

**TOMSK
POLYTECHNIC
UNIVERSITY**



**ТОМСКИЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ**

Ministry of Science and Higher Education of the Russian Federation
Federal State Autonomous Educational Institution of Higher Education
«National Research Tomsk Polytechnic University» (TPU)
30, Lenin ave., Tomsk, 634050, Russia
Tel. +7-3822-606333, +7-3822-701779,
Fax +7-3822-606444, e-mail: tpu@tpu.ru, tpu.ru
ОКПО (National Classification of Enterprises and Organizations): 02069303,
Company Number: 027000890168,
VAT/KPP (Code of Reason for Registration)
7018007264/701701001, BIC 016902004

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский
Томский политехнический университет» (ТПУ)
Ленина, пр., д. 30, г. Томск, 634050, Россия
тел.: +7-3822-606333, +7-3822-701779,
факс +7-3822-606444, e-mail: tpu@tpu.ru, tpu.ru
ОКПО 02069303, ОГРН 1027000890168,
ИНН/КПП 7018007264/701701001, БИК 016902004

ИШИТР ОИТ

09.04.04 «Программная инженерия»

ОТЧЁТ
по Заданию № 3
«Редакторы, поиск»

дисциплина:

Управление разработкой промышленного программного обеспечения

Исполнитель:

студент группы

Горковенко Р.В.

17.05.2024

Дата сдачи

Руководитель:

преподаватель

Небаба С.Г.

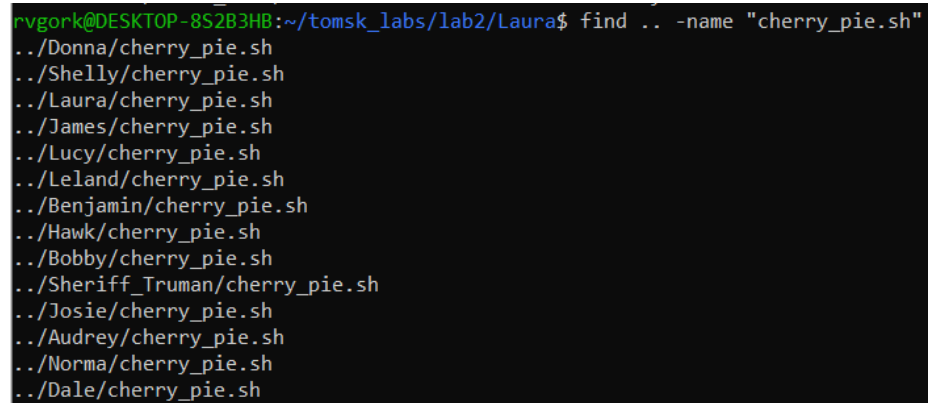
Задание

1. Используйте редактор файлов `vi` или `nano` по Вашему выбору. В данном задании необходимо будет выполнять работу с написанием кода из редактора `bash`-оболочки.
2. У вас уже есть несколько директорий из предыдущего задания, в которых находятся файлы `cherry_pie.sh`, зайдите в любую на выбор.
3. Из этой директории примените команду поиска таких же скриптов и выведите на экран результаты. При желании можно сохранить этот список для использования в следующей части задания.
4. В пустом `bash`-скрипте (в скрипте `cherry_pie.sh`) внесите изменения согласно заданию:
 1. Ищет все файлы `cherry_pie.sh` во всех директориях предыдущего задания (всего 14 директорий - т.е. 14 файлов).
 2. Все пустые найденные файлы (13 из 14-ти будут пустыми, в 14-м у Вас вносятся изменения по заданию) -- нужно проверить, что они пустые, пройдя по списку обнаруженных файлов -- заменяются на данный запущенный скрипт. Примечание: После первого успешного запуска скрипты `cherry_pie.sh` не будут пустыми нигде из этих 14 директорий. Значит, копирования не должно происходить. Не указывайте `hardcoded`-путь самого скрипта.
 3. В корневом каталоге предыдущего задания, в котором расположены 14 директорий, происходит запись в файл `white_lodge.txt` по следующей форме:
`<timestamp> <имя директории (только имя самой директории, а не путь)>
ordered a cherry pie`
Например: Dale ordered a cherry pie
 4. Примечание: Данный скрипт будет копироваться в п.4.2, т.е. он должен быть реализован так, чтобы при копировании скрипта в другую директорию логика работы скрипта не пострадала. Предыдущие записи не стираются. В отчете должно быть представлено минимум 3 запуска - оригинальный первый и из двух других директорий. Необходимо также продемонстрировать содержимое файла `white_lodge.txt` с минимум 3 записями по разным запускам.

Ход работы

Для выполнения работы использовалась Ubuntu 20.04 с помощью подсистемы WSL.

С помощью команды `find` из директории Laura в корневой директории из прошлой работы были найдены все файлы `cherry_pie.sh`.



```
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Laura$ find .. -name "cherry_pie.sh"
../Donna/cherry_pie.sh
../Shelly/cherry_pie.sh
../Laura/cherry_pie.sh
../James/cherry_pie.sh
../Lucy/cherry_pie.sh
../Leland/cherry_pie.sh
../Benjamin/cherry_pie.sh
../Hawk/cherry_pie.sh
../Bobby/cherry_pie.sh
../Sheriff_Truman/cherry_pie.sh
../Josie/cherry_pie.sh
../Audrey/cherry_pie.sh
../Norma/cherry_pie.sh
../Dale/cherry_pie.sh
```

Рисунок 1. Выполнение команды `find`.

