**Front matter**



lang: ru-RU title: "Лабораторная работа №12" subtitle: " Программирование в командном процессоре ОС UNIX.Ветвления и циклы" author: "Данилова Анастасия Сергеевна"

**Formatting**

toc-title: "Содержание" toc: true # Table of contents toc\_depth: 2 lof: true # List of figures lot: true # List of tablesfontsize: 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=MatchLowercasemonofontoptions: 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

**Цель работы**

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы сиспользованием логических управляющих конструкций и циклов.

**Задание**

1. Используя команды getopts grep, написать командный файл, который анализирует командную строку с ключами: – -iinputfile — прочитать данные из указанного файла; – -ooutputfile — вывести данные в указанный файл; – -pшаблон — указать шаблон для поиска; – -C — различать большие и малые буквы; – -n — выдавать номера строк. а затем ищет в указанном файле нужные строки, определяемые ключом -p.

2. Написать на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. Затем программа завершается с помощью функции exit(n), передавая информацию в о коде завершения в оболочку.

3. Написать командный файл, создающий указанное число файлов, пронумерованных последовательно от 1 до N (например 1.tmp, 2.tmp, 3.tmp,4.tmp и т.д.). Число файлов, которые необходимо создать, передаётся в аргументы командной строки. Этот же командный файл должен уметь удалять все созданные им файлы (если они существуют).

4. Написать командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицировать его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find).

**Теоретическое введение**



Команда **getopts** является встроенной командой командной оболочки bash, предназначенной для разборапараметров сценариев. Она обрабатывает исключительно однобуквенные параметры как с аргументами, так ибез них и этого вполне достаточно для передачи сценариям любых входных данных.

Базовый синтаксис команды выглядит следующим образом:

$ getopts строка-параметров переменная [набор-параметров]

**Выполнение лабораторной работы**

1. Напишем на языке Си программу, которая вводит число и определяет, является ли оно больше нуля, меньше нуля или равно нулю. (рис1) ![](C:\Users\nastd\Documents\рудн\4 модуль\операционные системы \Лаб12\скрины\рис1.jpg)

**рис 1**

2. Убедимся, что программа работает. Введем "4" и увидим, что четверка больше 0 (рис2) ![](C:\Users\nastd \Documents\рудн\4 модуль\операционные системы\Лаб12\скрины\рис2.jpg)

**рис 2**

3. Напишем командный файл, который с помощью команды tar запаковывает в архив все файлы в указанной директории. Модифицируем его так, чтобы запаковывались только те файлы, которые были изменены менее недели тому назад (использовать команду find). (рис 3) ![](C:\Users\nastd\Documents\рудн\4 модуль \операционные системы\Лаб12\скрины\рис3.jpg) **рис 3**

**Выводы**

Мы изучили основы программирования в оболочке ОС UNIX. Научились писать более сложные командныефайлы с использованием логических управляющих конструкций и циклов.

**Библиография**

<https://linux-faq.ru/page/komanda-getopts>

<https://losst.ru/komanda-find-v-linux>