

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
Белобров А. О.

Перевірив:
Блажко О. А.

Одеса – 2021

Мета: 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 procfs 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 процесору
3	1	<i>Cached</i>	<i>KernelStack</i>	<i>stepping</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
3	1	<i>my_delete_file</i>	Не більше 25	Не більше 6 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. Копіюємо файл підсвітки синтаксису скриптів у свій домашній каталог користувача (рис. 0).

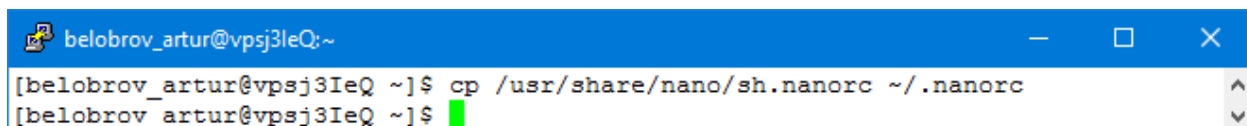



Рис. 0

1. Розробляють програму MyOSParam на мові BASH, яка виводить на екран лише окремі дані про параметри поточного стану ОС з віртуальної файлової системи procfs за заданих параметрах Cached, KernelStacks – для виведення даних про оперативну пам'ять, stepping – для виведення інформації щодо процесору (код – рис. 1.1, результат виконання – рис 1.2).



```
belobrov_artur@vpsj3leQ ~
GNU nano 2.3.1 File: MyOsParam.sh

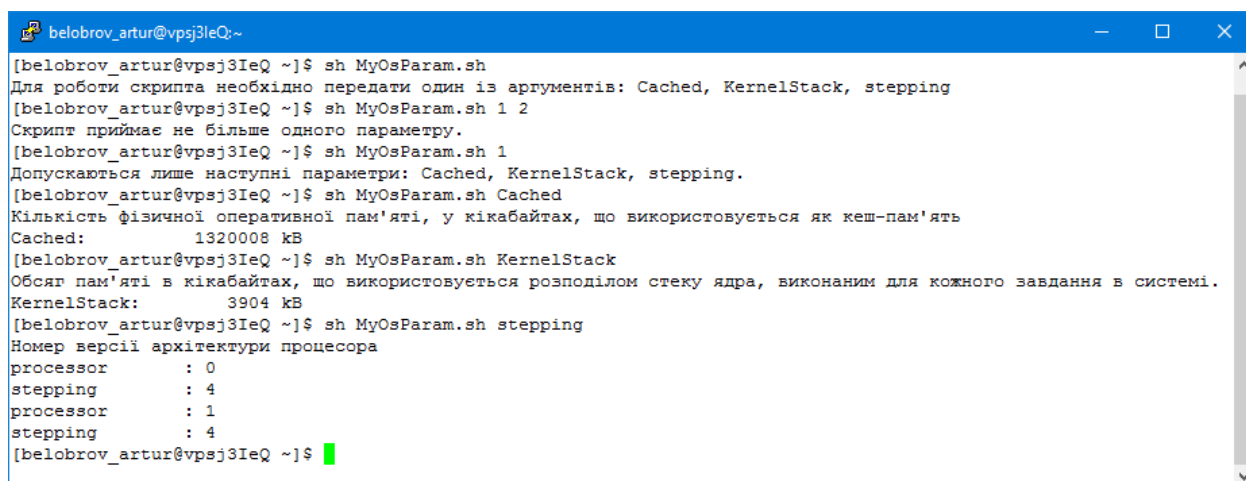
#!/bin/bash

if (( $# == 0 ))
then
echo "Для роботи скрипта необхідно передати один із аргументів: Cached, KernelStack, stepping"
else
if (( $# > 1 ))
then
echo "Скрипт приймає не більше одного параметру."
else
Par=0
case $* in
    "Cached")
        Par=1
        echo "Кількість фізичної оперативної пам'яті, у кілобайтах, що використовується як кеш-пам'ять"
        cat /proc/meminfo | grep "Cached":
    "KernelStack")
        Par=1
        echo "Обсяг пам'яті в кілобайтах, що використовується розподілом стеку ядра, виконанням для кожного завдання в системі."
        cat /proc/meminfo | grep "KernelStack":
    "stepping")
        Par=1
        echo "Номер версії архітектури процесора"
        cat /proc/cpuinfo | grep "processor" | grep "stepping":
    *)
        Par=0
esac

if (( Par == 0 ))
then
echo "Допускаються лише наступні параметри: Cached, KernelStack, stepping."
fi
fi
fi

Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

Рис. 1.1



