

ГУАП

КАФЕДРА № 44

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Доц., к.т.н.

---

должность, уч. степень, звание

---

подпись, дата

В.В.Балберин

---

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

## **Переменные окружения и файловые операции**

по курсу: **Операционные системы семейства GNU/Linux**

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4745M



---

подпись, дата

П.А. Константинов

---

инициалы, фамилия

**Санкт-Петербург 2018**

## Цели

- 1.Параметры функции `main()`.
- 2.Переменные среды и аргументы командной строки.
- 3.Семейство функций для системного вызова `exec()`.
- 4.Потоки ввода-вывода.
- 5.Файловый дескриптор.
- 6.Системный вызов `open()`.
- 7.Системные вызовы `read()`, `write()`.
- 8.Системные вызовы `close()`.

## Выполнение

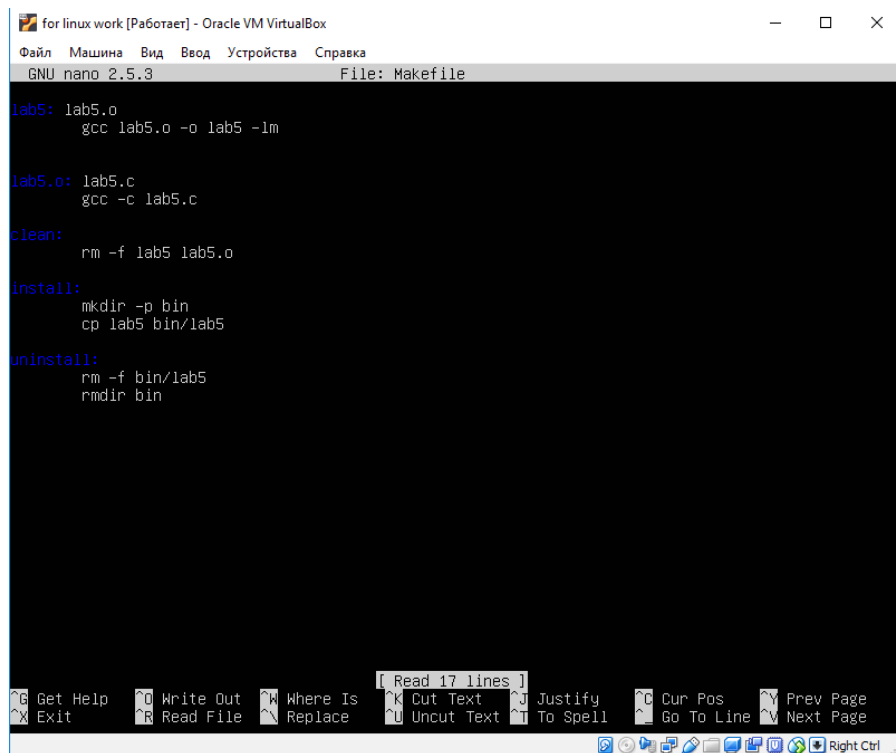
---

Данная лабораторная работа была выполнена на том же контейнере, что и предыдущая. Это было обусловлено его предыдущей настройкой под выполненные задачи.

Таким образом, мы просто запускаем наш контейнер

```
docker run -it konstantinov/abc-linux-4-1
```

Далее мы создаем Makefile по аналогии с предыдущей лабораторной:



The screenshot shows a terminal window titled "for linux work [Работает] - Oracle VM VirtualBox". Inside the terminal, the GNU nano 2.5.3 editor is open, editing a file named "Makefile". The content of the Makefile is as follows:

```
lab5: lab5.o
    gcc lab5.o -o lab5 -lm

lab5.o: lab5.c
    gcc -c lab5.c

clean:
    rm -f lab5 lab5.o

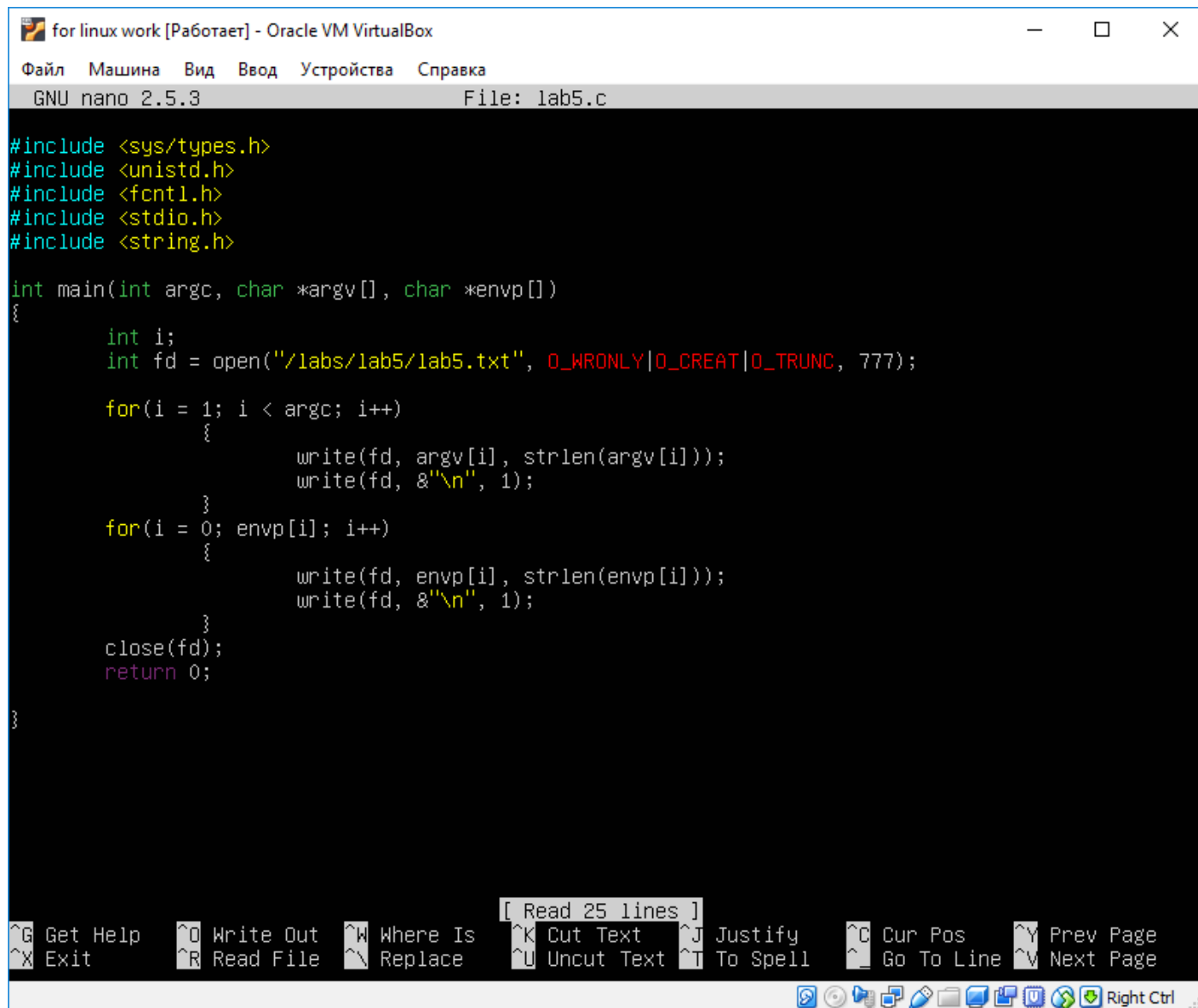
install:
    mkdir -p bin
    cp lab5 bin/lab5

uninstall:
    rm -f bin/lab5
    rmdir bin
```

The nano editor's status bar at the bottom displays various keyboard shortcuts and the current cursor position: "Read 17 lines", "Cut Text", "Justify", "Cur Pos", "Prev Page", "Exit", "Read File", "Replace", "Uncut Text", "To Spell", "Go To Line", "Next Page", and "Right Ctrl".

