

Міністерство освіти і науки України
Одеський національний політехнічний університет
Інститут комп'ютерних систем
Кафедра інформаційних систем

Лабораторна робота №5
з дисципліни «Операційні Системи»

Тема: «Основи використання скриптової мови інтерпретатору оболонки
командного рядку»

Виконав:
ст. гр. АІ-204
Колесник К. В.

Перевірив:
Блажко О. А.

Одеса – 2021

Мета роботи: придбання навичок автоматизації керування ОС з використанням скриптової мови інтерпретатору оболонки командного рядку.

Завдання для виконання:

1. Розробіть програму *MyOSParam* на мові *BASH*, яка виводить на екран лише окремі дані про параметри поточного стану ОС з віртуальної файлової системи *procsfs* у відповідності із варіантом з таблиці.

Програма повинна:

- 1) отримувати назву параметру з командного рядка;
- 2) знаходити значення параметру у відповідному файлі
- 3) виводити на екран значення параметру та опис призначення параметру українською мовою.

№ команди	№ учасника команди	1-й параметр пам'яті	2-й параметр пам'яті	Параметр процесеру
6	1	<i>Active(file)</i>	<i>Committed_AS</i>	<i>apicid</i>

2. Розробіть програму на мові *BASH*, яка виконує дії у відповідності з варіантом, представленим у таблиці.

№ команди	№ учасника команди	Назва файлу-скрипту	Обмеження на довжину назви	Обмеження на зміст назви
-----------	--------------------	---------------------	----------------------------	--------------------------

6	1	<i>my_delete_file</i>	Не менше 13	Не менше 2 цифр підряд
---	---	-----------------------	-------------	---------------------------

Для всіх варіантів передбачається наступний опис кроків алгоритму роботи програми:

1) запропонувати ввести назву об'єкту, використовуючи українське запрошення;

2) перевірити відсутність в ОС об'єкту з такою назвою, використовуючи відповідні команди та файли;

3) якщо об'єкт присутній, тоді вивести на екран відповідне повідомлення про помилку та завершити роботу програми;

4) якщо назва об'єкту не відповідає вказаному обмеженню, тоді вивести на екран відповідне повідомлення про помилку та завершити роботу програми;

5) якщо назва об'єкту відповідає вказаному обмеженню, виконати відповідну команду.

У кроках алгоритму об'єктом може виступати файл або каталог, в залежності від варіанту завдання.

3. В попередній лабораторній роботі ви створили файл CSV-формату за пунктом 2.3

Розробіть програму на мові BASH, яка автоматично:

– читає зміст файлу з урахуванням табличної структури (рядки, стовпчики), використовуючи перенаправлення потоку;

– створює каталог з назвою, співпадаючою з назвою 3-го стовпчика таблиці файлу

– у створеному каталозі створює файли, назви яких співпадають зі значеннями 3-го стовпчика таблиці файлу

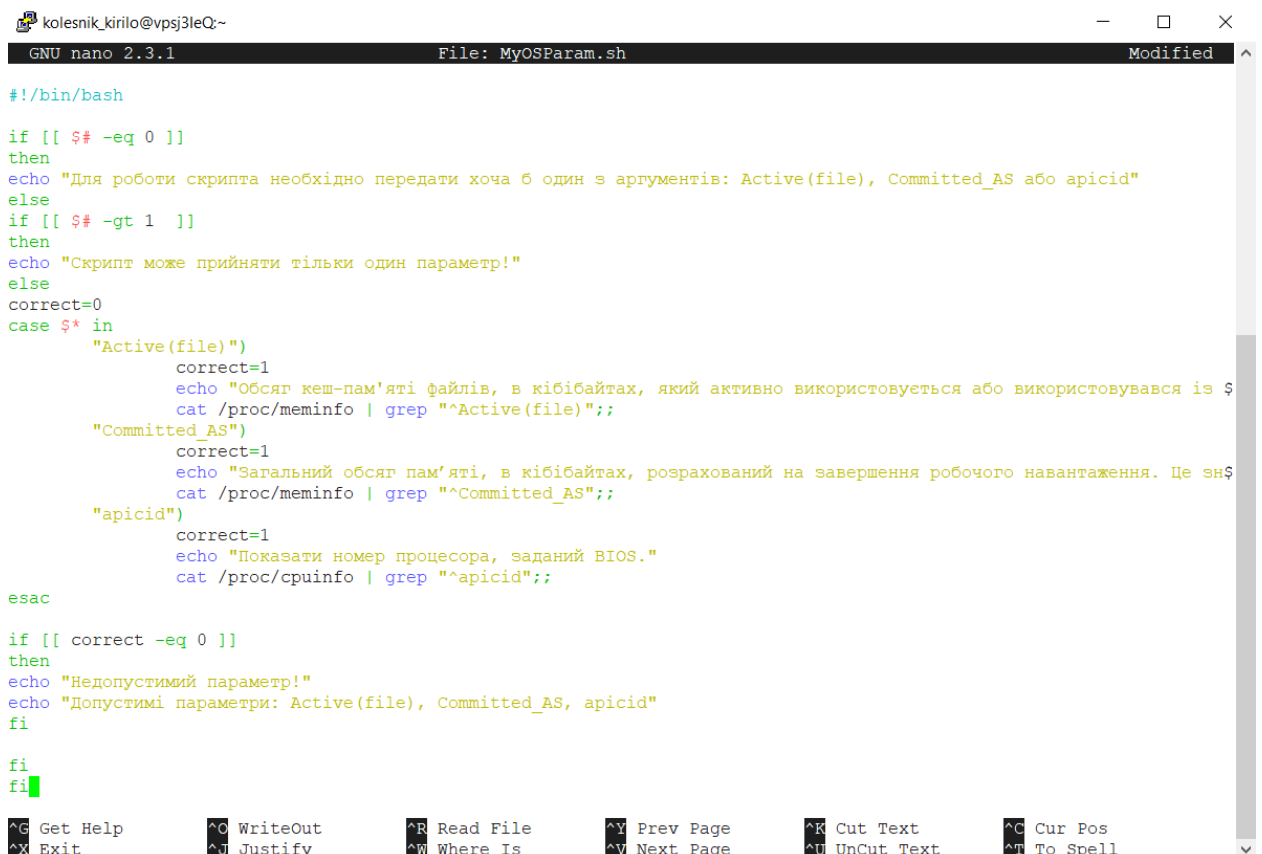
Хід роботи:

