>2001"
...,832) = 832
[pid 4085] pread64(4,"4GNU00574364B216442406@ 261327o"... ,68,824) = 68
[pid 4085] fstat(4,st_mode=S_IFREG|0755,st_size=157224,...) = 0
[pid 4085] mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x7fee42cca000
[pid 4085] pread64(4,"4GNU00574364B216442406@ 261327o"... ,68,824) = 68
[pid 4085] mmap(NULL,140408,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,4,0)
= 0x7fee42ca7000
[pid 4085] getppid() = 4084
[pid 4085] kill(4084,SIGUSR2 <unfinished ...>
[pid 4084] <... rt_sigtimedwait resumed>si_signo=SIGUSR2,si_code=SI_USER,
si_pid=4085,si_uid=1000,NULL,8) = 12 (SIGUSR2)
[pid 4085] <... kill resumed> = 0
[pid 4084] kill(4085,SIGUSR1 <unfinished ...>
[pid 4085] fstat(0, <unfinished ...>
[pid 4084] <... kill resumed> = 0
[pid 4085] <... fstat resumed>st_mode=S_IFREG|0664,st_size=161,...) = 0
[pid 4084] rt_sigtimedwait([USR1 USR2], <unfinished ...>
[pid 4085] brk(NULL) = 0x55ffe637f000
[pid 4085] brk(0x55ffe63a0000) = 0x55ffe63a0000
[pid 4085] read(0,"20
n10
n8
n8
n8
n10
n20
n33
n33
n33
n33
n40"... ,4096)
= 161
[pid 4085] rt_sigtimedwait([USR1],si_signo=SIGUSR1,si_code=SI_USER,
si_pid=4084,si_uid=1000,NULL,8) = 10 (SIGUSR1)

```

```

[pid 4085] getppid() = 4084
[pid 4085] kill(4084,SIGUSR1 <unfinished ...>
[pid 4084] <... rt_sigtimedwait resumed>si_signo=SIGUSR1,
si_code=SI_USER,si_pid=4085,si_uid=1000,NULL,8) = 10 (SIGUSR1)
[pid 4085] <... kill resumed> = 0
[pid 4084] write(1,"20
n10
n8
n8
n",10
20
10
8
8
...
[pid 4085] rt_sigtimedwait([USR1], <unfinished ...>
[pid 4084] <... write resumed> = 7
[pid 4084] kill(4085,SIGUSR1) = 0
[pid 4085] <... rt_sigtimedwait resumed>si_signo=SIGUSR1,si_code=SI_USER,
si_pid=4084,si_uid=1000,NULL,8) = 10 (SIGUSR1)
[pid 4084] rt_sigtimedwait([USR1 USR2], <unfinished ...>
[pid 4085] getppid() = 4084
[pid 4085] kill(4084,SIGUSR2 <unfinished ...>
[pid 4084] <... rt_sigtimedwait resumed>si_signo=SIGUSR2,si_code=SI_USER,
si_pid=4085,si_uid=1000,NULL,8) = 12 (SIGUSR2)
[pid 4085] <... kill resumed> = 0
[pid 4084] write(1,"8888
n8
n",7
8888
8
<unfinished ...>
[pid 4085] munmap(0x7fee42d09000,12 <unfinished ...>
[pid 4084] <... write resumed> = 7
[pid 4085] <... munmap resumed> = 0
[pid 4084] munmap(0x7f7421d6d000,12 <unfinished ...>
[pid 4085] lseek(0,-7,SEEK_CUR <unfinished ...>
[pid 4084] <... munmap resumed> = 0
[pid 4085] <... lseek resumed> = 154
[pid 4084] unlink("/dev/shm/pipe" <unfinished ...>
[pid 4085] exit_group(0 <unfinished ...>

```

```

[pid 4084] <... unlink resumed>          = 0
[pid 4085] <... exit_group resumed>      = ?
[pid 4084] close(3 <unfinished ...>
[pid 4085] +++ exited with 0 +++
<... close resumed>                      = 0
---SIGCHLD si_signo=SIGCHLD,si_code=CLD_EXITED,si_pid=4085,si_uid=1000,
si_status=0,si_utime=0,si_stime=0 ---
exit_group(0)                            = ?
+++ exited with 0 +++

```

5 Лабораторная работа №5

В ходе работы программы видно, что смена динамической библиотеки производится путем вызова системных функций: `openat`, где указывается имя открываемой библиотеки (с добавлением информации о том, что она находится в текущем рабочем каталоге – ключ `AT_FDCWD`) на чтение (ключ `O_RDONLY`), и возвращается дескриптор открытого файла-библиотеки; `read` - производит чтение данных из библиотеки (возможно, служебных); `fstat`, использующийся для получения информации по открытому файлу-библиотеке по дескриптору, которая затем используется в `mmap`; `mmap` осуществляет отображение содержимого файла в переменную программы для доступа к его символам.

