# Front matter

lang: ru-RU title: Программирование в UNIX Продолжение subtitle: ДР по ОС №12 author: Аникин Константин Сергеевич group: НПИбд-01-20

# Formatting

toc-title: "Содержание" toc: true # Table of contents toc\_depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4paper documentclass: scrreprt polyglossia-lang: russian polyglossia-otherlangs: english mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase indent: true pdf-engine: lualatex header-includes: - \linepenalty=10 # the penalty added to the badness of each line within a paragraph (no associated penalty node) Increasing the value makes tex try to have fewer lines in the paragraph. - \interlinepenalty=0 # value of the penalty (node) added after each line of a paragraph. - \hyphenpenalty=50 # the penalty for line breaking at an automatically inserted hyphen - \exhyphenpenalty=50 # the penalty for line breaking at an explicit hyphen - \binoppenalty=700 # the penalty for breaking a line at a binary operator - \relpenalty=500 # the penalty for breaking a line at a relation -

\clubpenalty=150 # extra penalty for breaking after first line of a paragraph - \widowpenalty=150 # extra penalty for breaking before last line of a paragraph -

\displaywidowpenalty=50 # extra penalty for breaking before last line before a display math - \brokenpenalty=100 # extra penalty for page breaking after a hyphenated line - \predisplaypenalty=10000 # penalty for breaking before a display - \postdisplaypenalty=0 # penalty for breaking after a display - \floatingpenalty = 20000 # penalty for splitting an insertion (can only be split footnote in standard LaTeX) - \raggedbottom # or \flushbottom - \usepackage{float} # keep figures where there are in the text

- \floatplacement{figure}{H} # keep figures where there are in the text

# Цели работы

Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

# Задание

Ознакомиться с теоретическим материалом. Написать четыре скрипта из задания.

# Теоретическое введение

[[1](http://esyr.org/uneex_disc/mounted/FreeCode/167/linux_shell/praktika_linux_12.htm)]

Командный файл (скрипт или сценарий) - это текстовый файл, состоящий из команд интерпретатора. При запуске этого файла последовательно выполняются все команды, содержащиеся в нем.

Как файл его можно создать командой cat или использовать текстовый редактор, например, vi.

Для тех, кто привык работать с текстовым редактором, встроенным в файловый менеджер Midnight Commander (MC - запуск), в котором имеется встроенный текстовый редактор с более привычным интерфейсом.

Краткий справочник команд Linux [[2](https://hpc.icc.ru/documentation/cmnds.pdf)]

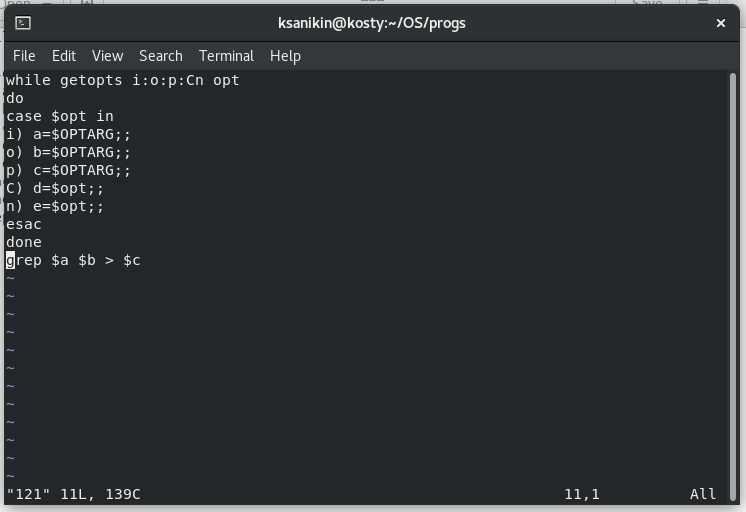
# Выполнение лабораторной работы

## Скрипт 1

Непосредственно сам скрипт. (рис. 1)

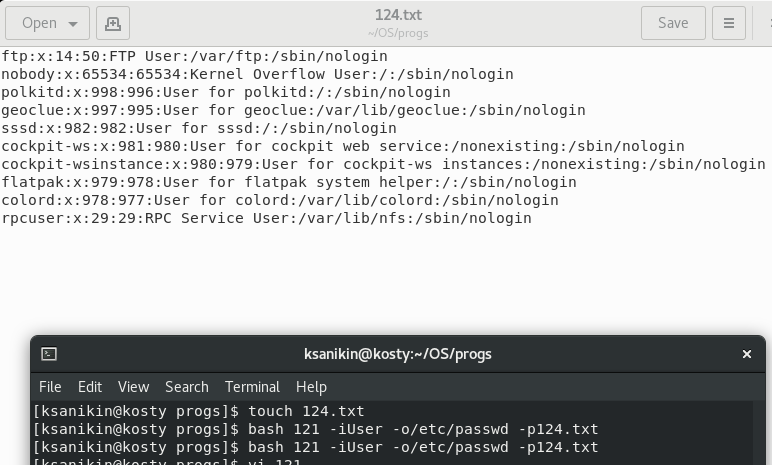
Не все ключи удалось реализовать, C и n остались за бортом.

