

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: А. О. Дубинин  
Преподаватель: Е. С. Миронов  
Группа: М8О-206Б  
Дата:  
Оценка:  
Подпись:

Москва, 2018

# Лабораторная работа №1

**Цель работы:** Приобретение практических навыков диагностики работы программного обеспечения.

**Задача:** При выполнении последующих лабораторных работ необходимо продемонстрировать ключевые системные вызовы, которые в них используются и то, что их использование соответствует варианту ЛР. Отчет содержит краткую сводку по исследованию последующих ЛР.

# 1 Описание

Для диагностики была использована утилита strace. Strace - это утилита, отслеживающая системные вызовы. Системные вызовы являются интерфейсом между процессом и операционной системой.

В лабораторных были использованы следующие системные вызовы:

pid_t fork(void);	Создает дочерний процесс. Если возвращает 0, то созданный процесс – ребенок, если > 0, то – родитель.
pid_t waitpid(pid_t pid, int *status, int options);	Приостанавливает выполнение текущего процесса до тех пор, пока дочерний процесс не завершится, или до появления сигнала, который либо завершает текущий процесс, либо требует вызвать функцию-обработчик.
int pipe(int *fd);	предоставляет средства передачи данных между двумя процессами.
int close(int fd);	Закрывает файловый дескриптор
ssize_t write(int fd, const void *buf, size_t count)	Запись в файл по его описателю. Возвращается количество успешно записанных байтов
ssize_t read(int fd, void *buf, size_t count)	Чтение из файла. Возвращается число списанных байтов
int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void (*start_routine)(void *), void *arg);	Создание нового потока. При успешном завершении функция возвращает 0, в противном случае - код ошибки.
int pthread_mutex_init(pthread_mutex_t *mutex, const pthread_mutexattr_t *attr);	Инициализация мьютекса, при успешном завершении функция возвращает 0, в противном случае - код ошибки.
int pthread_mutex_lock(pthread_mutex_t *mutex);	Захват мьютекса, при успешном завершении функция возвращает 0, в противном случае - код ошибки.
int pthread_join(pthread_t thread, void **value_ptr);	Ожидание завершения потока управления. При успешном завершении функция возвращает 0, в противном случае - код ошибки.

<code>void * mmap(void *start, size_t length, int prot, int flags, int fd, off_t offset);</code>	Функция <code>mmap</code> отражает <code>length</code> байтов, начиная со смещения <code>offset</code> файла (или другого объекта), определенного файловым дескриптором <code>fd</code> , в память, начиная с адреса <code>start</code> . Последний параметр (адрес) необязателен, и обычно бывает равен 0. Настоящее местоположение отраженных данных возвращается самой функцией <code>mmap</code> , и никогда не бывает равным 0.
<code>int sigemptyset(sigset_t *set);</code>	<code>sigemptyset</code> инициализирует набор сигналов, указанный в <code>set</code> , и "очищает" его от всех сигналов.
<code>int sigaddset(sigset_t *set, int signum);</code>	<code>sigaddset</code> добавляет сигналы <code>signum</code> к <code>set</code> и удаляет эти сигналы из набора соответственно.
<code>int sigprocmask(int how, const sigset_t *set, sigset_t *oldset);</code>	Системный вызов <code>sigprocmask</code> используется для того, чтобы изменить список заблокированных в данный момент сигналов.
<code>int sigwait(const sigset_t *set, int *sig);</code>	Функция <code>sigwait()</code> приостанавливает выполнение вызвавшей нити до тех пор, пока в ожидании не появится сигнал из заданного набора сигналов <code>set</code> . Функция принимает сигнал (удаляет его из списка ожидающих сигналов) и возвращает номер сигнала в <code>sig</code> .
<code>int kill(pid_t pid, int sig);</code>	Системный вызов <code>kill</code> может быть использован для послылки какого-либо сигнала какому-либо процессу или группе процесса.
<code>void *dlopen(const char *filename, int flag);</code>	<code>dlopen</code> загружает динамическую библиотеку, имя которой указано в строке <code>filename</code> , и возвращает прямой указатель на начало динамической библиотеки.