```
execve("./solution",["./solution"],0x7ffff2df0060 /* 49 vars */) = 0
brk(NULL) = 0x56039f75f000
arch_prctl(0x3001 /* ARCH_??? */,0x7fff72e824e0) = -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=67205,...) = 0
mmap(NULL,67205,PROT_READ,MAP_PRIVATE,3,0) = 0x7fa8e5a22000
...
fstat(3,st_mode=S_IFREG|0775,st_size=18296,...) = 0
getcwd("/home/dmitry/Work_place/OS_labs/Lab5",128) = 37
mmap(NULL,16424,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7fa8e5a2e000
mmap(0x7fa8e5a2f000,4096,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x1000) = 0x7fa8e5a2f000
mmap(0x7fa8e5a30000,4096,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x2000) = 0x7fa8e5a30000
mmap(0x7fa8e5a31000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x2000) = 0x7fa8e5a31000
close(3) = 0
mprotect(0x7fa8e5a31000,4096,PROT_READ) = 0
fstat(0,st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...) = 0
read(0,1 20
"1 20",1024) = 5
fstat(1,st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...) = 0
write(1,"Pi(20) = 3.189185
n",18,Pi(20) = 3.189185
) = 18
read(0,0
"0",1024) = 2
```

```

munmap(0x7fa8e5a2e000,16424) = 0
openat(AT_FDCWD,"./realization2.so",O_RDONLY|O_CLOEXEC) = 3
read(3,"77ELF>'0"... ,832) = 832
fstat(3,st_mode=S_IFREG|0775,st_size=19432,...) = 0
getcwd("/home/dmitry/Work_place/OS_labs/Lab5",128) = 37
mmap(NULL,16432,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7fa8e5a2e000
mmap(0x7fa8e5a2f000,4096,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3,0x1000) = 0x7fa8e5a2f000
mmap(0x7fa8e5a30000,4096,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x2000)
= 0x7fa8e5a30000
mmap(0x7fa8e5a31000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3,0x2000) = 0x7fa8e5a31000
close(3) = 0
mprotect(0x7fa8e5a31000,4096,PROT_READ) = 0
write(1,"Change dynamic library from 1 to"... ,35Change dynamic library from
1 to 2
) = 35
read(0,2 4 8 12 30 18
"2 4 8 12 30 18",1024) = 15
write(1,"Sort(array) = [ 8,12,18,30 ]",32Sort(array) = [ 8,12,18,30 ]
) = 32
read(0,1 40
"1 40",1024) = 5
write(1,"Pi(40) = 3.122260",18Pi(40) = 3.122260
) = 18
read(0,^Cstrace: Process 4276 detached
<detached ...>

```


6 Лабораторная работа №6-8

В ходе выполнения программы видно, что получение сообщений из сокета осуществляется с помощью системного вызова `recvfrom`, а отправка – с помощью вызова `sendto`. Также при ожидании отправки/получении сообщений происходит вызов `poll` с установленным временем блокировки.