0. Скопійовано файл підсвічування синтаксису скриптів у свій домашній каталог користувача (рис. 0).

```
[kolesnik_kirilo@vpsj3IeQ ~]$ cp /usr/share/nano/sh.nanorc ~/.nanorc  
[kolesnik_kirilo@vpsj3IeQ ~]$
```

Рис. 0

1. Розроблено програму MyOSParam на мові BASH, яка виводить на екран лише окремі дані про параметри поточного стану ОС з віртуальної файлової системи procfs за заданими параметрами Active(file), Committed_AS – для виведення даних про оперативну пам'ять, apicid – для виведення інформації щодо процесору (код – рис. 1.1, результат виконання – рис. 1.2).



```
kolesnik_kirilo@vpsj3IeQ:~  
GNU nano 2.3.1 File: MyOSParam.sh Modified  
#!/bin/bash  
  
if [[ $# -eq 0 ]]  
then  
echo "Для роботи скрипта необхідно передати хоча б один з аргументів: Active(file), Committed_AS або apicid"  
else  
if [[ $# -gt 1 ]]  
then  
echo "Скрипт може прийняти тільки один параметр!"  
else  
correct=0  
case $* in  
    "Active(file)")  
        correct=1  
        echo "Обсяг кеш-пам'яті файлів, в кібібайтах, який активно використовується або використовувався із $"  
        cat /proc/meminfo | grep "^Active(file)";;  
    "Committed_AS")  
        correct=1  
        echo "Загальний обсяг пам'яті, в кібібайтах, розрахований на завершення робочого навантаження. Це зн$"  
        cat /proc/meminfo | grep "^Committed_AS";;  
    "apicid")  
        correct=1  
        echo "Показати номер процесора, заданий BIOS."  
        cat /proc/cpuinfo | grep "^apicid";;  
esac  
  
if [[ correct -eq 0 ]]  
then  
echo "Недопустимий параметр!"  
echo "Допустимі параметри: Active(file), Committed_AS, apicid"  
fi  
  
fi  
fi
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

Рис. 1.1

```

kolesnik_kirilo@vpsj3leQ:~
[kolesnik_kirilo@vpsj3leQ ~]$ sh MyOSParam.sh
Для роботи скрипта необхідно передати хоча б один з аргументів: Active(file), Committed_AS або apicid
[kolesnik_kirilo@vpsj3leQ ~]$ sh MyOSParam.sh parametr
Недопустимий параметр!
Допустимі параметри: Active(file), Committed_AS, apicid
[kolesnik_kirilo@vpsj3leQ ~]$ sh MyOSParam.sh Committed_AS apicid
Скрипт може прийняти тільки один параметр!
[kolesnik_kirilo@vpsj3leQ ~]$ sh MyOSParam.sh Active\ (file\)
Обсяг кеш-пам'яті файлів, в кібібайтах, який активно використовується або використовувався із останнього часу, коли с
истема відновила пам'ять.
Active(file):      373356 kB
[kolesnik_kirilo@vpsj3leQ ~]$ sh MyOSParam.sh Committed_AS
Загальний обсяг пам'яті, в кібібайтах, розрахований на завершення робочого навантаження. Це значення представляє знач
ення найгіршого сценарію, а також включає пам'ять підкачки.
Committed_AS:     3487952 kB
[kolesnik_kirilo@vpsj3leQ ~]$ sh MyOSParam.sh apicid
Показати номер процесора, заданий BIOS.
apicid            : 0
apicid            : 1
[kolesnik_kirilo@vpsj3leQ ~]$

```

Рис. 1.2

2. Розроблено програму `my_delete_file` на мові BASH, яка виконує видалення файлу із обмеженням не менше 13ти символів на довжину назви та не менше 2х цифр підряд (код – рис. 2.1, результат виконання – рис. 2.2).