<pre>void *dlsym(void *handle, char *symbol);</pre>	<p>dlsym использует указатель на динамическую библиотеку, возвращаемую dlopen, и оканчивающееся нулем символьное имя, а затем возвращает адрес, указывающий, откуда загружается этот символ. Если символ не найден, то возвращаемым значением dlsym является NULL</p>
---	---

## 2 Диагностика strace

### Лабораторная работа №2:

```
art@mars:~/study/semester_3/OS/lab_2/variant_22$ strace ./main
```

```
...
close(3)                                = 0
arch_prctl(ARCH_SET_FS,0x7f2bfbdd3500) = 0
mprotect(0x7f2bfbbbd000,16384,PROT_READ) = 0
mprotect(0x55a2157f2000,4096,PROT_READ) = 0
mprotect(0x7f2bfbdde000,4096,PROT_READ) = 0
munmap(0x7f2bfbdd4000,105825)           = 0
pipe([3,4])                            = 0
clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,child_tid
= 15014
close(4)                                = 0
read(3,Write a command(<h>for help):
q
"q",1)                                  = 1
close(3)                                = 0
---SIGCHLD si_signo=SIGCHLD,si_code=CLD_EXITED,si_pid=15014,si_uid=1000,si_status=0,s
---
exit_group(0)                           = ?
+++ exited with 0 +++
```

### Лабораторная работа №3:

```
art@mars:~/workdir/OS/lab_3/cmake-build-debug$ strace ./lab_3 ../text.txt
```

```
execve("./lab_3",["./lab_3","../text.txt"],0x7ffcd4b91718 /* 56 vars */) =
0
brk(NULL)                               = 0x55dbea895000
access("/etc/ld.so.nohwcap",F_OK)        = -1 ENOENT (No such file or directory)
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=105825,...) = 0
mmap(NULL,105825,PROT_READ,MAP_PRIVATE,3,0) = 0x7f91dea7a000
close(3)                                 = 0
access("/etc/ld.so.nohwcap",F_OK)        = -1 ENOENT (No such file or directory)
```

```

openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libpthread.so.0",O_RDONLY|O_CLOEXEC)
= 3
read(3,"77ELF>000b"... ,832) = 832
fstat(3,st_mode=S_IFREG|0755,st_size=144976,...) = 0
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7f91dea78000
mmap(NULL,2221184,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7f91de64e000
mprotect(0x7f91de668000,2093056,PROT_NONE) = 0
mmap(0x7f91de867000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7f91de867000
mmap(0x7f91de869000,13440,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1,0)
= 0x7f91de869000
close(3) = 0
access("/etc/ld.so.nohwcap",F_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY|O_CLOEXEC) = 3
read(3,"77ELF>604"... ,832) = 832
fstat(3,st_mode=S_IFREG|0755,st_size=2030544,...) = 0
mmap(NULL,4131552,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7f91de25d000
mprotect(0x7f91de444000,2097152,PROT_NONE) = 0
mmap(0x7f91de644000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7f91de644000
mmap(0x7f91de64a000,15072,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1,0)
= 0x7f91de64a000
close(3) = 0
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7f91dea75000
arch_prctl(ARCH_SET_FS,0x7f91dea75740) = 0
mprotect(0x7f91de644000,16384,PROT_READ) = 0
mprotect(0x7f91de867000,4096,PROT_READ) = 0
mprotect(0x55dbea451000,4096,PROT_READ) = 0
mprotect(0x7f91dea94000,4096,PROT_READ) = 0
munmap(0x7f91dea7a000,105825) = 0
set_tid_address(0x7f91dea75a10) = 14122
set_robust_list(0x7f91dea75a20,24) = 0
rt_sigaction(SIGRTMIN,sa_handler=0x7f91de653cb0,sa_mask=[],sa_flags=SA_RESTORER|SA_SIGINFO)
= 0
rt_sigaction(SIGRT_1,sa_handler=0x7f91de653d50,sa_mask=[],sa_flags=SA_RESTORER|SA_SIGINFO)
= 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) = 0x55dbea895000
brk(0x55dbea8b6000) = 0x55dbea8b6000

