

**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

**Задача:** Написать собственную программу, которая демонстрирует работу с различными системными вызовами (8-15) операционной системы. Произвести диагностику работы написанной программы с помощью утилит ОС, изучив основные принципы применения используемых утилит.

Произвести диагностику стороннего ПО. Выявить ключевые особенности работы. Выявить предполагаемые ключевые системные вызовы, которые используются в стороннем программном обеспечении.

# 1 Описание

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

В программе были использованы следующие системные вызовы:

int access(const char *pathname, int mode)	Проверяет, существует ли файл, или имеет ли процесс права на чтение или запись. Возвращает 0 если все проверки прошли успешно, -1 в противном случае
int open(const char *pathname, int flags, mode_t mode)	Открыть или создать файл. Возвращается 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
lseek(int fildes, off_t offset, int whence)	Установить смещение для позиционирования операций чтения/записи
int fstat(int fildes, struct stat *buf)	Получить информацию о заданном файле: размер и прочее
int fchmod(int fildes, mode_t mode)	Изменить права доступа к файлу
time_t time(time_t *t)	Возвращает время в секундах, прошедшее с начала этой эпохи
int close(int fd);	Закрыть файловый дескриптор

## 2 Исходный код

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 #include <sys/stat.h>
6 #include <fcntl.h>
7 #include <time.h>
8
9 int main(int argc, char* argv[]) {
10
11     int ret;
12     const char pathPython[] = "hello.py";
13     const char list[] = "list.bin";
14
15     //is it exists?
16     ret = access(pathPython, F_OK);
17     if (ret == 0)
18         printf("access says: %s exists\n", pathPython);
19     else
20         printf("access says: %s not exists or can not be reached\n",
21             pathPython);
22
23     //open file
24     int fd = open(list, O_RDWR | O_CREAT);
25     if (fd < 0)
26         printf("Cannot open file");
27     else
28         printf("file '%s' opened with file descriptor %d\n", list, fd);
29
30     const int value = 109;
31     ret = write(fd, &value, sizeof(value));
32     printf("write wrote %d bytes to fd:%d and this is int:%d\n", ret, fd,
33         value);
34
35     off_t shift = 0;
36     ret = lseek(fd, shift, SEEK_SET);
37     printf("lseek set fd:%d offset to %d from begging of file\n", fd, ret);
38     int input;
39     ret = read(fd, &input, sizeof(input));
40     printf("read read %d bytes from fd:%d and this is int:%d\n", ret, fd,
41         input);
42
43     struct stat info;
44     //read info about file by fd
45     //stat does same but file described with char*
46     fstat(fd, &info);
47     printf(
```

```

48         "some info about fd:%d\nsize:%ld B\nconsists from %ld blocks of 512 B\
49         nblocksize of system is %ld B\n",
50         fd, info.st_size, info.st_blocks, info.st_blksize);
51     fchmod(fd, S_IWUSR | S_IRUSR | S_IRGRP | S_IROTH);
52     //user - rw
53     //group, others - r
54     ret = close(fd);
55     if (ret < 0) {
56         printf("file with fd:%d can not be closed\n", fd);
57         return -1;
58     }
59
60     time_t t = time(0);
61
62     printf("Hours past from 1970 : %ld\n", t / (3600 * 24));
63
64
65     return 0;
66 }

```

### 3 Тесты

```
art@mars:~/study/semester_3/OS/lab_1$ ./main
access says: hello.py exists
file 'list.bin'opened with file descriptor 3
write wrote 4 bytes to fd:3 and this is int:109
lseek set fd:3 offset to 0 from begging of file
read read 4 bytes from fd:3 and this is int:109
some info about fd:3
size:4 B
consists from 8 blocks of 512 B
blocksize of system is 4096 B
Hours past from 1970 : 17838
```

## 4 Диагностика strace

```
art@mars:~/study/semester_3/OS/lab_1$ strace ./main
```

Запуск программы. Первые аргументы это её название, затем идут переменные окружения