```

kolesnik_kirilo@vpsj3leQ:~
GNU nano 2.3.1      File: my_delete_file.sh

#!/bin/bash

while [[ "$file_name" = "" ]]
do
    echo -n "Введіть назву файлу: "
    read file_name

    if [[ -n "$file_name" ]]; then
        if [[ -f $file_name ]]; then
            if [[ ${#file_name} -lt 13 ]]; then
                echo "Назва файлу має містити не менше 13 символів."
            else
                if [[ $file_name =~ [0-9]{2,} ]]; then
                    rm $file_name
                    echo "Файл $file_name успішно видалено."
                else
                    echo "Назва файлу має містити не менше 2 цифр підряд"
                fi
            fi
        fi
    else
        echo "Файлу $file_name не існує."
    fi
done

[ Read 24 lines ]
^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page     ^U UnCut Text    ^T To Spell

```

Рис. 2.1

```
kolesnik_kirilo@vpsj3IeQ:~  
[kolesnik_kirilo@vpsj3IeQ ~]$ ls  
123          3.csv          kolesnik_lab_3  newCatalog      student.txt  
1234567890123 accounts.csv    lab3File.html   Operating-System.-Laboratory-Work-1 Нестеренко.pdf  
1.csv        KolesnikKirilo.csv my_delete_file.sh Operating-System.-Laboratory-Work-4  
ltestfileforlab5 KolesnikKirilo.docx MyOSParam.sh    os.lab1.cp1251.html  
2.csv        KolesnikKirilo.pdf MyOSParam.sh.save os.lab1.utf.html  
[kolesnik_kirilo@vpsj3IeQ ~]$ sh my_delete_file.sh  
Введіть назву файлу: 123  
Назва файлу має містити не менше 13 символів.  
[kolesnik_kirilo@vpsj3IeQ ~]$ sh my_delete_file.sh  
Введіть назву файлу: ltestfileforlab5  
Назва файлу має містити не менше 2 цифр підряд  
[kolesnik_kirilo@vpsj3IeQ ~]$ sh my_delete_file.sh  
Введіть назву файлу: 1234567890123  
Файл 1234567890123 успішно видалено.  
[kolesnik_kirilo@vpsj3IeQ ~]$ ls  
123          accounts.csv    lab3File.html   Operating-System.-Laboratory-Work-1 Нестеренко.pdf  
1.csv        KolesnikKirilo.csv my_delete_file.sh Operating-System.-Laboratory-Work-4  
ltestfileforlab5 KolesnikKirilo.docx MyOSParam.sh    os.lab1.cp1251.html  
2.csv        KolesnikKirilo.pdf MyOSParam.sh.save os.lab1.utf.html  
3.csv        kolesnik_lab 3  newCatalog      student.txt  
[kolesnik_kirilo@vpsj3IeQ ~]$
```

Рис. 2.2

3. Розроблено програму на мові BASH, яка автоматично читає зміст файлу з урахуванням табличної структури (рядки, стовпчики) використовуючи перенаправлення потоку; створює каталог з назвою, співпадаючою з назвою 3-го стовпчика таблиці файлу у створеному каталозі створює файли, назви яких співпадають зі значеннями 3-го стовпчика таблиці файлу (код – рис. 3.1, результат виконання – рис. 3.2, 3.3).

