Отчёт по лабораторной работе №6

Олег Россохин

Содержание

# 1 Цель работы

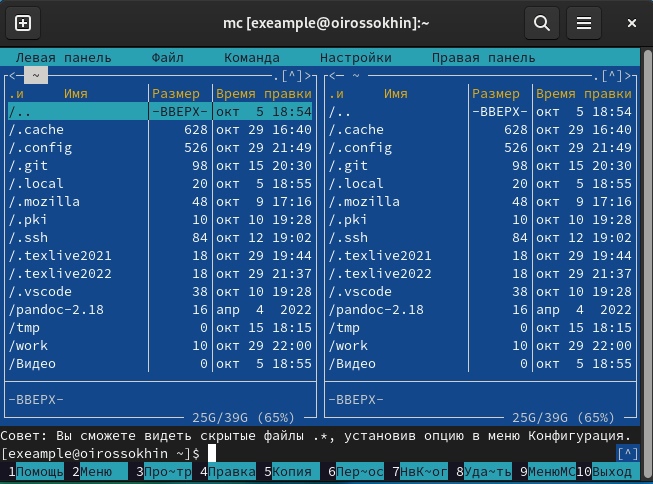
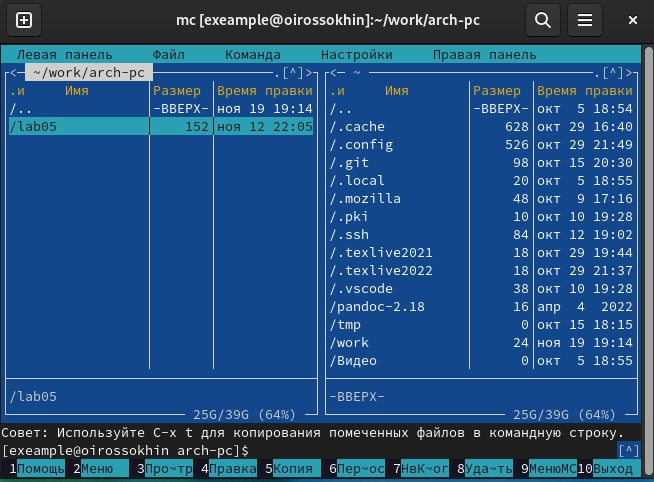
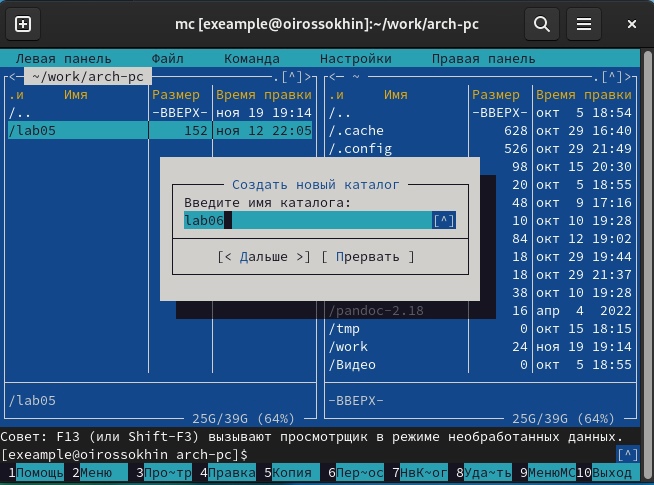
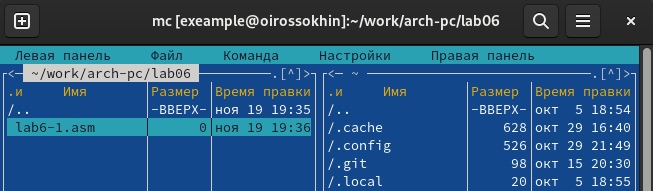