```
execve("./a.out", ["/a.out"], [/* 67 vars */]) = 0
```

```
brk(NULL) = 0x55d5b663a000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f166dd13000
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=123319, ...) = 0
mmap(NULL, 123319, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f166dcf4000
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>402"... , 832) = 832
fstat(3, st_mode=S_IFREG|0755, st_size=1960656, ...) = 0
mmap(NULL, 4061792, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f166d70f000
mprotect(0x7f166d8e5000, 2097152, PROT_NONE) = 0
mmap(0x7f166dae5000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0) = 0x7f166dae5000
mmap(0x7f166daeb000, 14944, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f166daeb000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f166dcf1000
arch_prctl(ARCH_SET_FS, 0x7f166dcf1740) = 0
mprotect(0x7f166dae5000, 16384, PROT_READ) = 0
mprotect(0x55d5b50cc000, 4096, PROT_READ) = 0
mprotect(0x7f166dd16000, 4096, PROT_READ) = 0
munmap(0x7f166dcf4000, 123319) = 0
```

Проверка существования файла

```
access("decision.py", F_OK) = 0

fstat(1, st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...) = 0
brk(NULL) = 0x55d5b663a000
brk(0x55d5b665b000) = 0x55d5b665b000
```

Вывод в stdout

```
write(1,"access says: decision.py exists",32access says: decision.py exists
) = 32
```

Открытие вернуло дескриптор 3

```
openat(AT_FDCWD,"list.bin",O_RDWR|O_CREAT,0134200) = 3
write(1,"file 'list.bin'opened with file"...46file 'list.bin'opened with file
descriptor 3
) = 46
```

Запись в файл 4 байт прошло успешно

```
write(3,"m",4) = 4
write(1,"write wrote 4 bytes to fd:3 and "...48write wrote 4 bytes to fd:3
and this is int:109
) = 48
```

Перемещение в начало файла

```
lseek(3,0,SEEK_SET) = 0
```

```
write(1,"lseek set fd:3 offset to 0 from "...48lseek set fd:3 offset to 0
from begging of file
) = 48
```

Чтение из файла 4 байт

```
read(3,"m",4) = 4
write(1,"read read 4 bytes from fd:3 and "...48read read 4 bytes from fd:3
and this is int:109
) = 48
```

Получение информации о файле

```
fstat(3,st_mode=S_IFREG|0644,st_size=4,...) = 0
write(1,"some info about fd:3",21some info about fd:3
) = 21
write(1,"size:4 B",9size:4 B
) = 9
```



```
write(1,"consists from 8 blocks of 512 B",32consists from 8 blocks of 512 B
) = 32
write(1,"blocksize of system is 4096 B",30blocksize of system is 4096 B
) = 30
```

Смена прав доступа к файлу

```
fchmod(3,0644) = 0
```

Закрытие файла

```
close(3) = 0
write(1,"Hours past from 1970 : 17490",28Hours past from 1970 : 17490) = 28
exit_group(0) = ?
+++ exited with 0 +++
```

## 5 Диагностика стороннего ПО

Диагностика работы утилиты gzip

Запуск /bin/gzip с параметрами -k untitled.js

```
execve("/bin/gzip",["gzip","-k","untitled.js"],[/* 62 vars */]) = 0
brk(NULL)                               = 0x1bb1000
```

Проверка существования неких файлов не завершается успехом

```
access("/etc/ld.so.nohwcap",F_OK)        = -1 ENOENT (No such file or directory)
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7fd434835000
access("/etc/ld.so.preload",R_OK)        = -1 ENOENT (No such file or directory)
```

Открытие на чтение файла,вероятно кэша

```
open("/etc/ld.so.cache",O_RDONLY|O_CLOEXEC) = 3
```

Получение информации о нём

```
fstat(3,st_mode=S_IFREG|0644,st_size=151570,...) = 0
mmap(NULL,151570,PROT_READ,MAP_PRIVATE,3,0) = 0x7fd43480f000
close(3)                                     = 0
access("/etc/ld.so.nohwcap",F_OK)        = -1 ENOENT (No such file or directory)
```

Открытие библиотеки

```
open("/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY|O_CLOEXEC) = 3
read(3,"77ELF>P "...,832) = 832
fstat(3,st_mode=S_IFREG|0755,st_size=1868984,...) = 0
mmap(NULL,3971488,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7fd434248000
mprotect(0x7fd434408000,2097152,PROT_NONE) = 0
mmap(0x7fd434608000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0) = 0x7fd434608000
mmap(0x7fd43460e000,14752,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1,0) = 0x7fd43460e000
close(3)                                     = 0
mmap(NULL,4096,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7fd43480e000
mmap(NULL,4096,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7fd43480d000
mmap(NULL,4096,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7fd43480c000
```

Установка состояния потока специфичного для архитектуры

