БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Факультет: ИНО

Специальность: ИиТП

Индивидуальная практическая работа № 1

по дисциплине “Операционные системы и среды”

Вариант 6

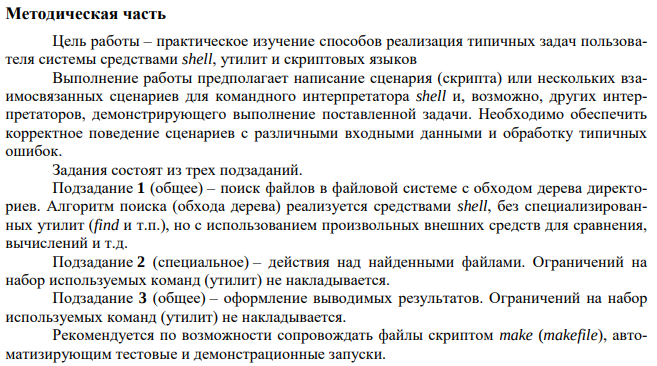
“Средства пользовательского уровня в среде Unix/Linux”

Выполнил студент: Дубейковский А.А.

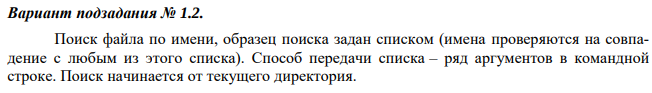
Группа № 893551

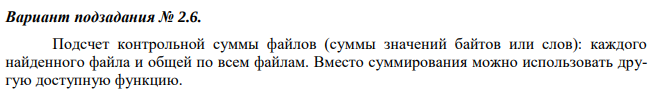
Зачётная книжка № 75350046

**Условие**

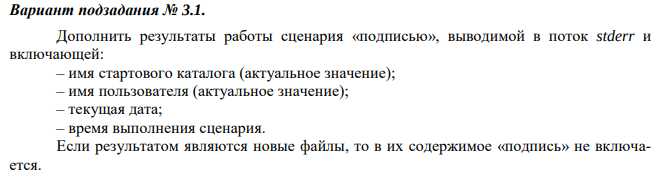


У подзадания 1 всего два варианта, т.к. мой вариант 6й по зачётной книжке, я взял чётный из данных.





У подзадания 3 только один вариант

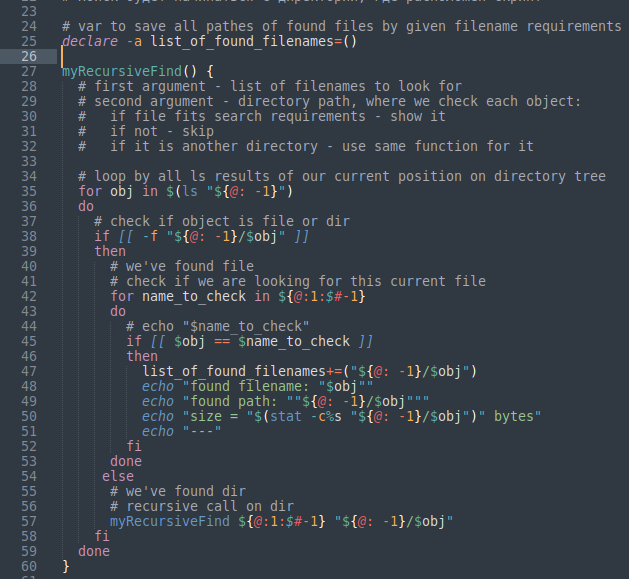
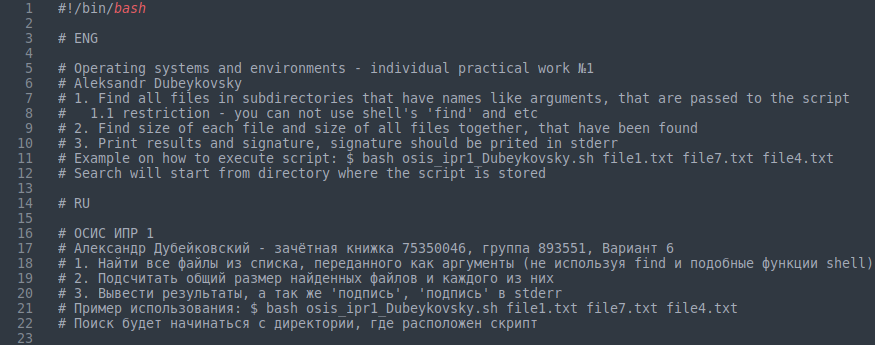