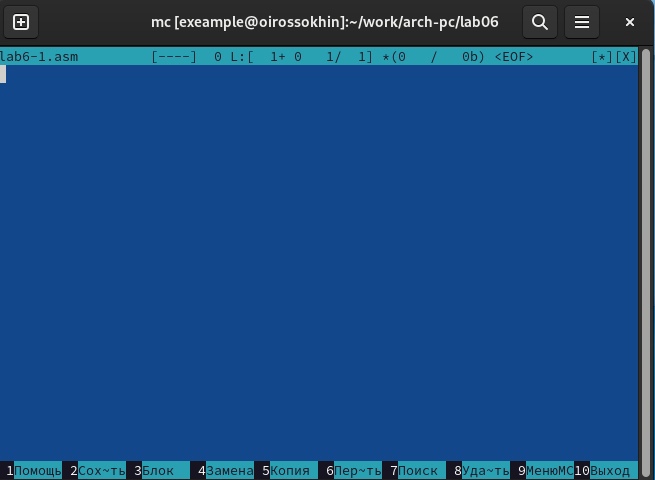
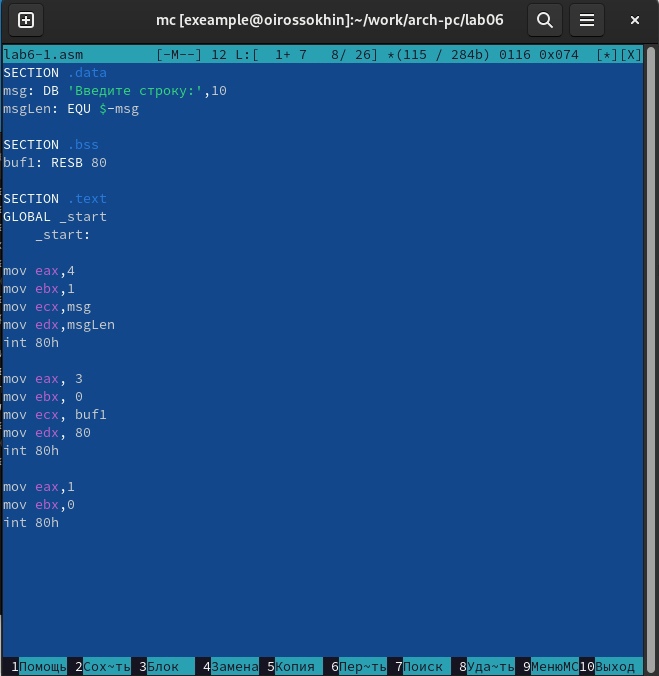
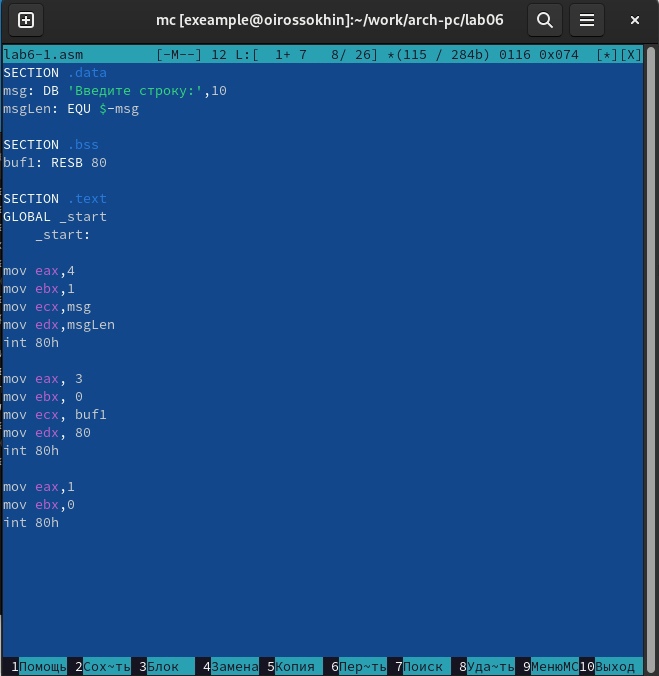
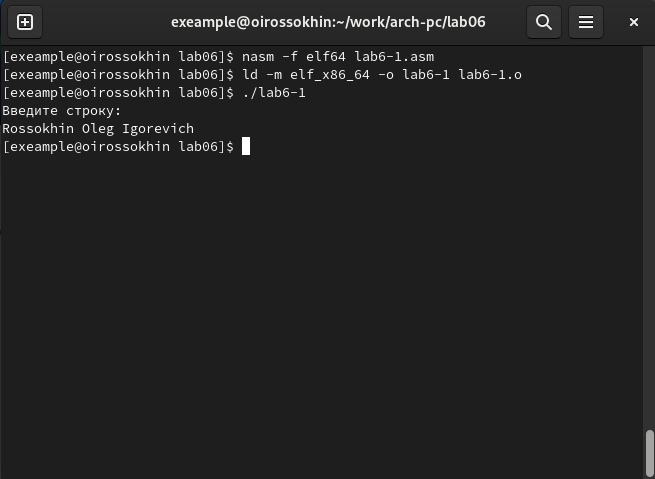
Приобретение практических навыков работы в Midnight Commander. Освоение инструкций языка ассемблера mov и int.

