

Міністерство освіти України  
Харківський національний університет радіоелектроніки

Звіт до практичної роботи № 3  
з курсу «Операційні системи UNIX»

Виконав: ст. гр. ПЗПІ-16-3

Губар С.О.

Прийняв: Сокорчук І.П.

Харків 2020

**Мета роботи:** Вивчити можливість застосування змінних середовища Bourne-shell (bash): системних, призначених для користувача, позиційних, спеціальних. Ознайомитися з підстановкою змінних.

### **Хід виконання роботи**

Створимо файл pz3.sh, позначимо його як виконуваний та додамо

```
#!/bin/bash
```

Встановимо значення змінних, з яким запускається скрипт \$1 та \$2 для подальшого використання

```
set abc def
```

Скористаємося перепризначенням виводу у файл test.txt (перший раз без доповнення, другий - з доповненням існуючого файлу)

```
echo "Hi pz 3" > test.txt
```

```
echo "Hi once again" >> test.txt
```

Використаємо pipe щоб зробити конвеєр - пошукаємо файли у назві яких є test, вичитаємо усі txt файли - відсортуємо їх контент, знайдемо лише унікальні строки та запишемо результат у result-file.out

```
ls -a | grep "test"
```

```
cat *.txt | sort | uniq > result-file.out
```

Зчитаємо зміст test.txt, перепризначимо вивід у утиліту tee, яка пише у stdout те, що отримала у stdin та перепризначимо stdout у файл outfile.out

```
cat < test.txt | tee > outfile.out
```

Виконаємо команду та запишемо результат виконання у num\_found

```
num_found=$(ls -a | grep "test" | wc -l)
```

Виведемо форматовано num\_found

```
echo $num_found
```

```
echo "Found $num_found"
```

Виведемо значення деяких змінних оточення

```
echo "Some variables"
```

```
echo "Home directory: $HOME"
```

```
echo $PS1
```

```
echo "Path $PATH"
```

Проведемо декілька експериментів з виконанням команд у разі невдачі

```
false && echo "Test"
```

```
# Test
```

```
false || echo "Test"
```

```
# Test Test2
```

```
false || echo "Test" && echo "Test2"
```

Реалізуємо простий цикл, який виводить числа від 1 до 10

```
counter=1
```

```
while [ $counter -le 10 ]
```

```
do
```

```
    echo $counter
```

```
    ((counter++))
```

```
done
```

Перевіримо кількість аргументів (які раніше встановили через set)

```
if test $# -lt 2
```

```
then
```

```
    echo "Two args needed"
```

```
fi
```

Виведемо значення перших двох аргументів. Скористаємося \ для того, щоб екранувати символ \$

```
echo "Arguments: "
```

```
echo "First argument $1"
```

```
echo "First argument \$1"
```

```
echo "First argument $2"
```

Підставимо значення my\_var через {}

```
my_var="test"
```

```
echo "Some interpolation ${my_var}"
```

**Висновки:** У ході виконання практичного заняття було проведено ознайомлення з змінними у bash. Також було вивчено, які бувають змінні оточення, змінні які передаються у скрипт та як їх використовувати.