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: 25 февраля 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 окончанию орбработки 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.

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$ cat l21/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 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}

for (( i=0; $i<$copy_count; ++i ))
do
```

```
    asciicode=$(printf "%d" "'$char'")
    let asciicode++

    newchar=$(printf "\x$(printf %x $asciicode)")
    char=$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/copy.sh -f l21/test
Start copying...
Done!
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs$ cd l21
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ./copy.sh -v -c 5 -f test_file.txt
Start copying...
0: copying test_file.txt to test_fileb.txt...
1: copying test_file.txt to test_filec.txt...
2: copying test_file.txt to test_filed.txt...
3: copying test_file.txt to test_filee.txt...
4: copying test_file.txt to test_filef.txt...
Done!
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ls
./   copy.sh*     l21-2012.djvu*  testb*           test_filec.txt*  test_filee.txt*  test_file.txt*
../  .gitignore*  test*           test_fileb.txt*  test_filed.txt*  test_filef.txt*  test.text.a*
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ./copy.sh -v -c 3 -f test_file.txt -l 'M'
Start copying...
0: copying test_file.txt to test_fileN.txt...
1: copying test_file.txt to test_fileO.txt...
2: copying test_file.txt to test_fileP.txt...
Done!vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ./copy.sh -v -c 3 -f test.text.a -l 'M'
Start copying...
0: copying test.text.a to test.textN.a...
1: copying test.text.a to test.textO.a...
2: copying test.text.a to test.textP.a...
Done!
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ls
./   copy.sh*     l21-2012.djvu*  testb*           test_filec.txt*  test_filee.txt*  test_fileN.txt*  test_fil
eP.txt*  test.text.a*   test.textO.a*
../  .gitignore*  test*           test_fileb.txt*  test_filed.txt*  test_filef.txt*  test_fileO.txt*  test_fil
e.txt*   test.textN.a*  test.textP.a*
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ rm -f test_file*.txt
removed 'test_fileb.txt'
removed 'test_filec.txt'
removed 'test_filed.txt'
removed 'test_filee.txt'
removed 'test_filef.txt'
removed 'test_fileN.txt'
removed 'test_fileO.txt'
removed 'test_fileP.txt'
removed 'test_file.txt'
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ rm -f test.text*.a
removed 'test.text.a'
removed 'test.textN.a'
removed 'test.textO.a'
removed 'test.textP.a'
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ rm testb
rm: remove regular file 'testb'? y
removed 'testb'
vitos@vitos-hp16:/mnt/c/important/docs/mai/labs/l21$ ./copy.sh -c -l
./copy.sh: line 47: [: -l: integer expression expected
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$ ./copy.sh -l '5' -f test -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 -l '5' -f test -m
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 ?
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/l21$ ./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
```

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 _____