```
belobrov_artur@vpsj3leQ ~
[belobrov_artur@vpsj3leQ ~]$ sh MyOsParam.sh
Для роботи скрипта необхідно передати один із аргументів: Cached, KernelStack, stepping
[belobrov_artur@vpsj3leQ ~]$ sh MyOsParam.sh 1 2
Скрипт приймає не більше одного параметру.
[belobrov_artur@vpsj3leQ ~]$ sh MyOsParam.sh 1
Допускаються лише наступні параметри: Cached, KernelStack, stepping.
[belobrov_artur@vpsj3leQ ~]$ sh MyOsParam.sh Cached
Кількість фізичної оперативної пам'яті, у кілобайтах, що використовується як кеш-пам'ять
Cached:          1320008 kB
[belobrov_artur@vpsj3leQ ~]$ sh MyOsParam.sh KernelStack
Обсяг пам'яті в кілобайтах, що використовується розподілом стеку ядра, виконанням для кожного завдання в системі.
KernelStack:     3904 kB
[belobrov_artur@vpsj3leQ ~]$ sh MyOsParam.sh stepping
Номер версії архітектури процесора
processor        : 0
stepping        : 4
processor        : 1
stepping        : 4
[belobrov_artur@vpsj3leQ ~]$
```

Рис. 1.2

2. Розробляють програму `my_delete_file` на мові BASH, яка виконує видалення файлу із обмеженням не більше 25ти символів на довжину назви та не більше 6ти цифр підряд (код – рис 2.1, результат виконання – рис 2.2).

```

belobrov_artur@vpsj3leQ:~
GNU nano 2.3.1      File: my_delete_file.sh      Modified
#!/bin/bash

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

    if [[ -n "$file_name" ]]
    then

        if [[ -f "$file_name" ]]
        then
            if [[ ${#file_name} -gt 25 ]]
            then
                echo "Назва файлу не може містити більше 25 символів."
            else
                if [[ $file_name =~ [0-9]{7,} ]]
                then
                    echo "Назва файлу не може містити більше 6 цифр підряд."
                else
                    rm $file_name
                    echo "файл $file_name успішно видалено."
                fi
            fi
        else
            echo "файлу $file_name не існує."
        fi
    fi
done

^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

```

belobrov_artur@vpsj3leQ:~
[belobrov_artur@vpsj3leQ ~]$ ls
1234567      1.csv      accounts.csv      BelobrovArthur.pdf      Cow.txt      Operating-System-Laboratory-Work-4      test.html
12345678901234567890123456      2.csv      BelobrovArthur.csv      belobrov_lab_3      my_delete_file.sh      os.labl.cpi1251.html
123456a      3.csv      BelobrovArthur.docx      Catalog      MyOsParam.sh      os.labl.utf.html

[belobrov_artur@vpsj3leQ ~]$ sh my_delete_file.sh
Введіть назву файлу: 1234567890
файлу 1234567890 не існує.

[belobrov_artur@vpsj3leQ ~]$ sh my_delete_file.sh
Введіть назву файлу: 12345678901234567890123456
Назва файлу не може містити більше 25 символів.

[belobrov_artur@vpsj3leQ ~]$ sh my_delete_file.sh
Введіть назву файлу: 123456a
файл 123456a успішно видалено.

[belobrov_artur@vpsj3leQ ~]$ ls
1234567      2.csv      BelobrovArthur.csv      belobrov_lab_3      my_delete_file.sh      os.labl.cpi1251.html
12345678901234567890123456      3.csv      BelobrovArthur.docx      Catalog      MyOsParam.sh      os.labl.utf.html
1.csv      accounts.csv      BelobrovArthur.pdf      Cow.txt      Operating-System-Laboratory-Work-4      test.html
[belobrov_artur@vpsj3leQ ~]$

```

Рис 2.2

3. Розробляють 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, 3.2, результат виконання – рис 3.3, 3.4).

```

GNU nano 2.8.1 File: my mkdir by csv.sh

#!/bin/bash

while [[ "$file_name" != "" ]]
do
    echo -n "Enter a new $dir: "
    read file_name

    if [[ -n "$file_name" ]]
    then
        if [[ -f "$file_name" ]]
        then
            if [[ $file_name == *.csv ]]
            then
                if (( $(head -n 1 $file_name | awk -F, '{print NF}' ) >= 3 ))
                then
                    line_num=0
                    files_count=0

                    while read line
                    do
                        if (( line_num == 0 ))
                        then
                            dir=$(echo $line | cut -d, -f3 | sed 's/ /_/g' | sed 's/["a-zA-Z0-9-V]//g')
                            echo "Creating directory $dir ..."
                            mkdir $dir
                            pushd $dir
                            line_num=1
                        else
                            file=$(echo $line | cut -d, -f3 | sed 's/ /_/g' | sed 's/["a-zA-Z0-9-V]//g')
                            if [[ -n "$file" ]]
                            then
                                if [[ -f "$file" ]]
                                then
                                    echo "File $file was created..."
                                else
                                    touch $file
                                    echo "Creating $dir/$file..."
                                fi
                            fi
                        fi
                    done
                fi
            fi
        fi
    fi
done