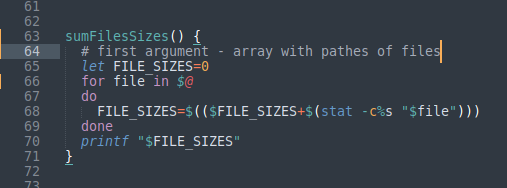
**Скрипт и пояснения к нему**

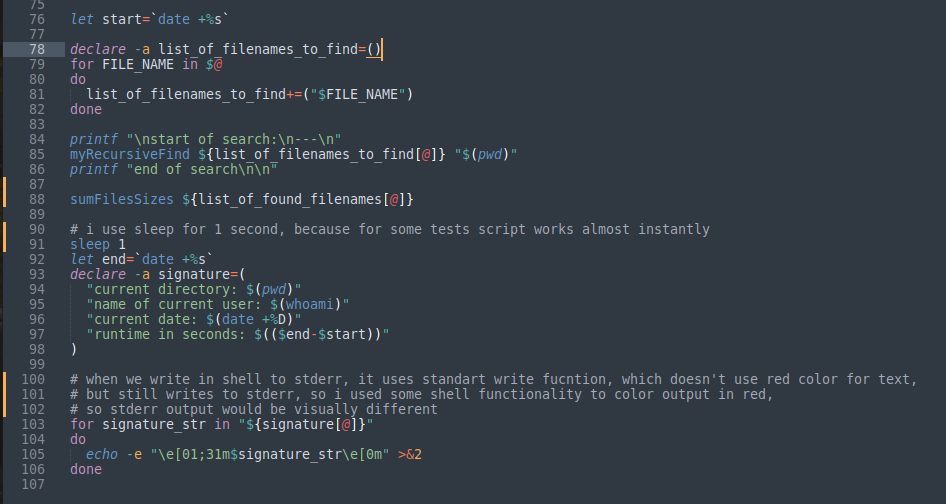
Все необходимые пояснения написаны в комментариях в скрипте, который будет показан тут в двух вариантах (как фото из sublime text для удобного восприятия и чтения, и как текст для копирования). А также скрипт и тесты к нему (готовое дерево директорий и файлы в них) выложены на гитхабе:

<https://github.com/ElephantT/MyBashScripts/tree/main/MyRecursiveFind>

Код из sublime text (удобный для чтения):





Код скрипта для копирования:

#!/bin/bash

# ENG

# Operating systems and environments - individual practical work №1

# Aleksandr Dubeykovsky

# 1. Find all files in subdirectories that have names like arguments, that are passed to the script

# 1.1 restriction - you can not use shell's 'find' and etc

# 2. Find size of each file and size of all files together, that have been found

# 3. Print results and signature, signature should be prited in stderr

# Example on how to execute script: $ bash osis\_ipr1\_Dubeykovsky.sh file1.txt file7.txt file4.txt

# Search will start from directory where the script is stored

# RU

# ОСИС ИПР 1

# Александр Дубейковский - зачётная книжка 75350046, группа 893551, Вариант 6

# 1. Найти все файлы из списка, переданного как аргументы (не используя find и подобные функции shell)

# 2. Подсчитать общий размер найденных файлов и каждого из них

# 3. Вывести результаты, а так же 'подпись', 'подпись' в stderr

# Пример использования: $ bash osis\_ipr1\_Dubeykovsky.sh file1.txt file7.txt file4.txt

# Поиск будет начинаться с директории, где расположен скрипт

# var to save all pathes of found files by given filename requirements

declare -a list\_of\_found\_filenames=()

myRecursiveFind() {

# first argument - array of filenames to look for

# second argument - directory path, where we check each object:

# if file fits search requirements - show it

# if not - skip

# if it is another directory - use same function for it

# loop by all ls results of our current position on directory tree

for obj in $(ls "${@: -1}")

do

# check if object is file or dir

if [[ -f "${@: -1}/$obj" ]]

then

# we've found file

# check if we are looking for this current file

for name\_to\_check in ${@:1:$#-1}

do

# echo "$name\_to\_check"

if [[ $obj == $name\_to\_check ]]

then

list\_of\_found\_filenames+=("${@: -1}/$obj")

echo "found filename: "$obj""

echo "found path: ""${@: -1}/$obj"""

echo "size = "$(stat -c%s "${@: -1}/$obj")" bytes"

echo "---"

fi

done

else

# we've found dir

# recursive call on dir

myRecursiveFind ${@:1:$#-1} "${@: -1}/$obj"

fi

done

}

sumFilesSizes() {

# first argument - array with pathes of files

let FILE\_SIZES=0

for file in $@

do

FILE\_SIZES=$(($FILE\_SIZES+$(stat -c%s "$file")))

done

printf "Size of all found files = "

printf "$FILE\_SIZES"

printf " bytes\n"

