

Одесский национальный политехнический университет
Институт компьютерных систем
Кафедра информационных систем

Лабораторная работа № 5

По дисциплине: «Операционные системы»

Тема: «Основы использование скриптового языка интерпретатора оболочки
командной строки »

Выполнила:
Студентка группы АИ-205
Колдунова А. А.
Проверил:
Блажко О.А.

Цель работы: приобретение навыков автоматизации управления ОС с использованием скриптового языка интерпретатора оболочки командной строки.

План работы.

1 Теоретические сведения

1.1 Расширенный функционал команд оболочки ОС интерфейса командной строки и утилит командной строки

1.2 Редактирование каталогов и файлов файловой системы

1.3 Перенаправление потоков данных

1.4 Конвейеризация команд

1.5 Команды оболочки и

1.5 Команды после инициализации командной строки не обрабатываются

1.6 Конфигурация работы оболочки 2. Задание к исполнению

2. Задание к исполнению

3 Требования к оформлению протокола выполнения лабораторной работы

Решение:

Задание 2.1

```
[koldunova_anastasiya@vpsj3IeQ ~]$ chmod +x MyOsProgram.sh
[koldunova_anastasiya@vpsj3IeQ ~]$ ./MyOsProgram.sh
[Перший параметр:
Inactive(anon)
[Параметр:
Опис: Загальний обсяг буфера або сторінок кеш-пам'яті, який є вільним і доступний. Це пам'ять, яка не була недавно використана і може бути звільнена для інших цілей за допомогою алгоритму підкачки.
Inactive(anon): 436404 kB
Другий параметр:
CommitLimit
[Параметр:
Опис: Обсяг пам'яті, який може бути виділений системою. Обчислюється на основі vm.overcommit_ratio (за замовчуванням - 50%) і розміру області підкачки.
Формула має такий вигляд CommitLimit = (vm.overcommit_ratio * об'єм_ОЗУ) + області_підкачки. Цей ліміт дотримується тільки при "суворої" політиці виділення пам'яті (vm.overcommit_memory = 2). За замовчуванням використовується "евристична" політика (vm.overcommit_memory = 0).
CommitLimit: 5135228 kB
Третій параметр:
[Параметр:
Core (вимірюється приблизно: До [р]) - торгові марки мікропроцесорів, вироблених компанією Intel. Процесори Core є наступниками процесорів попереднього покоління, представленіх моделями Pentium і Celeron. Для серверів є більш «просунуті» версії процесорів Core під маркою Xeon.
grep: cores: No such file or directory
```

Задание 2.2

```
[[koldunova_anastasiya@vpsj3IeQ ~]$ ls
1.csv 3.csv           MyOsProgram.sh      Operating-System.-Laboratory-Work-  file
2.csv KoldunovaAnastasiaLab4.csv  My_change_file.sh  accounts.csv                  koldunovaN_lab_3
[[koldunova_anastasiya@vpsj3IeQ ~]$ ./My_change_file.sh
Введіть назву файлу:
|file
```

```
[[koldunova_anastasiya@vpsj3IeQ ~]$ ls
1.csv 3.csv           MyOsProgram.sh      Operating-System.-Laboratory-Work-  koldunovaN_lab_3
2.csv KoldunovaAnastasiaLab4.csv  My_change_file.sh  accounts.csv                  my_new_file
[koldunova_anastasiya@vpsj3IeQ ~]$
```

```

#!/bin/bash
echo "?^?вед?^??^??^? назв?^? ?^?айл?^?: "
read my_file
if [ -f $my_file ]
then
if [ ${#VAR} -lt 30 ]
then
if [ grep "\d{5}" $my_file ]
then
echo "?^?е в?^?дпов?^?да?^? обмеженн?^?м"
else
mv $my_file my_new_file
fi
else
echo "?^?е в?^?дпов?^?да?^? обмеженн?^?м"
fi
else
echo "the $my_file д?^??^?ек?^?о?^??^? не ?^??^?н?^??^?"
fi

```

Задание 2.3

```

[koldunova_anastasiya@vpsj3IeQ ~]$ chmod +x lab5.sh
[koldunova_anastasiya@vpsj3IeQ ~]$ ls
1.csv  3.csv          MyOsProgram.sh  Operating-System.-Laboratory-Work- koldunovaN_lab_3  my_new_file
2.csv  KoldunovaAnastasiaLab4.csv  My_change_file.sh  accounts.csv                  lab5.sh
[koldunova_anastasiya@vpsj3IeQ ~]$ ./lab5.sh

```

```

/.ssh
/Operating-System.-Laboratory-Work-
/koldunovaN_lab_3
/?????????? ???????????

```

```

#!/bin/bash
name=$(cut -f3 -d, KoldunovaAnastasiaLab4.csv > new.csv|sed -n '1p; 1q' new.csv);
#echo "$a"
mkdir "$name"
IFS=$'\n' file=($(sed -e '1d' new.csv))
#echo ${files[0]}
cd "$name"
for index in ${!file[*]}
do
touch ${file[$index]}
done

```

/ ..

```
"????????????????????? ?????????~????????? ??????????????????????  
"????????????????????? ??????????????????????????????????????  
"????????????????????? ??????????????????????  
"????????????? ?????????????~????????? (?????????????????  
????????????? ?????????????~????????? ?? 2019 ?????????  
????????????????????????? ???~????????????? 2020 ?????????  
????????????????????????? ???~????????? 2021 ?????????  
????????????????????????? ???~????????? 2020 ?????????  
????????????????????????? ???~????????????? ?????????)  
????????????????????? ???~????????? ?? 2019 ?????????  
????????????????????? ???~????????? ?? 2019 ?????????  
????????????????????? ???~????????? ?? 2019 ?????????  
????????????????????? ???~????????? 2021 ?????????  
????????????????????? ???~????????? ?????????????? ?????????)  
????????????????? ???~????????? ~????????????? 2021 ?????????  
????????????????? ???~????????? ?? 2019 ?????????  
????????????????? ???~????????? ?? 2020 ?????????  
????????????????? ???~????????? ?????????????? ?????????)  
????????????????? ???~????????? ?? 2020 ?????????
```

Вывод: В данной лабораторной мы приобрели навыки автоматизации управления ОС с использованием скриптового языка интерпретатора оболочки командной строки.