```
#!/bin/bash

while [[ "$file_name" = "" ]]
do
    echo -n "Введіть назву файлу: "
    read file_name

    if [[ -n "$file_name" ]]; then
        if [[ -f $file_name ]]; then
            if [[ $file_name =~ .csv ]]; then
                line_num=0
                numberOfFiles=0
                while read line
                do
                    if [[ line_num -eq 0 ]]; then
                        dir=$(echo $line | cut -d, -f3 | sed 's/ /_/g' | sed 's/[^a-zA-Za-яA-Я0-9]//g')
                        echo "Створення каталогу $dir ..."
                        mkdir $dir
                        pushd $dir
                        line_num=1
                    else
                        file=$(echo $line | cut -d, -f3 | sed 's/ /_/g' | sed 's/[^a-zA-Za-яA-Я0-9]//g')
                        if [[ -n "$file" ]]; then
                            if [[ -f $file ]]; then
                                echo "Файл $file вже створений!"
                            else
                                touch $file
                                echo "Створення файлу $dir/$file ..."
                                (( numberOfFiles++ ))
                            fi
                        fi
                    fi
                done < $file_name
                echo "Створено каталог та $numberOfFiles файл(ів)."
```

Рис. 3.1

```
[kolesnik_kirilo@vpsj3IeQ ~]$ ls
123          3.csv          KolesnikKirilo.pdf  my_dir_csv.sh      Operating-System.-Laboratory-Work-1  student.txt
1.csv        accounts.csv        kolesnik_lab_3      MyOSParam.sh       Operating-System.-Laboratory-Work-4  Нестеренко.pdf
ltestfileforlab5  KolesnikKirilo.csv  lab3File.html       MyOSParam.sh.save  os.lab1.cp1251.html
2.csv        KolesnikKirilo.docx  my_delete_file.sh   newCatalog         os.lab1.utf.html
[kolesnik_kirilo@vpsj3IeQ ~]$ sh my_dir_csv.sh
Введіть назву файлу: test
Файлу test не існує!
[kolesnik_kirilo@vpsj3IeQ ~]$ sh my_dir_csv.sh
Введіть назву файлу: student.txt
Вказаний файл повинен мати розширення ".csv".
[kolesnik_kirilo@vpsj3IeQ ~]$ sh my_dir_csv.sh
Введіть назву файлу: KolesnikKirilo.csv
Створення каталогу Обсягивикидівт ...
~/Обсягивикидівт ~
Створення файлу Обсягивикидівт/201209 ...
Створення файлу Обсягивикидівт/526 ...
Створення файлу Обсягивикидівт/4497 ...
Створення файлу Обсягивикидівт/457 ...
Створення файлу Обсягивикидівт/8315 ...
Створення файлу Обсягивикидівт/16059 ...
Створення файлу Обсягивикидівт/3946 ...
Створення файлу Обсягивикидівт/485 ...
Створення файлу Обсягивикидівт/19007 ...
Створення файлу Обсягивикидівт/3158 ...
Створення файлу Обсягивикидівт/254 ...
```

Рис. 3.2

```

Створення файлу Обсягивикидівт/1/13 ...
Створення файлу Обсягивикидівт/202 ...
Створення файлу Обсягивикидівт/1185 ...
Створено каталог та 33 файл(ів).
[kolesnik_kirilo@vpsj3IeQ ~]$ ls
123          3.csv          KolesnikKirilo.pdf  my_dir_csv.sh      Operating-System.-Laboratory-Work-1  student.txt
1.csv        accounts.csv     kolesnik_lab_3      MyOSParam.sh       Operating-System.-Laboratory-Work-4  Нестеренко.pdf
1testfileforlab5  KolesnikKirilo.csv  lab3File.html      MyOSParam.sh.save  os.lab1.cp1251.html                 Обсягивикидівт
2.csv        KolesnikKirilo.docx  my_delete_file.sh  newCatalog          os.lab1.utf.html
[kolesnik_kirilo@vpsj3IeQ ~]$ pushd Обсягивикидівт
~/Обсягивикидівт ~
[kolesnik_kirilo@vpsj3IeQ Обсягивикидівт]$ ls
01 11503 1238 151 16059 1719 19007 202 242 2739 30036 3946 457 485 5307 703 833
03 1185 1264 1545 1645 187 201209 239 254 2803 3158 4497 476 526 614 8315
[kolesnik_kirilo@vpsj3IeQ Обсягивикидівт]$

```

Рис. 3.3

Висновок: Під час виконання цієї лабораторної роботи, було на практиці розглянуто принципи взаємодії типу “клієнт – сервер” із детальним дослідженням функцій та можливостей роботи із скриптовою мовою інтерпретатору оболонки командного рядку у Unix-подібних ОС відповідного інтерфейсу. Виявлено, що користувач отримує можливість створювати скрипти для виконання послідовних потрібних інструкцій, передбачених користувачем. Таким чином, всі процеси та операції, розглянуті під час виконання попередніх лабораторних робіт можна комбінувати за допомогою мови Bash.

На мою думку, найбільш складними виявилось останнє завдання, тому що воно виявилось найбільш об’ємним та передбачувало роботу з файлами та перенаправленням потоку. Тим не менш, методичні вказівки до цієї та попередніх робіт, а також поради з мережі інтернет допомогли вірно реалізувати скрипт.

