



Отчет по лабораторной работе № 21 по курсу Алгоритмы и структуры данных

Студент группы М8О-103Б-22 Клименко Виталий Максимович, № по списку 11

Контакты www, e-mail, icq, skype vitalikklimenko96@gmail.com

Работа выполнена: 1 марта 2022 г.

Преподаватель: доцент Никулин С.П.

Входной контроль знаний с оценкой _____

Отчет сдан « » _____ 202__ г., итоговая оценка ____

Подпись преподавателя _____

1. **Тема:** Программирование на интерпретируемых командных языках _____

2. **Цель работы:** Составить программу выполнения заданных действий _____

3. **Задание (вариант № 12):** Генерация заданного числа копий указанного файла. Имена копий генерировать добавлением к имени исходного файла следующей по порядку буквы или цифры, начиная с заданной буквы или цифры _____

4. **Оборудование (лабораторное):**
ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб,
НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____
Оборудование ПЭВМ студента, если использовалось:
Процессор Intel 4x 3.5GHz _____ с ОП 16 Гб _____ НМД HDD 200 Гб _____. Монитор Встроенный 1920x1080
Другие устройства Touchpad Synaptics _____
5. **Программное обеспечение (лабораторное):**
Операционная система семейства _____, наименование _____ версия _____
интерпретатор команд _____ версия _____
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____

Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных _____
Программное обеспечение ЭВМ студента, если использовалось:
Операционная система семейства UNIX _____, наименование Pop!_OS _____ версия 22.04 jammy
интерпретатор команд bash _____ версия 5.1.16
Система программирования _____ версия _____
Редактор текстов _____ версия _____
Утилиты операционной системы _____
Прикладные системы и программы _____
Местонахождение и имена файлов программ и данных на домашнем компьютере _____

6. Идея, метод, алгоритм решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Написать программу, которая обрабатывает введенные флаги и значения, связанные с ними. Для копирования использовать встроенную утилиту UNIX - cp.

7. Сценарий выполнения работы (план работы, первоначальный текст программы в черновике [можно на отдельном листе] и тесты либо соображения по тестированию)

В программе реализованы несколько функций - функция, выводящая все возможные флаги с их значениями, функция, которая выходит из программы с определенным сообщением.

Первое, что происходит в программе - задаются значения параметров по умолчанию, после этого обрабатываются флаги. По окончании обработки проверяется, валиден ли файл, введенный пользователем. После этого происходит итерирование с изменением префикса посредством инкрементирования кода литеры префикса и копирование с помощью встроенной утилиты UNIX - cp.

Флаги:

- f [FILE_PATH] – ввод пути копируемого файла
- l '[CHAR]' – ввод литеры, с которой начинаются постфиксы скопированных файлов
- c [INT] – ввод числа, которое определяет количество скопированных файлов
- v – выписывать строчку с информацией после каждого скопированного файла
- остальные флаги выводят табличку с правильным использованием программы

единственный обязательный флаг – -f

*Пункты 1-7 отчета составляются строго до начала лабораторной работы.
Допущен к выполнению работы. Подпись преподавателя _____*