```

```

openat(AT_FDCWD,"../text.txt",O_RDONLY) = 3
fstat(3,st_mode=S_IFREG|0664,st_size=387,...) = 0
read(3,"202220762100207620752065 206320642065-21022076 206120762063 "...,4096)
= 387
read(3,"",4096) = 0
lseek(3,0,SEEK_SET) = 0
read(3,"202220762100207620752065 206320642065-21022076 206120762063 "...,4096)
= 387
read(3,"",4096) = 0
mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0x7f91dda5c000
mprotect(0x7f91dda5d000,8388608,PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f91de25bfb0,flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CL
= 14123
futex(0x7f91de25c9d0,FUTEX_WAIT,14123,NULL) = 0
munmap(0x7f91dc259000,8392704) = 0
fstat(1,st_mode=S_IFCHR|0620,st_rdev=makedev(136,0),...) = 0
//Вывод ответа
...
close(3) = 0
close(1) = 0
exit_group(0) = ?
+++ exited with 0 +++
art@mars:~/workdir/OS/lab_3/cmake-build-debug$

```

#### Лабораторная работа №4:

```

art@mars:~/workdir/OS/lab_4/cmake-build-debug$ strace ./variant_22
...
//отображение файла
mmap(NULL,100,PROT_READ|PROT_WRITE,MAP_SHARED|MAP_ANONYMOUS,-1,0) = 0x7f376d2ee000
rt_sigprocmask(SIG_BLOCK,[USR1 USR2],NULL,8) = 0
getpid() = 14762
pipe([3,4]) = 0
clone(child_stack=NULL,flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,child_tid
= 14763
close(4) = 0
Write a command(<h>for help):
rt_sigtimedwait([USR1 USR2],q
si_signo=SIGUSR1,si_code=SI_USER,si_pid=14763,si_uid=1000,NULL,8) = 10 (SIGUSR1)
close(3) = 0

```



```

---SIGCHLD si_signo=SIGCHLD,si_code=CLD_EXITED,si_pid=14763,si_uid=1000,si_status=0,s
---
exit_group(0)                                = ?
+++ exited with 0 +++
art@mars:~/workdir/OS/lab_4/cmake-build-debug$

```

## Лабораторная работа №5:

```

art@mars:~/workdir/OS/lab_5$ strace ./dyn
execve("./dyn",["./dyn"],0x7ffe93271c40 /* 56 vars */) = 0
brk(NULL)                                = 0x55fa5be33000
access("/etc/ld.so.nohwcap",F_OK)         = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload",R_OK)         = -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"./tls/haswell/x86_64/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD,"./tls/haswell/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD,"./tls/x86_64/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./tls/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./haswell/x86_64/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD,"./haswell/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./x86_64/libdl.so.2",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./libdl.so.2",O_RDONLY|O_CLOEXEC) = -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=105825,...) = 0
mmap(NULL,105825,PROT_READ,MAP_PRIVATE,3,0) = 0x7fde666a5000
close(3)                                = 0
access("/etc/ld.so.nohwcap",F_OK)         = -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libdl.so.2",O_RDONLY|O_CLOEXEC) = 3
read(3,"77ELF>P6"... ,832) = 832
fstat(3,st_mode=S_IFREG|0644,st_size=14560,...) = 0
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7fde666a3000
mmap(NULL,2109712,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7fde66294000
mprotect(0x7fde66297000,2093056,PROT_NONE) = 0

```

```

mmap(0x7fde66496000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7fde66496000
close(3) = 0
openat(AT_FDCWD,"./tls/haswell/x86_64/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD,"./tls/haswell/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./tls/x86_64/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./tls/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./haswell/x86_64/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD,"./haswell/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./x86_64/libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./libc.so.6",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file
or directory)
access("/etc/ld.so.nohwcap",F_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY|O_CLOEXEC) = 3
read(3,"77ELF>604"... ,832) = 832
fstat(3,st_mode=S_IFREG|0755,st_size=2030544,...) = 0
mmap(NULL,4131552,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7fde65ea3000
mprotect(0x7fde6608a000,2097152,PROT_NONE) = 0
mmap(0x7fde6628a000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7fde6628a000
mmap(0x7fde66290000,15072,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1,0)
= 0x7fde66290000
close(3) = 0
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7fde666a0000
arch_prctl(ARCH_SET_FS,0x7fde666a0740) = 0
mprotect(0x7fde6628a000,16384,PROT_READ) = 0
mprotect(0x7fde66496000,4096,PROT_READ) = 0
mprotect(0x55fa5a289000,4096,PROT_READ) = 0
mprotect(0x7fde666bf000,4096,PROT_READ) = 0
munmap(0x7fde666a5000,105825) = 0
openat(AT_FDCWD,"./tls/haswell/x86_64/libtree.so",O_RDONLY|O_CLOEXEC) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD,"./tls/haswell/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)

```

```

openat(AT_FDCWD,"./tls/x86_64/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./tls/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./haswell/x86_64/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT
(No such file or directory)
openat(AT_FDCWD,"./haswell/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
openat(AT_FDCWD,"./x86_64/libtree.so",O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such
file or directory)
openat(AT_FDCWD,"./libtree.so",O_RDONLY|O_CLOEXEC) = 3
read(3,"77ELF>60"... ,832) = 832
brk(NULL) = 0x55fa5be33000
brk(0x55fa5be54000) = 0x55fa5be54000
fstat(3,st_mode=S_IFREG|0755,st_size=12920,...) = 0
getcwd("/home/art/study/semester_3/OS/lab_5",128) = 36
mmap(NULL,2105528,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7fde65ca0000
mprotect(0x7fde65ca2000,2093056,PROT_NONE) = 0
mmap(0x7fde65ea1000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7fde65ea1000
close(3) = 0
mprotect(0x7fde65ea1000,4096,PROT_READ) = 0
fstat(1,st_mode=S_IFCHR|0620,st_rdev=makedev(136,0),...) = 0
write(1,"Commands",9Commands
) = 9
write(1,"",1
) = 1
write(1,"1 -Insert",111 -Insert
) = 11
write(1,"2 -Search",112 -Search
) = 11
write(1,"3 -Delete",113 -Delete
) = 11
write(1,"4 -Print tree",154 -Print tree
) = 15
write(1,"0 -Quit",90 -Quit
) = 9
write(1,"",1
) = 1
write(1,"Command: ",10Command:
) = 10

```

```

fstat(0,st_mode=S_IFCHR|0620,st_rdev=makedev(136,0),...) = 0
read(0,0
"0",1024) = 2
exit_group(0) = ?
+++ exited with 0 +++

```

## Лабораторная работа №6:

```
//Client:
```

```

...
read(3,"# Internet (IP) protocols## Up"... ,4096) = 2932
read(3,"",4096) = 0
close(3) = 0
eventfd2(0,EFD_CLOEXEC) = 3
fcntl(3,F_GETFL) = 0x2 (flags O_RDWR)
fcntl(3,F_SETFL,O_RDWR|O_NONBLOCK) = 0
fcntl(3,F_GETFL) = 0x802 (flags O_RDWR|O_NONBLOCK)
fcntl(3,F_SETFL,O_RDWR|O_NONBLOCK) = 0
getrandom("61630438435d35849e51281b3e",16,0) = 16
getrandom("6398044d506b7841f8160872b",16,0) = 16
eventfd2(0,EFD_CLOEXEC) = 4
fcntl(4,F_GETFL) = 0x2 (flags O_RDWR)
fcntl(4,F_SETFL,O_RDWR|O_NONBLOCK) = 0
fcntl(4,F_GETFL) = 0x802 (flags O_RDWR|O_NONBLOCK)
fcntl(4,F_SETFL,O_RDWR|O_NONBLOCK) = 0
epoll_create1(EPoll_CLOEXEC) = 5
epoll_ctl(5,EPOLL_CTL_ADD,4,0,u32=2250577952,u64=94641854945312) = 0
epoll_ctl(5,EPOLL_CTL_MOD,4,EPOLLIN,u32=2250577952,u64=94641854945312) = 0
mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0x7f0fe3033000
mprotect(0x7f0fe3034000,8388608,PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f0fe3832b70,flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_PARENT|CLONE_THREAD) = 14882
futex(0x7f0fe438207c,FUTEX_WAKE_PRIVATE,2147483647) = 0
futex(0x7f0fe4382088,FUTEX_WAKE_PRIVATE,2147483647) = 0
openat(AT_FDCWD,"/proc/self/task/14882/comm",O_RDWR) = 6
write(6,"ZMQbg/0",7) = 7
close(6) = 0
eventfd2(0,EFD_CLOEXEC) = 6
fcntl(6,F_GETFL) = 0x2 (flags O_RDWR)
fcntl(6,F_SETFL,O_RDWR|O_NONBLOCK) = 0
fcntl(6,F_GETFL) = 0x802 (flags O_RDWR|O_NONBLOCK)

```

```

fcntl(6,F_SETFL,O_RDWR|O_NONBLOCK)      = 0
epoll_create1(EPOCH_CLOEXEC)              = 7
epoll_ctl(7,EPOCH_CTL_ADD,6,0,u32=2250593392,u64=94641854960752) = 0
epoll_ctl(7,EPOCH_CTL_MOD,6,EPOCHLIN,u32=2250593392,u64=94641854960752) = 0
mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0x7f0fe2832000
mprotect(0x7f0fe2833000,8388608,PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f0fe3031b70,flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD)
= 14883
openat(AT_FDCWD,"/proc/self/task/14883/comm",O_RDWR) = 8
write(8,"ZMQbg/1",7)                        = 7
close(8)                                    = 0
eventfd2(0,EPOCH_CLOEXEC)                  = 8
fcntl(8,F_GETFL)                           = 0x2 (flags O_RDWR)
fcntl(8,F_SETFL,O_RDWR|O_NONBLOCK)         = 0
fcntl(8,F_GETFL)                           = 0x802 (flags O_RDWR|O_NONBLOCK)
fcntl(8,F_SETFL,O_RDWR|O_NONBLOCK)         = 0
poll([fd=8,events=POLLIN],1,0)             = 0 (Timeout)
brk(0x561386281000)                        = 0x561386281000
write(6,"",8)                              = 8
write(8,"",8)                              = 8
fstat(1,st_mode=S_IFCHR|0620,st_rdev=makedev(136,0),...) = 0
write(1,"Enter client id:",17Enter client id:
)      = 17
fstat(0,st_mode=S_IFCHR|0620,st_rdev=makedev(136,0),...) = 0
read(0,1
"1",1024)                                = 2
poll([fd=8,events=POLLIN],1,0)             = 1 ([fd=8,events=POLLIN])
read(8,"",8)                              = 8
poll([fd=8,events=POLLIN],1,0)             = 0 (Timeout)
write(6,"",8)                              = 8
poll([fd=8,events=POLLIN],1,-1)            = 1 ([fd=8,events=POLLIN])
read(8,"",8)                              = 8
poll([fd=8,events=POLLIN],1,0)             = 0 (Timeout)
poll([fd=8,events=POLLIN],1,-1)            = 1 ([fd=8,events=POLLIN])
read(8,"",8)                              = 8
poll([fd=8,events=POLLIN],1,0)             = 0 (Timeout)
write(6,"",8)                              = 8
write(1,"",1
)      = 1
write(1,"Welcome~",9Welcome~
)      = 9

```

```

...
//server
...
read(3,"# Internet (IP) protocols## Up"... ,4096) = 2932
read(3,"",4096) = 0
close(3) = 0
eventfd2(0,EFD_CLOEXEC) = 3
fcntl(3,F_GETFL) = 0x2 (flags O_RDWR)
fcntl(3,F_SETFL,O_RDWR|O_NONBLOCK) = 0
fcntl(3,F_GETFL) = 0x802 (flags O_RDWR|O_NONBLOCK)
fcntl(3,F_SETFL,O_RDWR|O_NONBLOCK) = 0
getrandom("196c5a26351326b98929a21b72",16,0) = 16
getrandom("4b4e7b6672e5b0b13487f",16,0) = 16
eventfd2(0,EFD_CLOEXEC) = 4
fcntl(4,F_GETFL) = 0x2 (flags O_RDWR)
fcntl(4,F_SETFL,O_RDWR|O_NONBLOCK) = 0
fcntl(4,F_GETFL) = 0x802 (flags O_RDWR|O_NONBLOCK)
fcntl(4,F_SETFL,O_RDWR|O_NONBLOCK) = 0
epoll_create1(E POLL_CLOEXEC) = 5
epoll_ctl(5,E POLL_CTL_ADD,4,0,u32=1554872560,u64=93928194668784) = 0
epoll_ctl(5,E POLL_CTL_MOD,4,E POLLIN,u32=1554872560,u64=93928194668784) = 0
mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0x7f83bde41000
mprotect(0x7f83bde42000,8388608,PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f83be640b70,flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_NEWNS)
= 14899
futex(0x7f83bf19007c,FUTEX_WAKE_PRIVATE,2147483647) = 0
futex(0x7f83bf190088,FUTEX_WAKE_PRIVATE,2147483647) = 0
openat(AT_FDCWD,"/proc/self/task/14899/comm",O_RDWR) = 6
write(6,"ZMQbg/0",7) = 7
close(6) = 0
eventfd2(0,EFD_CLOEXEC) = 6
fcntl(6,F_GETFL) = 0x2 (flags O_RDWR)
fcntl(6,F_SETFL,O_RDWR|O_NONBLOCK) = 0
fcntl(6,F_GETFL) = 0x802 (flags O_RDWR|O_NONBLOCK)
fcntl(6,F_SETFL,O_RDWR|O_NONBLOCK) = 0
epoll_create1(E POLL_CLOEXEC) = 7
epoll_ctl(7,E POLL_CTL_ADD,6,0,u32=1554888000,u64=93928194684224) = 0
epoll_ctl(7,E POLL_CTL_MOD,6,E POLLIN,u32=1554888000,u64=93928194684224) = 0
mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0) = 0x7f83bd640000
mprotect(0x7f83bd641000,8388608,PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f83bde3fb70,flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_NEWNS)