# 2 Теоретическое введение

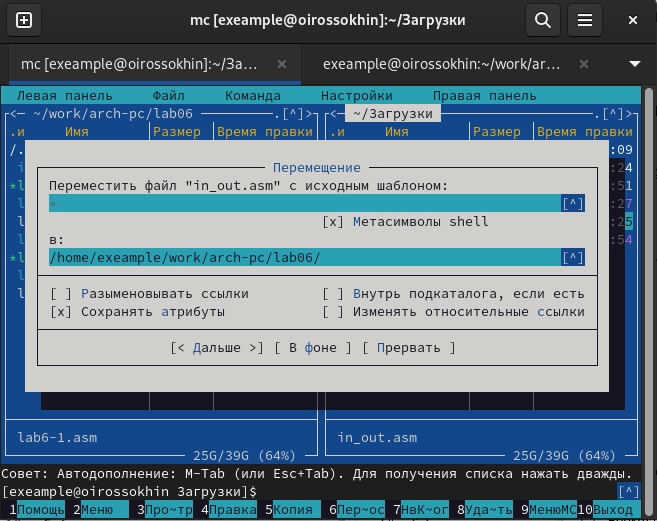
## 2.1 Основы работы с Midnight Commander

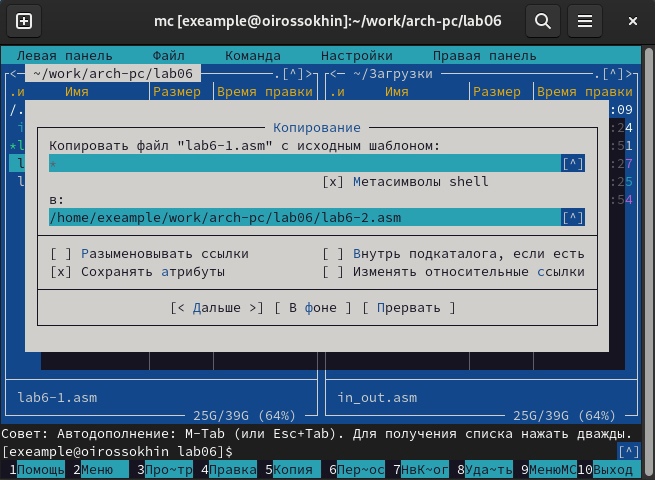
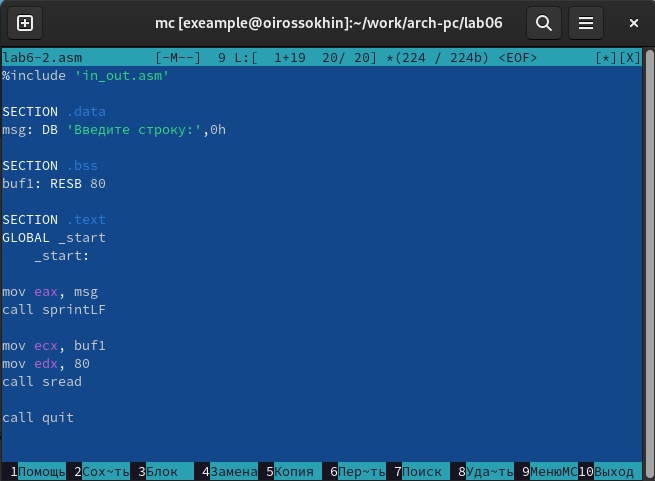
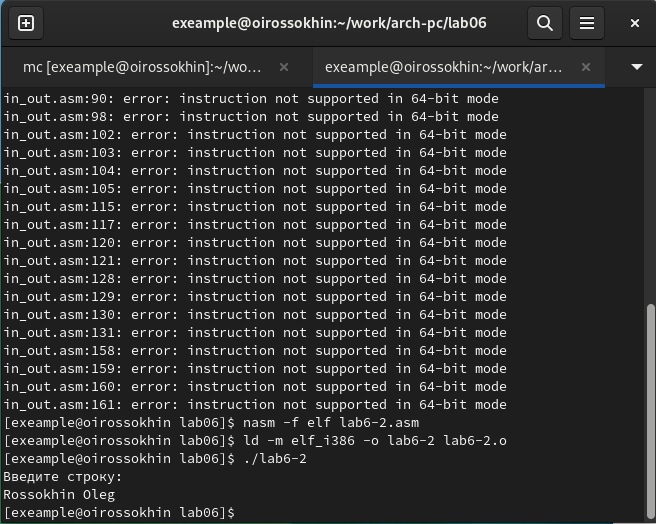
Midnight Commander — это программа, которая позволяет просматривать структуру каталогов и выполнять основные операции по управлению файловой системой, т.е. mc является файловым менеджером. Midnight Commander позволяет сделать работу с файлами более удобной и наглядной по сравнению с терминалом.

