

Отчет по лабораторной работе № 21 по курсу Фундаментальная информатика

Студент группы М8О-204Б-22, Филиппов Фёдор Иванович, № по списку 18

Контакты: gooselinjk@yandex.ru

Работа выполнена: “1” октября 2023 года

Преподаватель: Потенко М.А., каф.806

Входной контроль знаний с оценкой _____

Отчёт сдан “3” октября 2023 года, ИО _____

Подпись преподавателя _____

1. Тема: Программирование на интерпретируемых командных языках

2. Цель работы: Составить программу выполнения заданных действий над файлами на языке Bash

3. Задание (вариант №15): Удлинение файлов с указанным суффиксом до заданной длины путем присоединения к ним необходимого числа копий некоторого файла

4. Оборудование

ЭВМ — ноутбук HP, процессор — Ryzen 5500U, с ОП 16384 МБ и НМД 1048576 МБ, Терминал Windows Powershell (с возможностью переключения на UNIX)

5. Программное обеспечение

Операционная система семейства Windows, наименование Windows 11 Home, версия 22H2

Редактор текстов — Sublime Text

Утилиты операционной системы — терминал Windows Powershell

Прикладные системы и программы — Visual Studio Code, Visual Studio

6. Идея, метод, алгоритм решения задачи (в формах: словесной, псевдокода, графической или формальные с пред- и постусловиями)

Основная идея этого скрипта заключается в том, чтобы увеличить размер файлов определенного типа до заданного значения путем добавления содержимого другого файла.

Проверяется, передано ли два аргумента в командной строке. Если нет, выводится сообщение об использовании скрипта и скрипт завершает свою работу.

Сохраняются переданные параметры в переменных `suff` и `length`.

Выводятся параметры, чтобы пользователь видел их значения.

Поиск файлов с расширением, указанным в `suff`, осуществляется с помощью команды `find`.

Файлы, которые можно увеличить, добавляются в массив `files_to_increase`.

Пользователю предлагается ввести путь к файлу, который будет использован для увеличения остальных файлов.

Проверяется, существует ли этот файл, и его размер.

Для каждого файла, который можно увеличить, выполняется цикл, в котором содержимое файла добавляется в конец текущего файла до тех пор, пока размер текущего файла не станет больше или равен заданному `length`.

Выводятся измененные файлы с их новыми размерами.

7. Сценарий выполнения работы (план работы, первоначальный текст программы в черновике и тесты, либо соображения по тестам)

Мой план реализации включал бы следующие шаги: сначала я бы добавил проверку переданных аргументов, чтобы убедиться, что пользователь ввел оба необходимых параметра. Затем я бы извлек параметры, такие как суффикс и желаемый размер файла, и провел поиск файлов в текущем каталоге с соответствующим суффиксом. Далее я бы проверил найденные файлы и сформировал список тех, которые требуют увеличения. Затем бы последовала проверка выбранного файла для заполнения увеличиваемых файлов, чтобы убедиться, что он не является частью списка найденных файлов. После этого я бы добавил код для увеличения размера файлов до указанного значения, используя выбранный файл для заполнения. Я бы проверил, чтобы каждый файл достиг или

превысил желаемый размер. Чтобы гарантировать правильную работу программы, я бы провел тестирование на различных входных данных, включая файлы разных размеров, ситуации отсутствия доступных файлов для заполнения и ошибочный ввод пути к файлу. Я бы также убедился, что программа сообщает об ошибках корректно в случае возникновения проблем.

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

```
1  #!/bin/bash
2
3  # Проверяем, что передано два параметра
4  if [ $# -ne 2 ]; then
5      echo "Использование: $0 <суффикс> <длина[байт]>"
6      exit 1
7  fi
8
9  # Извлекаем параметры
10 suff="$1"
11 length="$2"
12
13 # Выводим параметры
14 echo "Расширение файлов: $suff"
15 echo "Желаемый размер: $length байт"
16
17 echo "Найденные файлы:"
18 file_list=$(find . -type f -name ".*$suff")
19 files_to_increase=()
20
21 # Смотрим, какие файлы подходят для увеличения
22 for file_path in ${file_list[@]}; do
23     file=$(basename "$file_path")
24     file_size=$(stat -c%s "$file")
25     echo "$file [$file_size байт]"
26     if [ "$file_size" -lt "$length" ]; then
27         files_to_increase+=("$file_path")
28     fi
29 done
30
31 # Получаем путь файла, который следует прибавить к остальным
32 read -r -p "Введите путь к файлу, который следует использовать для заполнения: " add_file_path
33
34 # Проверка на то, есть ли этот файл в списке найденных
35 for file_path in ${file_list[@]}; do
36     if [ "$(realpath $file_path)" = "$(realpath $add_file_path)" ]; then
37         echo "Нельзя использовать файлы из списка"
38         exit 1
39     fi
```

```

40 done
41
42 # Существует ли файл
43 if [ -f "$add_file_path" ]; then
44     if [ -s "$add_file_path" ]; then
45         file_size=$(stat -c%s "$add_file_path")
46         echo "Файл существует: $add_file_path [$file_size байт]"
47     else
48         echo "Размер указанного файла равен 0 байт"
49         exit 2
50     fi
51 else
52     echo "Введенный путь неверен: $add_file_path"
53     exit 3
54 fi
55
56 # Пробегаемся по файлам
57 for file_path in ${files_to_increase[@]}; do
58     # Увеличиваем до тех пор, пока размер не станет >= $length
59     while true; do
60         file_size=$(stat -c%s "$file_path")
61
62         if [ "$file_size" -lt "$length" ]; then
63             cat "$add_file_path" >> "$file_path"
64         else
65             break
66         fi
67     done
68 done
69
70 # Вывод измененных файлов с их новыми размерами
71 echo "Размер файлов увеличен:"
72 for file_path in ${files_to_increase[@]}; do
73     file=$(basename "$file_path")
74     file_size=$(stat -c%s "$file")
75     echo "$file [$file_size байт]"
76 done
77
78 exit 0

```

```

root@ravenFIF:/mnt/c/Users/theo_rvn/Desktop/coding/21# ./script.sh txt 2000
Расширение файлов: txt
Желаемый размер: 2000 байт
Найденные файлы:
biba.txt [1256 байт]
boba.txt [1777 байт]
test.txt [1622 байт]
Введите путь к файлу, который следует использовать для заполнения: ../23/main.c
Файл существует: ../23/main.c [1255 байт]
Размер файлов увеличен:
biba.txt [2511 байт]
boba.txt [3032 байт]
test.txt [2877 байт]
root@ravenFIF:/mnt/c/Users/theo_rvn/Desktop/coding/21#

```

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 других 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

Недочеты при выполнении работы могут быть устранены следующим образом: _____

11. Выводы:

В данной лабораторной работе я изучил принципы работы с интерпретируемым командным языком Bash.

Понимание основных концепций и синтаксиса этого языка позволяют быстро и эффективно создавать скрипты для решения различных задач: по большей части, при работе с файловой системой. Например, в ходе выполнения работы были использованы различные элементы языка Bash, включая работу с аргументами командной строки, циклы и операции поиска и присоединения одних файлов к другим. Полученные знания могут быть использованы в будущем для создания скриптов, которые помогут в автоматизации рутинных процессов работы с файловой и операционной системой.