

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:
Студент групи АІ-203
Токарєв О.С.

Одеса 2021

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

2 Завдання до виконання

Завдання 2.1

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 3.

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.

Таблиця 3 – Типи файлів та окремі параметри за варіантами завдань

№ команди	№ учасника команди	1-й параметр пам'яті	2-й параметр пам'яті	Параметр процесору
1	1	<i>MemTotal</i>	<i>Shmem</i>	<i>vendor_id</i>
1	2	<i>MemFree</i>	<i>Slab</i>	<i>cpu family</i>
2	1	<i>MemAvailable</i>	<i>SReclaimable</i>	<i>model</i>
2	2	<i>Buffers</i>	<i>SUnreclaim</i>	<i>model name</i>
3	1	<i>Cached</i>	<i>KernelStack</i>	<i>stepping</i>
3	2	<i>SwapCached</i>	<i>PageTables</i>	<i>microcode</i>
4	1	<i>Active</i>	<i>NFS_Unstable</i>	<i>cpu MHz</i>
4	2	<i>Inactive</i>	<i>Bounce</i>	<i>cache size</i>
5	1	<i>Active(anon)</i>	<i>WritebackTmp</i>	<i>siblings</i>
5	2	<i>Inactive(anon)</i>	<i>CommitLimit</i>	<i>cpu cores</i>
6	1	<i>Active(file)</i>	<i>Committed_AS</i>	<i>apicid</i>
6	2	<i>Inactive(file)</i>	<i>VmallocTotal</i>	<i>initial apicid</i>
7	1	<i>Unevictable</i>	<i>VmallocUsed</i>	<i>fpu</i>
7	2	<i>Mlocked</i>	<i>VmallocChunk</i>	<i>fpu_exception</i>
8	1	<i>SwapTotal</i>	<i>Percpu</i>	<i>cpuid level</i>
8	2	<i>SwapFree</i>	<i>HardwareCorrupted</i>	<i>wp</i>
9	1	<i>Dirty</i>	<i>AnonHugePages</i>	<i>bogomips</i>
9	2	<i>Writeback</i>	<i>CmaTotal</i>	<i>clflush size</i>
10	1	<i>AnonPages</i>	<i>CmaFree</i>	<i>cache_alignment</i>
10	2	<i>Mapped</i>	<i>HugePages_Total</i>	<i>address sizes</i>

Завдання 2.2 Розробіть програму на мові *BASH*, яка виконує дії у відповідності з варіантом, представленим у таблиці 4:

- *my_create_file* – створити файл;
- *my_create_directory* – створити каталог (команда *mkdir*);
- *my_change_directory* – змінити назву каталогу (команда *mv*);
- *my_change_file* – змінити назву файлу (команда *mv*);
- *my_delete_file* – видалити файл (команда *rm*);
- *my_delete_directory* – видалити каталог (команда *rmdir*).

4	2	<i>my_create_directory</i>	Не більше 28	Не більше 3 цифр підряд
---	---	----------------------------	--------------	-------------------------

Для всіх варіантів передбачається наступний опис кроків алгоритму роботи програми:

- 1) запропонувати ввести назву об'єкту, використовуючи українське запрошення;
- 2) перевірити присутність об'єкту (для команд видалення або зміни) або відсутність об'єкту (для команд створення) з такою назвою у файловій підсистемі, використовуючи відповідні команди та файли;
- 3) якщо об'єкт відсутній (для команд видалення або зміни) або присутній (для команд створення), 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.

Для перевірки правильності роботи програми необхідно:

- для команд видалення або зміни - попередньо створити об'єкт, виконати програму, а потім ще раз її виконати, що приведе до помилки;
- для команд створення - попередньо видалити об'єкт, виконати програму, а потім ще раз її виконати, що приведе до помилки.

Завдання 2.3 В попередній лабораторній роботі ви створили файл CSV-формату за пунктом 2.2.3

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.