В рамках задания был реализован `bash`-скрипт, созданный на базе пустого скрипта `cherry_pie.sh` в директории Hawk, созданных в процессе выполнения предыдущей работы. Скрипт копирует сам себя во все остальные файлы `cherry_pie.sh`, если те пустые, в пределах директорий, созданных в ходе предыдущей работы, а также записывает сообщение с датой, временем и именем своей директории в файл `white_lodge.txt`. Код скрипта приведён в листинге 1.

Листинг 1 – Скрипт «`cherry_pie.sh`»

```
#!/bin/bash
# Getting the directory where the script is saved
CORE_DIR=$( cd -- "$( dirname -- "${BASH_SOURCE[0]}" )" &> /dev/null && pwd )

while IFS='' read -r -d '' filename; do
    if [ ! -s $filename ]; then
        cp "${BASH_SOURCE[0]}" "$filename"
        chmod +x "$filename"
    fi
done < <(find .. -name "cherry_pie.sh" -print0)

my_name=$(basename "`pwd`")
date_time=$(date)
echo "$date_time $my_name ordered a cherry pie" >> "../white_lodge.txt"
```

Результаты тестирования скрипта приведены на рисунках 1 и 2.

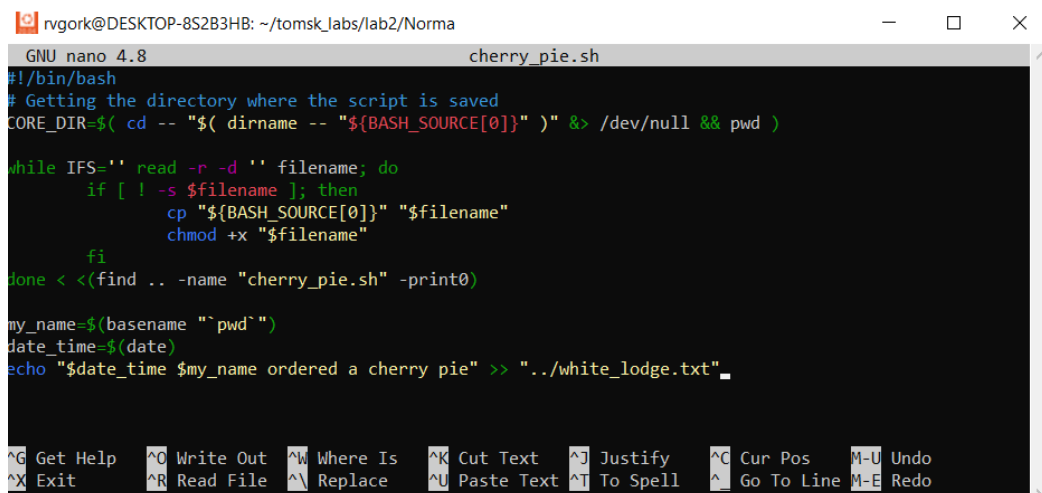
```

rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Hawk$ ls
cherry_pie.sh  good_coffee.sh  post_box
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Hawk$ cd ../Norma
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Norma$ ls
cherry_pie.sh  good_coffee.sh  post_box
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Norma$ cat cherry_pie.sh
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Norma$ cd ../Hawk
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Hawk$ ./cherry_pie.sh
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Hawk$ cd ../Norma
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Norma$ ls
cherry_pie.sh  good_coffee.sh  post_box
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Norma$ ./cherry_pie.sh
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Norma$ cd ../Lucy
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Lucy$ ./cherry_pie.sh
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2/Lucy$ cd ..
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2$ cat white_lodge.txt
Fri May 17 12:55:42 MSK 2024 Hawk ordered a cherry pie
Fri May 17 12:56:31 MSK 2024 Norma ordered a cherry pie
Fri May 17 12:57:17 MSK 2024 Lucy ordered a cherry pie
rvgork@DESKTOP-8S2B3HB:~/tomsk_labs/lab2$

```

Рисунок 2. Тестирование скрипта

Перед запуском скрипта в директории Hawk скрипт в директории Norma даже не является исполняемым. При попытке его прочитать в консоль ничего не выводится, что говорит о том, что он пуст. После запуска скрипта в директории Hawk он также был запущен в директории Norma и Lucy. После этого был прочитан файл white_lodge.txt. Можно убедиться, что в нём хранятся записи обо всех трёх запусках.



```

GNU nano 4.8 cherry_pie.sh
#!/bin/bash
# Getting the directory where the script is saved
CORE_DIR=$( cd -- "$( dirname -- "${BASH_SOURCE[0]}" )" && /dev/null && pwd )

while IFS='' read -r -d '' filename; do
    if [ ! -s $filename ]; then
        cp "${BASH_SOURCE[0]}" "$filename"
        chmod +x "$filename"
    fi
done < <(find .. -name "cherry_pie.sh" -print0)

my_name=$(basename "`pwd`")
date_time=$(date)
echo "$date_time $my_name ordered a cherry pie" >> "../white_lodge.txt"

```

Рисунок 3. Содержимое скрипта в директории Norma

На рисунке 3 представлено содержимое cherry_pie.sh в директории Norma, открытого в редакторе nano. Можно убедиться, что его код соответствует коду скрипта в директории Hawk, приведённому в листинге 1.

Вывод

В ходе работы было осуществлено знакомство с работой с редакторами и поиском в оболочке bash. Был реализован bash-скрипт, копирующий свой код в пустые файлы с таким же именем в соседних директориях, а также оставляющий сообщения о своём выполнении в отдельном файле. В ходе тестирования скрипта было выявлено, что он работает корректно.