```
....
[pid 6922] rt_sigprocmask(SIG_BLOCK,~[RTMIN RT_1], <unfinished ...>
[pid 6921] <... fcntl resumed>          = 0x802 (flags O_RDWR|O_NONBLOCK)
strace: Process 6923 attached
[pid 6920] <... epoll_wait resumed>[EPOLLIN,u32=1879088064,u64=140687827835840],
256,-1) = 1
[pid 6923] epoll_wait(7, <unfinished ...>
[pid 6920] recvfrom(10, <unfinished ...>
[pid 6923] <... epoll_wait resumed>[EPOLLOUT,u32=1610618672,u64=140451336165168],
256,-1) = 1
[pid 6920] <... recvfrom resumed>"^~3 3",8192,0,NULL,NULL)
= 14
[pid 6923] epoll_ctl(7,EPOLL_CTL_MOD,9,EPOLLIN,u32=1610618672,u64=140451336165168
<unfinished ...>
[pid 6920] write(8,"",8 <unfinished ...>
[pid 6918] <... poll resumed>          = 1 ([fd=8,revents=POLLIN])
[pid 6923] <... epoll_ctl resumed>     = 0
[pid 6920] <... write resumed>         = 8
[pid 6918] read(8, <unfinished ...>
[pid 6923] epoll_wait(7, <unfinished ...>
[pid 6920] epoll_wait(7, <unfinished ...>
[pid 6918] <... read resumed>"",8) = 8
[pid 6918] poll([fd=8,events=POLLIN],1,0) = 0 (Timeout)
[pid 6918] fstat(1,st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0),...) = 0
[pid 6918] write(1,"OK: 6921",9OK: 6921
) = 9
[pid 6918] read(0,ping 1
"ping 1",1024) = 7
[pid 6918] poll([fd=8,events=POLLIN],1,0) = 0 (Timeout)
[pid 6918] write(6,"",8) = 8
[pid 6920] <... epoll_wait resumed>[EPOLLIN,u32=4025681696,u64=93874830903072],
256,-1) = 1
[pid 6918] poll([fd=8,events=POLLIN],1,1000 <unfinished ...>
[pid 6920] poll([fd=6,events=POLLIN],1,0) = 1 ([fd=6,revents=POLLIN])
```

```

[pid 6920] read(6,"",8) = 8
[pid 6920] epoll_ctl(7,EPLL_CTL_MOD,10,EPLLIN|EPOLLOUT,u32=1879088064,
u64=140687827835840) = 0
[pid 6920] sendto(10,"^",14,0,NULL,0 <unfinished ...>
[pid 6923] <... epoll_wait resumed>[EPOLLIN,u32=1610618672,u64=140451336165168],
256,-1) = 1
[pid 6920] <... sendto resumed>          = 14
[pid 6923] recvfrom(9, <unfinished ...>
[pid 6920] poll([fd=6,events=POLLIN],1,0 <unfinished ...>
...
[pid 6923] epoll_ctl(7,EPLL_CTL_MOD,9,EPLLIN|EPOLLOUT,u32=1610618672,
u64=140451336165168) = 0
[pid 6923] sendto(9,"^",14,0,NULL,0) = 14
[pid 6920] <... epoll_wait resumed>[EPOLLIN,u32=1879088064,u64=140687827835840],
256,-1) = 1
[pid 6923] poll([fd=6,events=POLLIN],1,0 <unfinished ...>
[pid 6920] recvfrom(10, <unfinished ...>
[pid 6923] <... poll resumed>          = 0 (Timeout)
[pid 6920] <... recvfrom resumed>"^",8192,0,NULL,
NULL) = 14
[pid 6923] epoll_wait(7, <unfinished ...>
[pid 6920] write(8,"",8 <unfinished ...>
[pid 6918] <... poll resumed>          = 1 ([fd=8,events=POLLIN])
[pid 6923] <... epoll_wait resumed>[EPOLLOUT,u32=1610618672,u64=140451336165168],
256,-1) = 1
[pid 6920] <... write resumed>          = 8
[pid 6918] read(8, <unfinished ...>
[pid 6923] epoll_ctl(7,EPLL_CTL_MOD,9,EPLLIN,u32=1610618672,u64=140451336165168
<unfinished ...>
[pid 6920] epoll_wait(7, <unfinished ...>
[pid 6918] <... read resumed>"",8) = 8
[pid 6923] <... epoll_ctl resumed>      = 0
[pid 6918] poll([fd=8,events=POLLIN],1,0 <unfinished ...>
[pid 6923] epoll_wait(7, <unfinished ...>
[pid 6918] <... poll resumed>          = 0 (Timeout)
[pid 6918] write(1,"OK: 1",60K: 1
)      = 6

```

7 Выводы

В результате выполнения данной лабораторной работы я познакомился с утилитой `strace` и её возможностями отслеживания всех системных вызовов программами. Как мне кажется, это довольно удобно использовать при отладки программ и изучении их процесса работы, однако для большего понимания таких логов необходимо иметь представление обо всех вызываемых системных вызовах, например для чего используется тот или иной передаваемый аргумент в них.

Также я также узнал, как используются системные вызовы при работе с теми или иными технологиями операционной системы. Например, при создании дочернего процесса с помощью `fork` происходит системный вызов `clone`, при печати/чтении данных с помощью `printf/scanf` вызываются соответственно `read/write` и т. д.

Список литературы

[1] *Linux Man Pages*

URL: <http://ru.manpages.org> (дата обращения: 18.12.2020).