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. - \interlinepenalty=50 # the penalty for line breaking at an automatically inserted hyphen - \exhyphenpenalty=50 # the penalty for line breaking at an explicit hyphen - \binopenalty=700 # the penalty for breaking a line at a binary operator - \relpenalty=500 # the penalty for breaking a line at a relation - \club lubenalty=150 # extra penalty for breaking after first line of a paragraph - \widowpenalty=150 # extra penalty for breaking before last line before a display math - \brokenpenalty=100 # extra penalty for page breaking after a hyphenated line - \predisplaypenalty=1000 # penalty for breaking before a display - \postdisplaypenalty=0 # penalty for 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) - \rangedbottom # 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]

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]

Выполнение лабораторной работы

Скрипт 1

Непосредственно сам скрипт. (рис. 1)

Не все ключи удалось реализовать, С и п остались за бортом.

Также ключи і о р поменяны местами. Это остатки рабочей версии, исправить которую не хватило времени.

Результат команды grep записывается в файл, данный ключом р

```
ksanikin@kosty:~/OS/progs
 2
File Edit View Search Terminal Help
while getopts i:o:p:Cn opt
do
case $opt in
i) a=$OPTARG;;
o) b=$0PTARG;;
p) c=$0PTARG;;
C) d=$opt;;
n) e=$opt;;
esac
done
grep $a $b > $c
                                                                                  All
"121" 11L, 139C
                                                                   11,1
```

Рис. 1: Скрипт 1

Результаты работы скрипта 1 (рис. 2)

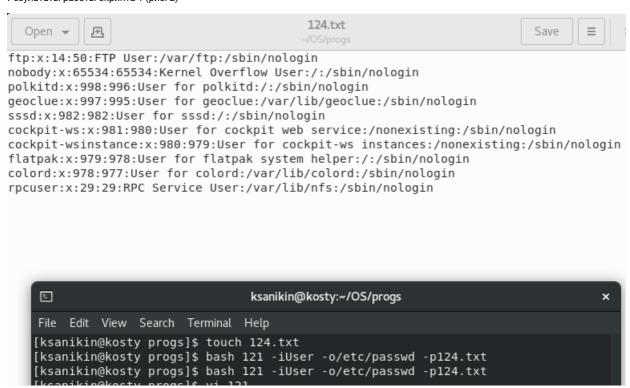


Рис. 2: Результаты работы скрипта 1

Скрипт 2

Скрипт для активации сшного файла. (рис. 3)

Используется команда дсс 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)

```
ksanikin@kosty:~/OS/progs

File Edit View Search Terminal Help

#include <stdio.h>
#include <stdlib.h>
int main (void)

{
    float a;
        scanf("%f", &a);
        if (a>0) printf("Bigger than zero");
        else if (a==0) printf("Equal to zero");
        else printf("Less than zero");
        exit(a);
        return a;
}
```

Рис. 31: Скрипт 2

Результаты работы скрипта 2 (рис. 4)

```
[ksanikin@kosty progs]$ bash 122
5
Bigger than zero5
[ksanikin@kosty progs]$
```

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

```
p
  2
                                      ksanikin@kosty:~/OS/progs
b
  File Edit View Search Terminal Help
○let a=$1
Slet b=$1
cwhile ((a>0))
c do touch $a.txt
<sup>a</sup>let a=a-1
 done
cls
 while ((b>0))
 do rm $<mark>b</mark>.txt
 let b=b-1
 done
 ls
  "123" 12L, 111C
                                                                            9,8
```

Рис. 5: Скрипт 3

Результаты работы скрипта 3 (рис. 6)

```
[ksanikin@kosty progs]$ bash 123 5
111 113 121 122.c 123 124.txt 2.txt 4.txt archive.tar tar touch
112 114 122 122c 124 1.txt 3.txt 5.txt mcedit test
111 113 121 122.c 123 124.txt mcedit test
112 114 122 122c 124 archive.tar tar touch
[ksanikin@kosty progs]$
```

Рис. 6: Результаты работы скрипта 3

Скрипт 4

Непосредственно сам скрипт. (рис. 7)

Находим все файлы недельной давности, копируем их в папку tmp

Архивируем taroм файлы из tmp в архив, после чего удаляем файлы из tmp

Рис. 7: Скрипт 4

Результаты работы скрипта 4 (рис. 8)

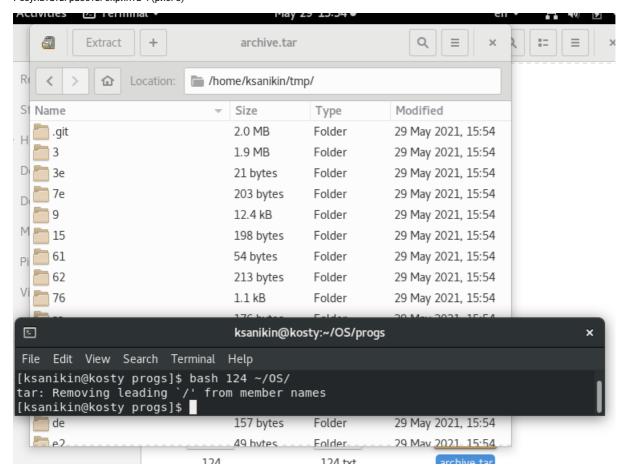


Рис. 8: Результаты работы скрипта 4

Выводы

Скрипты созданы и работают. Работа выполнена полностью.

Библиографический список

- [1]: Создание командных файлов
- [2]: Краткий справочник команд Linux