Также ключи i o p поменяны местами. Это остатки рабочей версии, исправить которую не хватило времени. Результат команды grep записывается в файл, данный ключом p



*Рис. 1: Скрипт 1*

Результаты работы скрипта 1 (рис. 2)

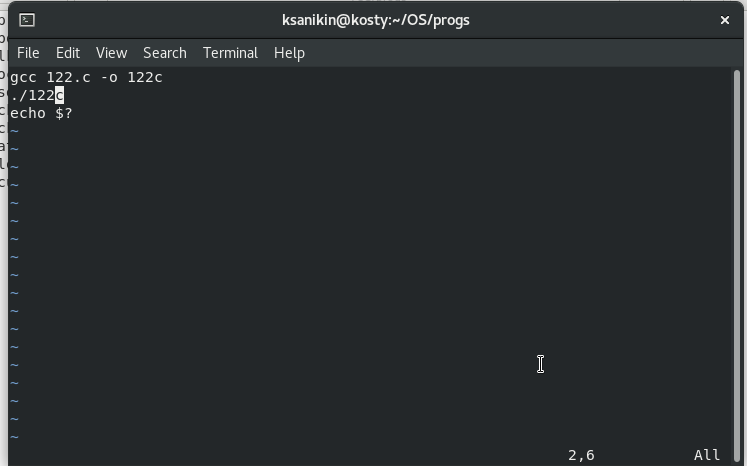


*Рис. 2: Результаты работы скрипта 1*

## Скрипт 2

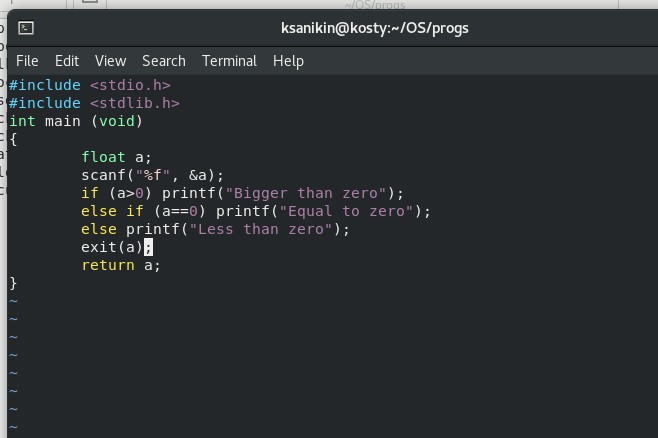
Скрипт для активации cшного файла. (рис. 3)

Используется команда gcc для превращения файла в исполняемый, после чего файл активируется



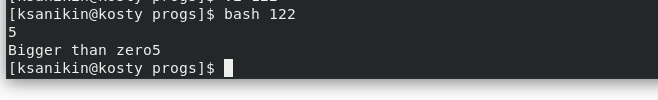
*Рис. 3: Скрипт 2*

И сам сшный файл. Не уверен, надо ли здесь использовать и return и exit (рис. 31)



*Рис. 31: Скрипт 2*

Результаты работы скрипта 2 (рис. 4)