8. Распечатка протокола (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем)

```
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs$
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs$ 121/copy.sh ?
Info:

Usage:
REQUIRED ARGUMENTS:

    -f, --file [FILE_PATH]          path to a file that needs to be copied

OPTIONAL ARGUMENTS:

    -l, --first_char '[CHAR]'      from which character, file prefixes start
    -c, --copy_count [COPY_COUNT]  how many times file should be copied
    -v, --verbose                  verbosely make copies of a file

vitos@vitos-hp16:/mnt/c/important/docs/mai/labs$ cd 121
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/121$ ./copy.sh -f test -l 'c' -c 5
Start copying...
Done!
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/121$ ls
./ ../ copy.sh* .gitignore* 121-2012.djvu* protocol.txt* report.pdf* test* testc* testd* teste* testf* testg*
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/121$ ./copy.sh -f test -v
Start copying...
0: copying test to testa...
Done!
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/121$ ./copy.sh -f
Empty filename!

Usage:
REQUIRED ARGUMENTS:

    -f, --file [FILE_PATH]          path to a file that needs to be copied

OPTIONAL ARGUMENTS:

    -l, --first_char '[CHAR]'      from which character, file prefixes start
    -c, --copy_count [COPY_COUNT]  how many times file should be copied
    -v, --verbose                  verbosely make copies of a file

vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/121$ ./copy.sh ?
Info:

Usage:
REQUIRED ARGUMENTS:

    -f, --file [FILE_PATH]          path to a file that needs to be copied

OPTIONAL ARGUMENTS:

    -l, --first_char '[CHAR]'      from which character, file prefixes start
    -c, --copy_count [COPY_COUNT]  how many times file should be copied
    -v, --verbose                  verbosely make copies of a file

vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/121$ ./copy.sh -m asdb
Info:

Usage:
REQUIRED ARGUMENTS:

    -f, --file [FILE_PATH]          path to a file that needs to be copied

OPTIONAL ARGUMENTS:

    -l, --first_char '[CHAR]'      from which character, file prefixes start
    -c, --copy_count [COPY_COUNT]  how many times file should be copied
    -v, --verbose                  verbosely make copies of a file

vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/121$ ./copy.sh -f test -v -c 10
Start copying...
0: copying test to testa...
1: copying test to testb...
2: copying test to testc...
3: copying test to testd...
4: copying test to teste...
5: copying test to testf...
6: copying test to testg...
7: copying test to testh...
8: copying test to testi...
9: copying test to testj...
```

```

Done!
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ls
./ copy.sh*      l21-2012.djvu*  report.pdf*  testa*  testc*  teste*  testg*  testi*
../ .gitignore*  protocol.txt*   test*        testb*  testd*  testf*  testh*  testj*
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ./copy.sh -f copy.sh -c 3 -v
Start copying...
0: copying copy.sh to copya.sh...
1: copying copy.sh to copyb.sh...
2: copying copy.sh to copyc.sh...
Done!
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ./copy.sh -f copy.sh -c -v
./copy.sh: line 45: [: -v: integer expression expected
Start copying...
Done!
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ./copy.sh -f copy.sh -c
Copy count is not present

Usage:
REQUIRED ARGUMENTS:

        -f, --file [FILE_PATH]          path to a file that needs to be copied

OPTIONAL ARGUMENTS:

        -l, --first_char '[CHAR]'        from which character, file prefixes start
        -c, --copy_count [COPY_COUNT]    how many times file should be copied
        -v, --verbose                    verbosely make copies of a file

vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ./copy.sh -f -c
Invalid filename!

Usage:
REQUIRED ARGUMENTS:

        -f, --file [FILE_PATH]          path to a file that needs to be copied

OPTIONAL ARGUMENTS:

        -l, --first_char '[CHAR]'        from which character, file prefixes start
        -c, --copy_count [COPY_COUNT]    how many times file should be copied
        -v, --verbose                    verbosely make copies of a file

vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ./copy.sh
You need to input file that needs to be copied!

Usage:
REQUIRED ARGUMENTS:

        -f, --file [FILE_PATH]          path to a file that needs to be copied

OPTIONAL ARGUMENTS:

        -l, --first_char '[CHAR]'        from which character, file prefixes start
        -c, --copy_count [COPY_COUNT]    how many times file should be copied
        -v, --verbose                    verbosely make copies of a file

vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ cat copy.sh
#!/usr/bin/bash

set -e

print_usage() {
    echo -e "Usage:"
    echo -e "REQUIRED ARGUMENTS:\n"
    echo -e "\t-f, --file [FILE_PATH]\t\t path to a file that needs to be copied\n"
    echo -e "OPTIONAL ARGUMENTS:\n"
    echo -e "\t-l, --first_char '[CHAR]'\t\t from which character, file prefixes start"
    echo -e "\t-c, --copy_count [COPY_COUNT]\t how many times file should be copied"
    echo -e "\t-v, --verbose\t\t\t\t\t verbosely make copies of a file\n"
}

bail() {
    echo -e "$1\n"
    print_usage
    exit 1
}

# default parameters
file=""
char='a'
copy_count=1
verbose=false

while [[ $# -gt 0 ]]; do # handle flags
    case $1 in

```

```

-f|--file)
    file=$2

    if [ ${#file} -eq 0 ]; then
        bail "Empty filename!"
    elif [ ${file:0:1} = - ]; then
        bail "Invalid filename!"
    fi

    shift
    shift ;;
-c|--copy_count)
    copy_count=$2

    if [ ${#copy_count} -eq 0 ]; then
        bail "Copy count is not present"
    elif [ $copy_count -eq 0 ]; then
        bail "Why would you want to copy your file zero times?"
    fi

    shift
    shift ;;
-l|--first_char)
    char=$2

    if [ ${#char} -eq 0 ]; then
        bail "Empty initial character"
    fi

    shift
    shift ;;
-v|--verbose)
    shift
    verbose=true ;;
*)
    bail "Info:" ;;
esac
done

if [ ${#file} -eq 0 ]; then
    bail "You need to input file that needs to be copied!"
fi

echo "Start copying..."

char=${char:1}
asciiicode=$(printf "%d" "'$char'")
let asciiicode--

char=$(printf "\x$(printf %x $asciiicode)")

for (( i=0; $i<$copy_count; ++i ))
do
    asciiicode=$(printf "%d" "'$char'")
    let asciiicode++

    newchar=$(printf "\x$(printf %x $asciiicode)")
    char=$newchar

    last_dot_index=${#file}
    for (( j=0; $j<${#file}; ++j )); do
        if [ "${file:$j:1}" = "." ]; then
            last_dot_index=$j
        fi
    done
    extension_length=$(( ${#file} - $last_dot_index ))

    filename=${file:0:$last_dot_index}$char
    extension=${file:$last_dot_index:$extension_length}

    new_file=$filename$extension

    if [ $verbose = true ]; then
        echo "$i: copying $file to $new_file..."
    fi

    eval "cp \ $file \ $new_file"
done

echo "Done!" vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$

```

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы: _____

11. **Выводы:** В ходе выполнения лабораторной работы я научился писать интерпретируемые программы на языке программирования bash, что очень полезно для повседневного использования. В бэше я больше узнал про обработку строк.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента _____