

МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности
Кафедра математического обеспечения и применения ЭВМ

Отчет
По лабораторной работе №6
на тему: **«Работа в ОС MINIX»**

по курсу
**«Операционные системы и системное программное
обеспечение»**
Вариант №2

Выполнил:
студент гр. КТ6о2-8
Нестеренко П.А.

Проверил:
Ассистент каф. МОП ЭВМ
Альминене Т. А.

1 Цель работы

Выработка навыков работы в операционной системе MINIX 3 с использованием эмулятора VirtualBox, а также овладение техникой модификации систем с открытым исходным кодом

2 Задание

В ходе выполнения лабораторной работы необходимо:

1. Установить программу для создания виртуальных машин – «Oracle VM VirtualBox»
2. В «VirtualBox» скачать и установить образ системы «MINIX» версии «R3.2.1-972156d»
3. Модифицировать команду `cat` так, чтобы при указании ключа `H` она перед выводом каждого файла выдавала его имя.

3 Ход работы

В ходе работы было успешно установлено средство для создания виртуальных машин «Oracle VM VirtualBox» (рисунки 1). После установки «VirtualBox» была успешно сконфигурирована виртуальная машина, в которую затем была установлена ОС MINIX (рисунки 2).

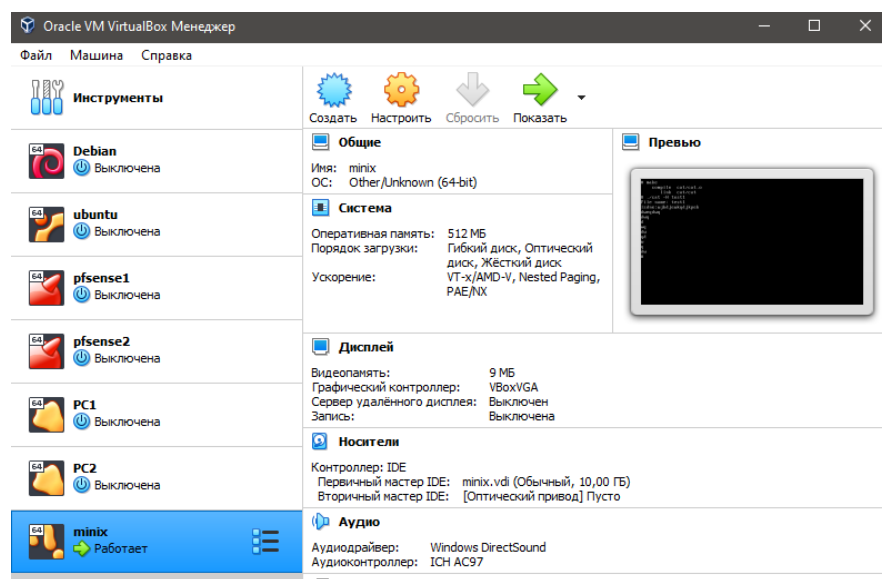
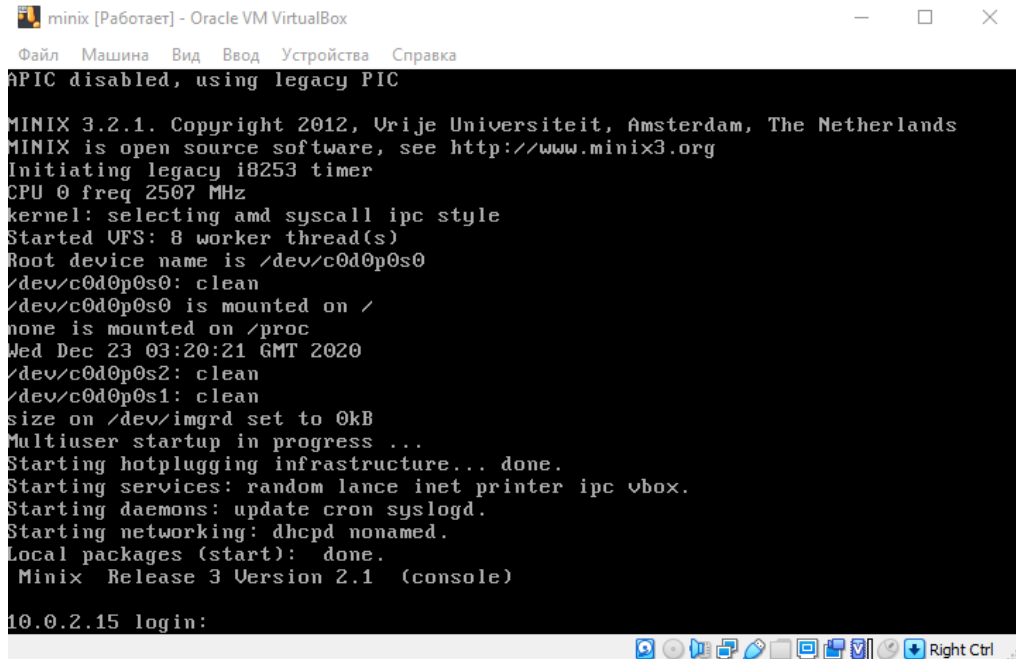