```

```

= 14900
openat(AT_FDCWD, "/proc/self/task/14900/comm", O_RDONLY) = 8
write(8, "ZMQbg/1", 7) = 7
close(8) = 0
eventfd2(0, EFD_CLOEXEC) = 8
fcntl(8, F_GETFL) = 0x2 (flags O_RDONLY)
fcntl(8, F_SETFL, O_RDONLY|O_NONBLOCK) = 0
fcntl(8, F_GETFL) = 0x802 (flags O_RDONLY|O_NONBLOCK)
fcntl(8, F_SETFL, O_RDONLY|O_NONBLOCK) = 0
openat(AT_FDCWD, "4040", O_RDONLY) = 9
fstat(9, {st_mode=S_IFREG|0644, st_size=25, ...}) = 0
read(9, "F", 4096) = 25
read(9, "", 4096) = 0
poll([fd=8, events=POLLIN], 1, 0) = 0 (Timeout)
socket(AF_INET, SOCK_STREAM|SOCK_CLOEXEC, IPPROTO_TCP) = 10
setsockopt(10, SOL_SOCKET, SO_REUSEADDR, [1], 4) = 0
bind(10, {sa_family=AF_INET, sin_port=htons(4040), sin_addr=inet_addr("0.0.0.0")}, 16)
= 0
listen(10, 100) = 0
getsockname(10, {sa_family=AF_INET, sin_port=htons(4040), sin_addr=inet_addr("0.0.0.0")}, [
= 0
write(6, "", 8) = 8
write(8, "", 8) = 8
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 1), ...}) = 0
write(1, "Server initialized", 19)Server initialized
) = 19
poll([fd=8, events=POLLIN], 1, -1) = 1 ([fd=8, revents=POLLIN])
read(8, "", 8) = 8
poll([fd=8, events=POLLIN], 1, 0) = 0 (Timeout)
poll([fd=8, events=POLLIN], 1, -1) = 1 ([fd=8, revents=POLLIN])
read(8, "", 8) = 8
poll([fd=8, events=POLLIN], 1, 0) = 0 (Timeout)
poll([fd=8, events=POLLIN], 1, -1) = 1 ([fd=8, revents=POLLIN])
read(8, "", 8) = 8
write(6, "", 8) = 8
poll([fd=8, events=POLLIN], 1, 0) = 0 (Timeout)
poll([fd=8, events=POLLIN], 1, -1) = 1 ([fd=8, revents=POLLIN])
read(8, "", 8) = 8
poll([fd=8, events=POLLIN], 1, 0) = 0 (Timeout)
write(6, "", 8) = 8
write(1, "Recieved message from 1 action: "..., 35)Recieved message from 1 action:

```

```
0
) = 35
write(1,"Send answer to client: [OK]",28Send answer to client: [OK]
) = 28
...
```



### 3 Выводы

Выполнив первую лабораторную работу по курсу «Операционные системы», я приобрёл практические навыки диагностики работы программного обеспечения с помощью утилиты `strace`, узнал много новых системных вызовов. Понял, что любую готовую программу можно с помощью `strace` посмотреть все её обращения к ядру по порядку.

## Список литературы

- [1] *strace*  
URL: <https://strace.io/>
- [2] *Проект OpenNet MAN() FreeBSD и Linux*  
URL: <https://www.opennet.ru/man2.shtml>