```

Рис. 3.1

```

GNU nano 2.8.1 File: my mkdir by curv.sh

do
    if (( line_num == 0 ))
    then
        dir=$(echo $line | cut -d, -f3 | sed 's/ /_/g' | sed 's/["a-zA-Zs-sd-0]//g')
        echo "Creating temporary dir ..."
        mkdir $dir
        pushd $dir
        line_num=1
    else
        file=$(echo $line | cut -d, -f3 | sed 's/ /_/g' | sed 's/["a-zA-Zs-sd-0]//g')
        if [[ -d "$file" ]]
        then
            if [[ -f "$file" ]]
            then
                echo "$dir $file are overlapping..."
            else
                touch $file
                echo "Creating $dir/$file..."
                (( files_count++ ))
            fi
        fi
        fi
        fi
done < $file_name

echo "Creating 1 variable to $files_count subdirs..."
else
    echo "Warning! $dir already exists 2-x times..."
fi
    echo "Warning! $dir contains files previously \"$dir\"."
fi
    echo "$dir $file_name are done."
fi
done

```

```
belobrov_artur@vpsj3IeQ:~$ ls
11 accounts.csv belobrov_lab_3 my_mkdir_by_csv.sh os.lab1.utf.html
111.csv BelobrovArthur.csv Catalog MyOsParam.sh test.html
2.csv BelobrovArthur.docx Cow.txt Operating-System-Laboratory-Work-4
3.csv BelobrovArthur.pdf my_delete_file.sh os.lab1.cp1251.html
[belobrov_artur@vpsj3IeQ ~]$ sh my_mkdir_by_csv.sh
Введіть назву файла: 1
файлу 1 не існує.
[belobrov_artur@vpsj3IeQ ~]$ sh my_mkdir_by_csv.sh
Введіть назву файла: 11
Вказаний файл повинен мати розширення ".csv".
[belobrov_artur@vpsj3IeQ ~]$ cat 111.csv
A, B
aa, bb
[belobrov_artur@vpsj3IeQ ~]$ sh my_mkdir_by_csv.sh
Введіть назву файла: 111.csv
Вказаний файл містить менше 3-х стовбців.
[belobrov_artur@vpsj3IeQ ~]$ sh my_mkdir_by_csv.sh
Введіть назву файла: BelobrovArthur.csv
Створення каталогу ВеликаРогатаХудобаУськогоГолів ...
~/ВеликаРогатаХудобаУськогоГолів ~
Створення файлу ВеликаРогатаХудобаУськогоГолів/414 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/443 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/76 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/1529 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/900 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/2707 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/354 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/399 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/317 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/183 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/1900 ...
Створення файлу ВеликаРогатаХудобаУськогоГолів/223 ...
Створено 1 каталог та 12 файл(ів).
[belobrov_artur@vpsj3IeQ ~]$ ls
11 BelobrovArthur.csv Cow.txt os.lab1.cp1251.html
111.csv BelobrovArthur.docx my_delete_file.sh os.lab1.utf.html
2.csv BelobrovArthur.pdf my_mkdir_by_csv.sh test.html
3.csv belobrov_lab_3 MyOsParam.sh ВеликаРогатаХудобаУськогоГолів
accounts.csv Catalog Operating-System-Laboratory-Work-4
[belobrov_artur@vpsj3IeQ ~]$
```

Рис. 3.3

```
belobrov_artur@vpsj3IeQ:~/P/PuP>PePeP"P PsPiP"C,P"PfCfPrPsP±P"PfCfCНьPsPiPsP"PpP=C=Pl
[belobrov_artur@vpsj3IeQ ~]$ pushd ВеликаРогатаХудобаУськогоГолів/
~/ВеликаРогатаХудобаУськогоГолів ~
[belobrov_artur@vpsj3IeQ ВеликаРогатаХудобаУськогоГолів]$ ls
1529 183 1900 223 2707 317 354 399 414 443 76 900
[belobrov_artur@vpsj3IeQ ВеликаРогатаХудобаУськогоГолів]$
```

Рис. 3.4

Висновки: в ході виконання лабораторної роботи, були придбані навички автоматизації керування ОС з використанням скриптової мови інтерпретатору оболонки командного рядку. Найскладнішою частиною було налаштування перевірок назв файлів та їх інтерпретації для 3го завдання.