Рисунок 1 – Oracle VM VirtualBox

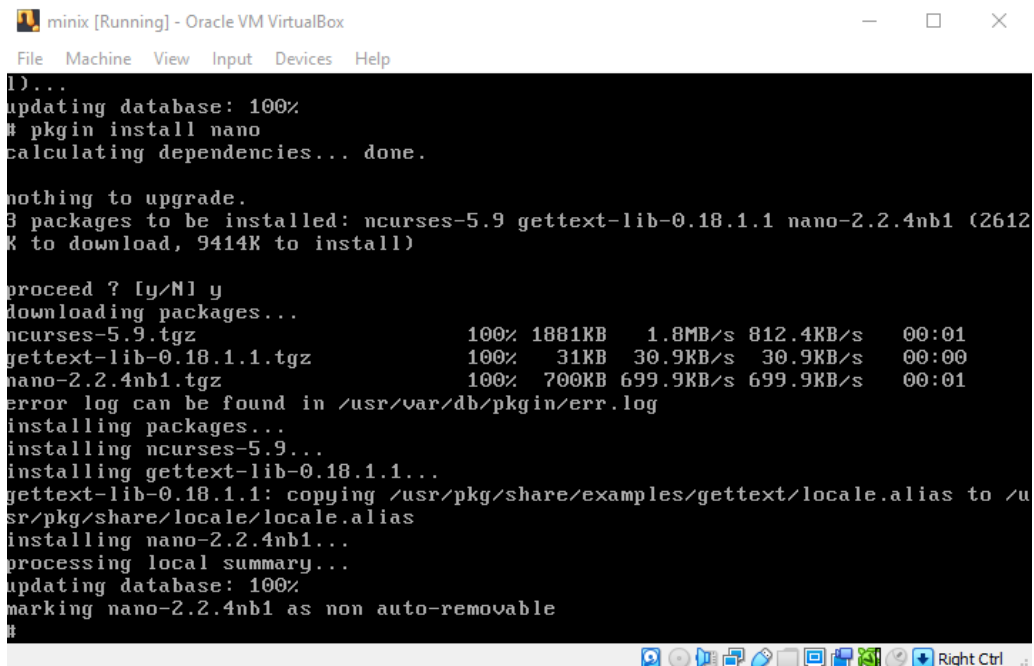


```
minix [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
APIC disabled, using legacy PIC

MINIX 3.2.1. Copyright 2012, Urije Universiteit, Amsterdam, The Netherlands
MINIX is open source software, see http://www.minix3.org
Initiating legacy i8253 timer
CPU 0 freq 2507 MHz
kernel: selecting amd syscall ipc style
Started UFS: 8 worker thread(s)
Root device name is /dev/c0d0p0s0
/dev/c0d0p0s0: clean
/dev/c0d0p0s0 is mounted on /
none is mounted on /proc
Wed Dec 23 03:20:21 GMT 2020
/dev/c0d0p0s2: clean
/dev/c0d0p0s1: clean
size on /dev/imgrd set to 0kB
Multiuser startup in progress ...
Starting hotplugging infrastructure... done.
Starting services: random lance inet printer ipc vbox.
Starting daemons: update cron syslogd.
Starting networking: dhcpd nonamed.
Local packages (start): done.
Minix Release 3 Version 2.1 (console)
10.0.2.15 login:
```

Рисунок 2 – успешный запуск установленной ОС «MINIX»