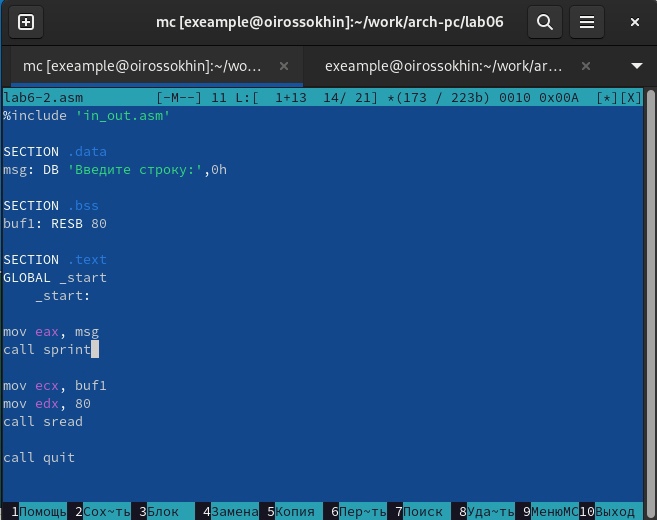
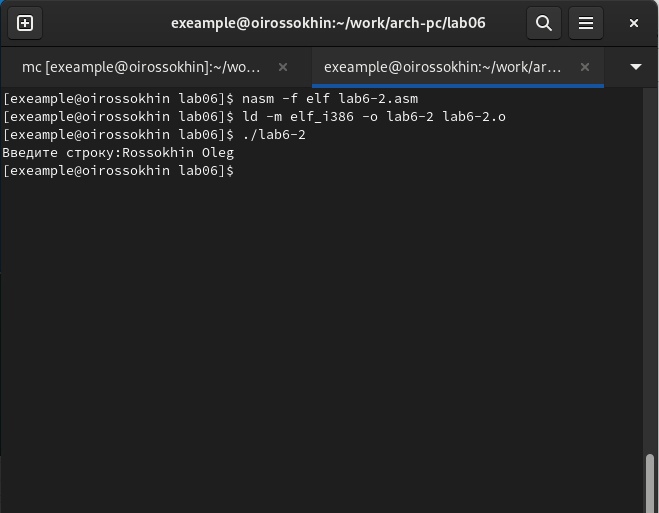
# 3 Выполнение лабораторной работы

1. Откроем Midnight Commander 
2. Пользуясь клавишами ↑ , ↓ и Enter перейдем в каталог ~/work/arch-pc, созданный при выполнении лабораторной работы №5 
3. С помощью клавиши F7 создадим папку lab06 и перейдем в созданный каталог 
4. Пользуясь строкой ввода и командой touch создадим файл lab6-1.asm 
5. С помощью клавиши F4 откроем файл lab6-1.asm для редактирования 
6. Введем текст программы вывода сообщения на экран, сохраним изменения и закроем файл 
7. С помощью клавиши F3 откроем файл lab6-1.asm для просмотра и убедимся, что файл содержит текст программы 
8. Оттранслируем текст программы lab6-1.asm в объектный файл. Далее выполним компоновку объектного файла и запустим получившийся исполняемый файл. Программа напишет ‘Введите строку:’ и будет ожидать ввода текста с клавиатуры. На запрос введу свои ФИО. 