```
arch_prctl(ARCH_SET_FS,0x7fd43480d700) = 0
mprotect(0x7fd434608000,16384,PROT_READ) = 0
mprotect(0x616000,4096,PROT_READ) = 0
mprotect(0x7fd434837000,4096,PROT_READ) = 0
munmap(0x7fd43480f000,151570) = 0
rt_sigaction(SIGINT,NULL,SIG_DFL,[],0,8) = 0
rt_sigaction(SIGHUP,NULL,SIG_DFL,[],0,8) = 0
rt_sigaction(SIGPIPE,NULL,SIG_DFL,[],0,8) = 0
rt_sigaction(SIGTERM,NULL,SIG_DFL,[],0,8) = 0
rt_sigaction(SIGXCPU,NULL,SIG_DFL,[],0,8) = 0
rt_sigaction(SIGXFSZ,NULL,SIG_DFL,[],0,8) = 0
rt_sigaction(SIGINT,0x4035c0,[HUP INT PIPE TERM XCPU XFSZ],SA_RESTORER,0x7fd43427d4b0
= 0
rt_sigaction(SIGHUP,0x4035c0,[HUP INT PIPE TERM XCPU XFSZ],SA_RESTORER,0x7fd43427d4b0
= 0
rt_sigaction(SIGPIPE,0x4035c0,[HUP INT PIPE TERM XCPU XFSZ],SA_RESTORER,0x7fd43427d4b0
= 0
rt_sigaction(SIGTERM,0x4035c0,[HUP INT PIPE TERM XCPU XFSZ],SA_RESTORER,0x7fd43427d4b0
= 0
rt_sigaction(SIGXCPU,0x4035c0,[HUP INT PIPE TERM XCPU XFSZ],SA_RESTORER,0x7fd43427d4b0
= 0
rt_sigaction(SIGXFSZ,0x4035c0,[HUP INT PIPE TERM XCPU XFSZ],SA_RESTORER,0x7fd43427d4b0
= 0
```

Открытие входного файла

```
open("untitled.js",O_RDONLY|O_NOCTTY|O_NONBLOCK|O_NOFOLLOW) = 3
```

Получение информации о нем

```
fstat(3,st_mode=S_IFREG|0644,st_size=2658,...) = 0
rt_sigprocmask(SIG_BLOCK,[HUP INT PIPE TERM XCPU XFSZ],[],8) = 0
```

Создание выходного сжатого файла

```
open("untitled.js.gz",O_WRONLY|O_CREAT|O_EXCL,0600) = 4
rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
```

Чтение 2658 байт из файла (попытка считать 65536)

```
read(3,"var series_url = /manga/" + ser"...",65536) = 2658
read(3,"",62878) = 0
```

Вывод сжатых данных в файл

```
write(4,"71300476265Yuntitled.js25VQo3200~5757"...",818) = 818
```

Заккрытие входного файла

```
close(3) = 0
```

Установка времени последнего доступа/ создания с точностью до наносекунд

```
utimensat(4,NULL,[1509290727,336919733,1509290727,336919733],0) = 0
```

Установка владельца

```
fchown(4,1000,1000) = 0
```

Установка прав доступа

```
fchmod(4,0644) = 0
close(4) = 0
lseek(0,0,SEEK_CUR) = -1 ESPIPE (Illegal seek)
close(0) = 0
close(1) = 0
close(2) = 0
```

Заккрытие всех открытых файлов и завершение

```
exit_group(0) = ?
+++ exited with 0 +++
```

Из данных результатов можно понять какая библиотека подключается при архивировании и в каких местах лежат кэшированные данные, и в случае некорректного запуска gzip понять что пошло не так. Тем не менее получить алгоритм сжатия исходя из системных вызовов не представляется возможным.

## 6 Выводы

Выполнив первую лабораторную работу по курсу «Операционные системы», я приобрёл практические навыки диагностики работы программного обеспечения как собственного, так истороннего, изучил системные вызовы и утилиты диагностики.

## Список литературы

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