Для работы с исходным кодом файла программы «echo» был установлен текстовый редактор «nano» с помощью команды «pkgin install nano» (рисунок 3).



```
minix [Running] - Oracle VM VirtualBox
File  Machine  View  Input  Devices  Help
1)...
updating database: 100%
# pkgin install nano
calculating dependencies... done.

nothing to upgrade.
3 packages to be installed: ncurses-5.9 gettext-lib-0.18.1.1 nano-2.2.4nb1 (2612
K to download, 9414K to install)

proceed ? [y/N] y
downloading packages...
ncurses-5.9.tgz                100% 1881KB   1.8MB/s  812.4KB/s   00:01
gettext-lib-0.18.1.1.tgz      100%  31KB    30.9KB/s  30.9KB/s    00:00
nano-2.2.4nb1.tgz             100%  700KB    699.9KB/s 699.9KB/s   00:01
error log can be found in /usr/var/db/pkgin/err.log
installing packages...
installing ncurses-5.9...
installing gettext-lib-0.18.1.1...
gettext-lib-0.18.1.1: copying /usr/pkg/share/examples/gettext/locale.alias to /u
sr/pkg/share/locale/locale.alias
installing nano-2.2.4nb1...
processing local summary...
updating database: 100%
marking nano-2.2.4nb1 as non auto-removable
#
```

Рисунок 3 – установка «nano»

Исходный код программы расположен по пути «/usr/src/bin/cat/cat.c». На рисунке 4 изображен код до внесенных изменений. На рисунке 5 расположен исходный код после изменений, которые приводят к решению поставленной задачи. После завершения работы с файлом, программа была скомпилирована и протестирована с помощью команд «make» – для компиляции файла с расширением «.c» в исполняемый файл, и «./cat *args*» для тестирования измененной логики работы, где «*args*» – входные параметры программы.

```
int
main(int argc, char *argv[])
{
    int ch;
    struct flock stdout_lock;

    setprogname(argv[0]);
    (void)setlocale(LC_ALL, "");

    while ((ch = getopt(argc, argv, "beflnstuv")) != -1)
        switch (ch) {
            case 'B':
                bsize = (size_t)strtol(optarg, NULL, 0);
                break;
            case 'b':
                bflag = nflag = 1; /* -b implies -n */
                break;
            case 'e':
                eflag = vflag = 1; /* -e implies -v */
                break;
            case 'f':
                fflag = 1;
                break;
            case 'l':
                lflag = 1;
                break;
            case 'n':
                nflag = 1;
                break;
            case 's':
                sflag = 1;
                break;
            case 't':
                tflag = vflag = 1; /* -t implies -v */
                break;
            case 'u':
                setbuf(stdout, NULL);
                break;
            case 'v':
                vflag = 1;
                break;
            default:
            case '?':
                (void)fprintf(stderr,
                    "Usage: %s [-beflnstuv] [-B bsize] [-] "
                    "[file ...]\n", getprogname());
                return EXIT_FAILURE;
        }
```

Рисунок 4 – исходный код файла «cat.c» (изменённый фрагмент)

```

int
main(int argc, char *argv[])
{
    int ch;
    struct flock stdout_lock;

    setprogname(argv[0]);
    (void)setlocale(LC_ALL, "");

    while ((ch = getopt(argc, argv, "Hbeflnstuv")) != -1)
        switch (ch) {
            case 'H':
                printf("File name: %s\n", argv[2]);
                break;
            case 'B':
                bsize = (size_t)strtol(optarg, NULL, 0);
                break;
            case 'b':
                bflag = nflag = 1; /* -b implies -n */
                break;
            case 'e':
                eflag = vflag = 1; /* -e implies -v */
                break;
            case 'f':
                fflag = 1;
                break;
            case 'l':
                lflag = 1;
                break;
            case 'n':
                nflag = 1;
                break;
            case 's':
                sflag = 1;
                break;
            case 't':
                tflag = vflag = 1; /* -t implies -v */
                break;
            case 'u':
                setbuf(stdout, NULL);
                break;
            case 'v':
                vflag = 1;
                break;
            default:
            case '?':
                /* ... */
        }
}

```

Рисунок 5 – измененный код «cat.c» (изменённый фрагмент)

Демонстрация измененной и скомпилированной программы представлена на рисунке 6.

```
# ls
Makefile echo.1 echo.c
# make
    compile echo/echo.o
    link echo/echo
# ./echo 123
123
# ./echo 1 2 3
1 2 3
# ./echo -r 123
123
# ./echo -r 1 2 3
3 2 1
# ./echo -r -n 1 2 3
3 2 1 -n
# ./echo -n -r 1 2 3
3 2 1# ./echo -n 1 2 3
1 2 3# ./echo -r a b c d e
e d c b a
# -
```

Рисунок 6 – демонстрация работы измененной программы

5 Выводы

При выполнении лабораторной работы я познакомился со средой для создания виртуальных машин «Oracle VM VirtualBox», научился устанавливать и настраивать ОС «MINIX 3» на виртуальной машине, а так же получил навыки изменения исходного кода системных программ и перекомпиляции UNIX-подобной операционной системы.