## 3.1 Подключение внешнего файла in\_out.asm

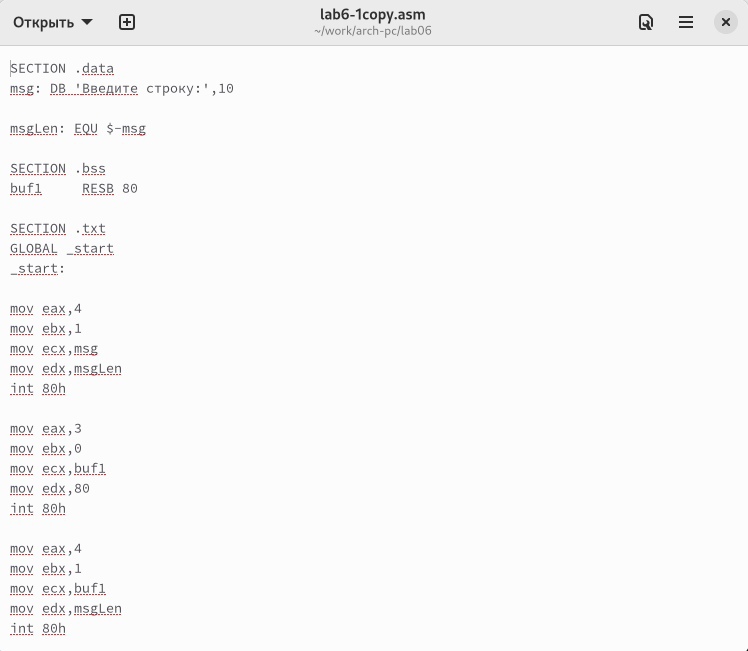
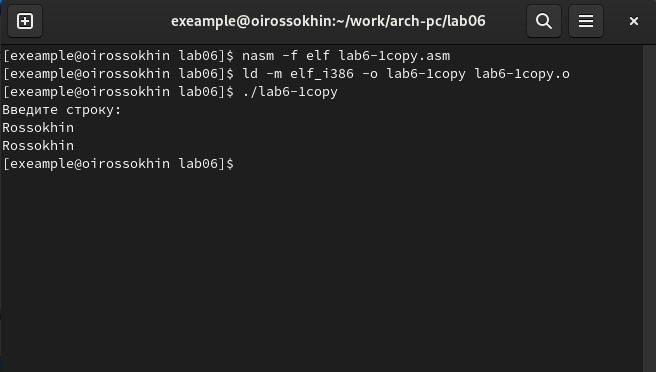
9-10. Скачаем файл in\_out.asm и с помощью MC и клавиши F6 переместим его в каталог с программой lab6-1.asm. 

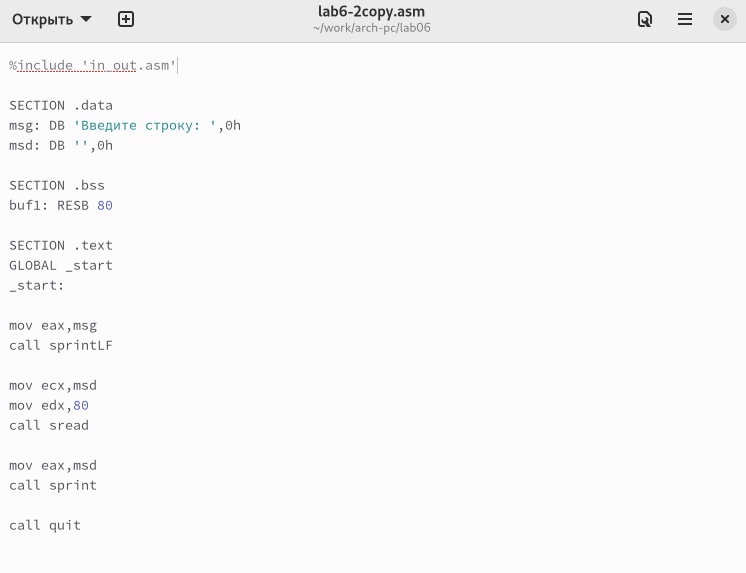
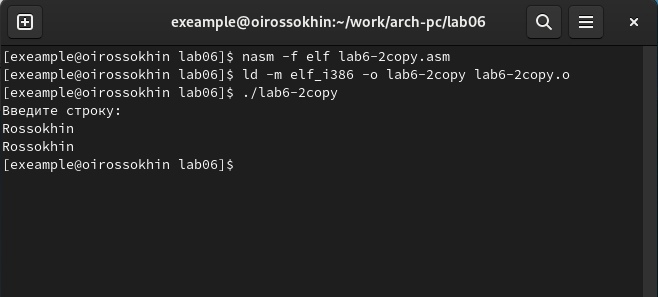
1. С помощью клавиши F5 создадим копию файла lab6-1.asm с именем lab6-2.asm. 
2. Исправим текст программы в файле lab6-2.asm с использование подпрограмм из внешнего файла in\_out.asm. Создадим исполняемый файл и проверим его работу.  