1. Створюємо (на скріншоті файл вже створено) та відкриваємо файл скрипту:

```
login as: tokarev_oleksandr
tokarev_oleksandr@91.219.60.189's password:
Last login: Sun Mar 28 17:53:23 2021 from 46.185.84.233
[tokarev_oleksandr@vpsj3IeQ ~]$ vim myosparam.sh
```

Пишемо та зберігаємо скрипт:

```
#!/bin/bash
readarray -t memlist < /proc/meminfo

for (( i = 0; i < ${#memlist[@]}; i++))
do
    if [[ ${memlist[$i]} =~ ^Inactive: ]]; then
        [[ ${memlist[$i]} =~ [^0-9]+([0-9]+) ]]
        echo "Inactive memory: ${BASH_REMATCH[1]} kB"
    fi
    if [[ ${memlist[$i]} =~ ^Bounce: ]]; then
        [[ ${memlist[$i]} =~ [^0-9]+([0-9]+) ]]
        echo "Bounce: ${BASH_REMATCH[1]} kB"
    fi
done
readarray -t cpulist < /proc/cpuinfo

x=1
for (( i = 0; i < ${#cpulist[@]}; i++))
do
    if [[ ${cpulist[$i]} =~ cache ]]; then
        [[ ${cpulist[$i]} =~ [^0-9]+([0-9]+) ]]
        echo "Cache size $x: ${BASH_REMATCH[1]} KB"
        (( x++ ))
    fi
done
```

Тестуємо його:

```
[tokarev_oleksandr@vpsj3IeQ ~]$ sh myosparam.sh
Inactive memory: 866152 kB
Bounce: 0 kB
Cache size 1: 30720 KB
Cache size 2: 64 KB
Cache size 3: 30720 KB
Cache size 4: 64 KB
```

2. Створюємо (файл вже створено) та відкриваємо файл скрипту; пишемо та зберігаємо скрипт:

```
#!/bin/bash
echo "Введите название директории: "
read directoryname
if [[ -f ${directoryname} ]]; then
    echo "Ошибка! Директория уже существует."
else
    if [[ ${#directoryname} > 28 ]]; then
        echo "Ошибка! Название директории превышает лимит в 28 символов."
    else
        if [[ ${directoryname} =~ [0-9]{3,} ]]; then
            echo "Ошибка! Название в названии может быть не больше 3 цифр подряд."
        else
            mkdir $directoryname
            echo "Директорию ${directoryname} создано!"
        fi
    fi
fi
```

Тестуємо його:

```
[tokarev_oleksandr@vpsj3IeQ ~]$ sh my_create_directory.sh
Введите название директории:
zxsczxczxczxczxczxczxczxczxc
Ошибка! Название директории превышает лимит в 28 символов.
```

```
[tokarev_oleksandr@vpsj3IeQ ~]$ sh my_create_directory.sh
Введите название директории:
11111
Ошибка! Название в названии может быть не больше 3 цифр подряд.
```

```
[tokarev_oleksandr@vpsj3IeQ ~]$ sh my_create_directory.sh
Введите название директории:
dr
Директорию dr создано!
```

3. Створюємо (на скріншоті файл вже створено) та відкриваємо файл скрипту:

```
[tokarev_oleksandr@vpsj3IeQ ~]$ touch csv3.sh
```

```
[tokarev_oleksandr@vpsj3IeQ ~]$ vim csv3.sh
```

Пишемо та зберігаємо скрипт:

```
#!/bin/bash
count=0
while read line ;
do
    IFS=","
    set -- $line
    for i in $3
    do
        if [[ $count = 0 ]]; then
            count=1
            mkdir $i
            dirname=$i
        else
            touch "$dirname/$i"
        fi
    done
done < ./tokarev_4_2.csv
```

Тестуємо його:

```
[tokarev_oleksandr@vpsj3ie0 ~]$ sh csv3.sh
[tokarev_oleksandr@vpsj3ie0 ~]$ ls
accounts.csv  dr                  my_create_directory.sh  Operating-System.-Laboratory-Work-1.  os.lab1.utf.html  tokarev_4_2.csv  Tokarev.html  Tokarev.pdf
csv3.sh       my_create_directory myosparam.sh            os.lab1.cpl251.html                  tokarev_4_2_2.csv Tokarev.doc      tokarev_lab_3  всього
[tokarev_oleksandr@vpsj3ie0 ~]$ cd всього
[tokarev_oleksandr@vpsj3ie0 всього]$ ls
10985 13123 1380 16246 1764 1982 2218 2302 2607 2865 3089 3282 3842 4308 464772 5057 5504 9242
1154 1359 14556 1707 17659 1989 2299 23059 273720 2997 3166 3444 4149 4423 5034 5474 6513
```

Висновок: Під час лабораторної роботи ми придбали навички автоматизації керування ОС з використанням скриптової мови інтерпретатору оболонки командного рядку.