Makefile

По материалам методических указаний пишем наш собственный C-файл:



```
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[], char *envp[])
{
    int i;
    int fd = open("/labs/lab5/lab5.txt", O_WRONLY|O_CREAT|O_TRUNC, 777);

    for(i = 1; i < argc; i++)
    {
        write(fd, argv[i], strlen(argv[i]));
        write(fd, &"\n", 1);
    }
    for(i = 0; envp[i]; i++)
    {
        write(fd, envp[i], strlen(envp[i]));
        write(fd, &"\n", 1);
    }
    close(fd);
    return 0;
}
```

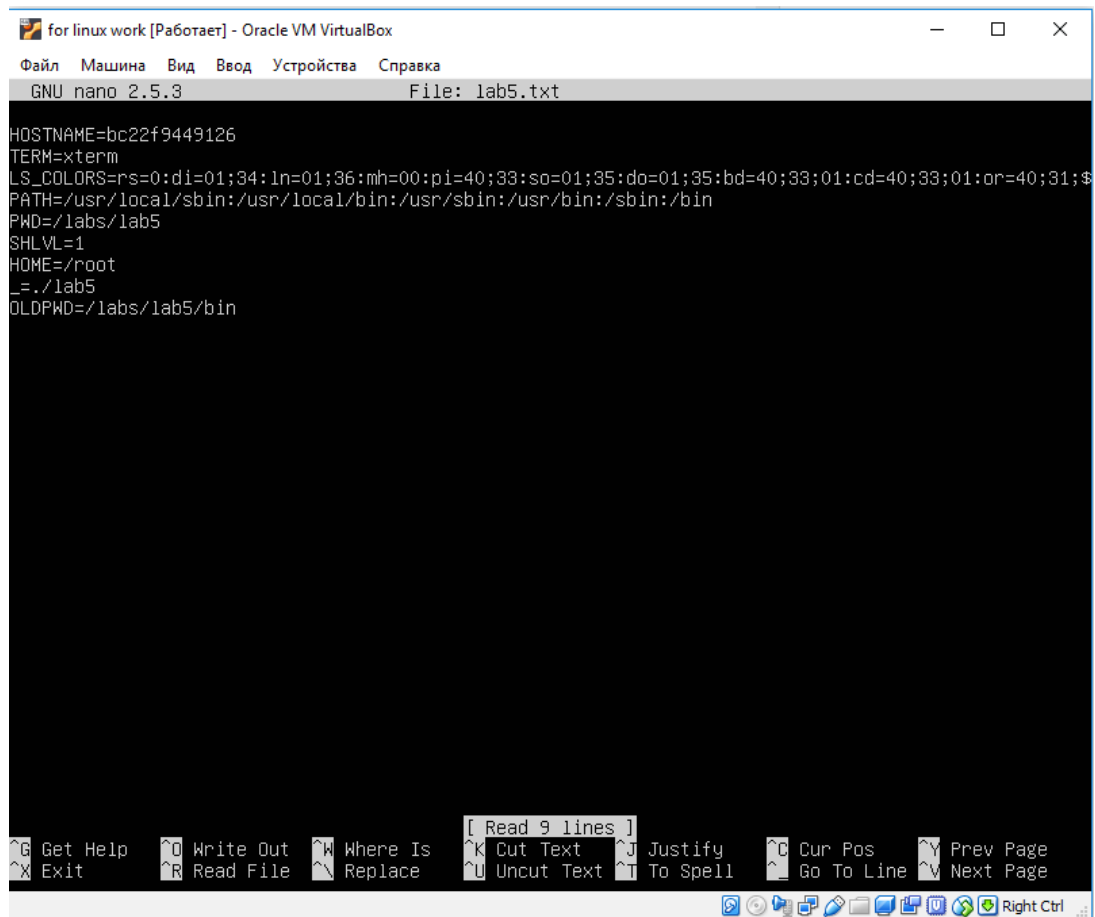
[ Read 25 lines ]

Get Help	Write Out	Where Is	Cut Text	Justify	Cur Pos	Prev Page
Exit	Read File	Replace	Uncut Text	To Spell	Go To Line	Next Page

Right Ctrl

Код C-файла

Проверяем работу без аргументов:



```
for linux work [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
GNU nano 2.5.3      File: lab5.txt

HOSTNAME=bc22f9449126
TERM=xterm
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;$
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PWD=/labs/lab5
SHLVL=1
HOME=/root
_=./lab5
OLDPWD=/labs/lab5/bin

[ Read 9 lines ]
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos   ^Y Prev Page
^X Exit      ^R Read File  ^M Replace    ^U Uncut Text ^T To Spell   ^G Go To Line ^V Next Page
Right Ctrl
```

Выходной текстовый файл

## Выводы

Были изучены методы получения переменных окружения и аргументов приложения. Также были изучены системные вызовы, связанные с файлами(open-close, read, write)