13.В файле lab6-2.asm заменим подпрограмму sprintLF на sprint. Создадим исполняемый файл и проверим его работу. *В чем разница?*  

заметим, что разница двух команд sprint и sprintLF заключается в переносе строки.

# 4 Самостоятельная работа

1-2. Создайте копию файла lab6-1.asm. Внесите изменения в программу (без использования внешнего файла in\_out.asm), так чтобы она работала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран. Затем получите исполняемый файл и проверьте его работу. На приглашение ввести строку введите свою фамилию.  

3-4. Создайте копию файла lab6-2.asm. Исправьте текст программы с исполь- зование подпрограмм из внешнего файла in\_out.asm, так чтобы она ра- ботала по следующему алгоритму: • вывести приглашение типа “Введите строку:”; • ввести строку с клавиатуры; • вывести введённую строку на экран. Создайте исполняемый файл и проверьте его работу.  

# 5 Вопросы для самопроверки

1. Каково назначение mc?
   1. Midnight Commander - программа, позволяющая просмотреть структуру каталогов и выполнить основные операции по управлению файловой системой.
2. Какие операции с файлами можно выполнить как с помощью команд bash, так и с помощью меню (комбинаций клавиш) mc? Приведите несколько примеров.
   1. Как с помощью команды cd в **bash** можно перейти в нужный каталог, так и в **mc** это можно сделать с помощью клавиш.
   2. Копирование файла осуществляется как в **bash** командой cp, так и в **mc** клавишей F5
   3. Перемещение файла в **bash** осуществляется командой mv, а в **mc** - клавишей F6.
3. Какова структура программы на языке ассемблера NASM?
   1. Программа на языке ассемблера NASM, как правило, состоит из трёх секций: SECTION .text(код программы); SECTION .data(секция инициализированных данных); SECTION .bss(секция неинициализированных данных).
4. Для описания каких данных используются секции bss и data в языке ассемблера NASM?
   1. SECTION .data содержит переменные, для которых задано начальное значение.
   2. SECTION .bss используется содержит переменные, для которых **не** задано начальное значение.
5. Для чего используются компоненты db, dw, dd, dq и dt языка ассемблера NASM?
   1. db - определяет переменную размером в 1 байт;
   2. dw - определяет переменную размеров в 2 байта(слово);
   3. dd - определяет переменную размером в 4 байта(двойное слово);
   4. dq - определяет переменную размером в 8 байт;
   5. dt - определяет переменную размером в 10 байт.
6. Какое произойдёт действие при выполнении инструкции mov eax, esi?
   1. mov **eax**, **esi** перешлет значение регистра **esi** в регистр **eax**
7. Для чего используется инструкция int 80h?
   1. Инструкция int 80h используется для выполнения системного вызова какой-либо функции ядра Linux. При этом происходит передача управления ядру операционной системы. Чтобы узнать, какую именно системную функцию нужно выполнить, ядро извлекает номер системного вызова из регистра eax. Поэтому перед вызовом прерывания необходимо поместить в этот регистр нужный номер. # Выводы

По итогам проделанной работы мы приобрели навыки работы с Midnight Commander, а также освоили некоторые инструкции языка ассемблер.

# Список литературы

::: {#refs} ::: https://esystem.rudn.ru/pluginfile.php/1584633/mod\_resource/content/1/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%966.pdf