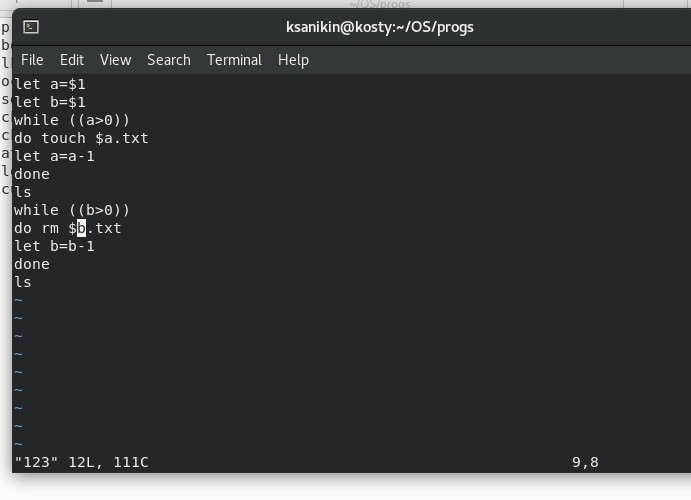


*Рис. 4: Результаты работы скрипта 2*

## Скрипт 3

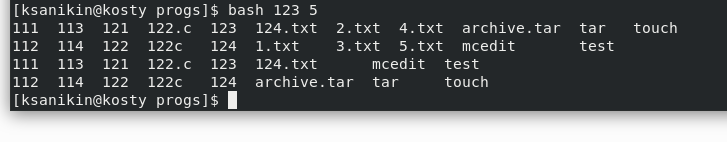
Непосредственно сам скрипт. (рис. 5)

Назначаем переменные, создаем временные файлы, показываем их присутствие ls После чего удаляем каждый временный файл и снова ls



*Рис. 5: Скрипт 3*

Результаты работы скрипта 3 (рис. 6)

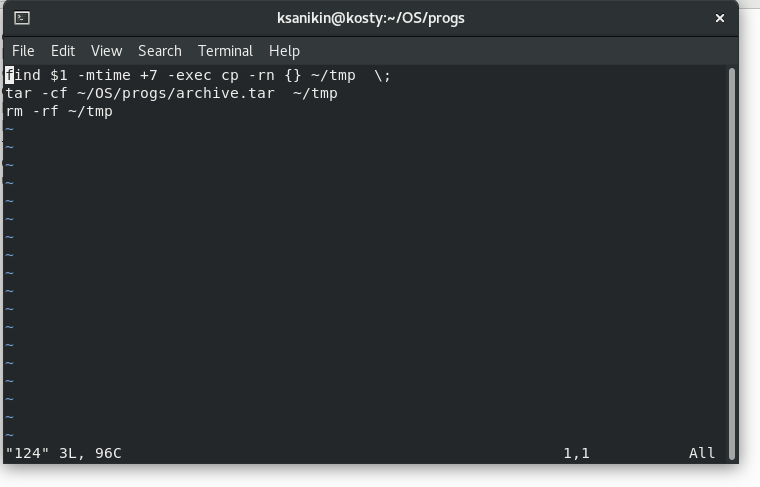


*Рис. 6: Результаты работы скрипта 3*

## Скрипт 4

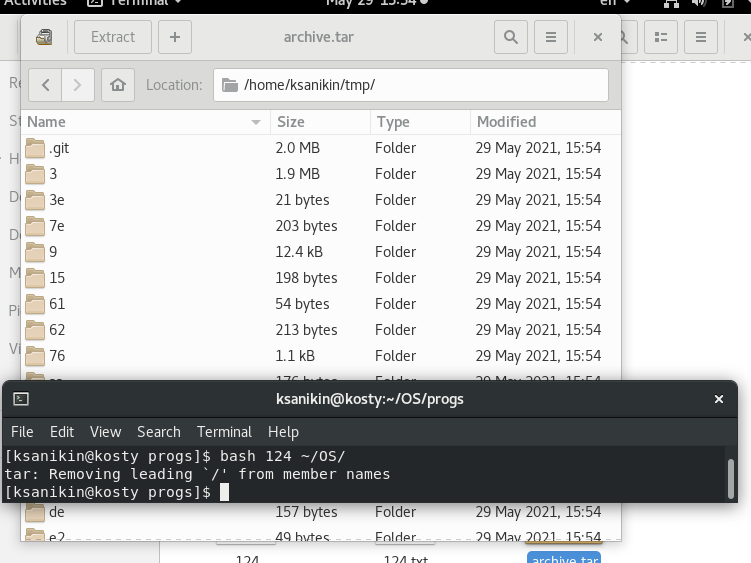
Непосредственно сам скрипт. (рис. 7)

Находим все файлы недельной давности, копируем их в папку tmp Архивируем tarом файлы из tmp в архив, после чего удаляем файлы из tmp



*Рис. 7: Скрипт 4*

Результаты работы скрипта 4 (рис. 8)



*Рис. 8: Результаты работы скрипта 4*

# Выводы

Скрипты созданы и работают. Работа выполнена полностью.

## Библиографический список

1. : [Создание командных файлов](http://esyr.org/uneex_disc/mounted/FreeCode/167/linux_shell/praktika_linux_12.htm)
2. : [Краткий справочник команд Linux](https://hpc.icc.ru/documentation/cmnds.pdf)