}

let start=`date +%s`

declare -a list\_of\_filenames\_to\_find=()

for FILE\_NAME in $@

do

list\_of\_filenames\_to\_find+=("$FILE\_NAME")

done

printf "\nstart of search:\n---\n"

myRecursiveFind ${list\_of\_filenames\_to\_find[@]} "$(pwd)"

printf "end of search\n\n"

sumFilesSizes ${list\_of\_found\_filenames[@]}

# i use sleep for 1 second, because for some tests script works almost instantly

sleep 1

let end=`date +%s`

declare -a signature=(

"current directory: $(pwd)"

"name of current user: $(whoami)"

"current date: $(date +%D)"

"runtime in seconds: $(($end-$start))"

)

# when we write in shell to stderr, it uses standart write fucntion, which doesn't use red color for text,

# but still writes to stderr, so i used some shell functionality to color output in red,

# so stderr output would be visually different

for signature\_str in "${signature[@]}"

do

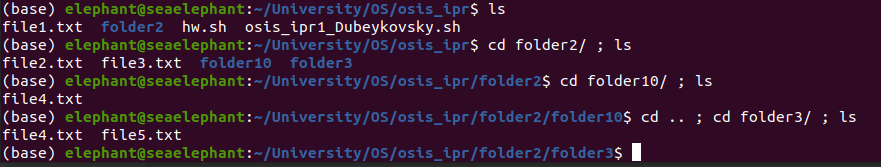
echo -e "\e[01;31m$signature\_str\e[0m" >&2

done

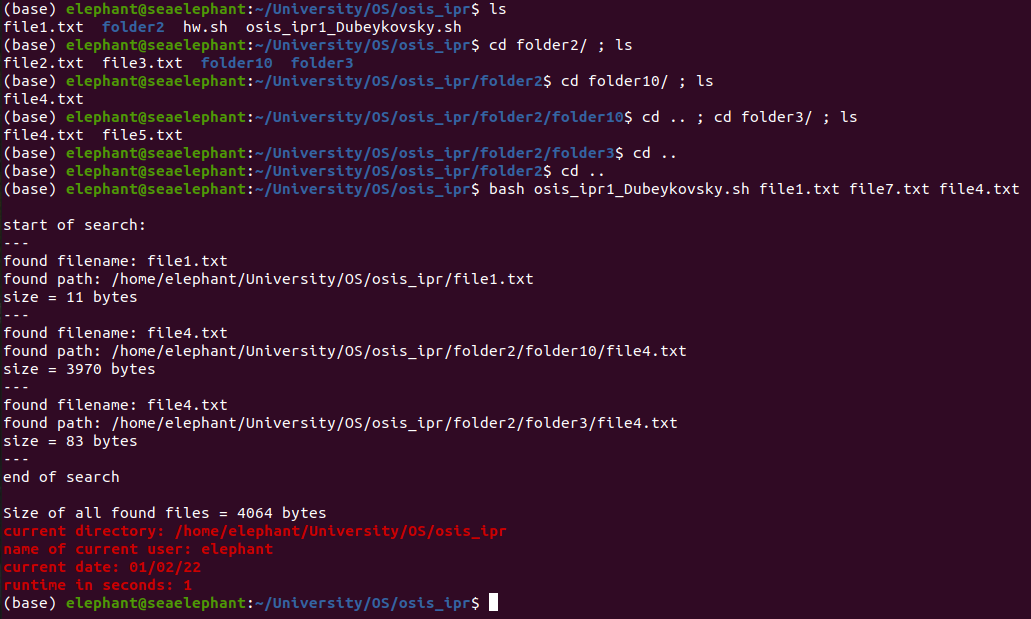
**Тесты**

Тесты запускались на ОС семейства Linux – Ubuntu 20.04.3 LTS (Focal Fossa).

Структура директорий и их содержание для тестов:



Запустим скрипт:



Данный скрипт, а также директории и файлы для тестов (из примера выше) можно скачать на гит-хабе и запустить на своём компьютере в linux shell:

<https://github.com/ElephantT/MyBashScripts/tree/main/MyRecursiveFind>