

Міністерство освіти України
Харківський національний університет радіоелектроніки

Звіт до практичної роботи № 4
з курсу «Операційні системи UNIX»

Виконав: ст. гр. ПЗПІ-16-3

Губар С.О.

Прийняв: Сокорчук І.П.

Харків 2020

Мета роботи: отримати практичні навички написання скриптів за допомогою Bash. Вивчити особливості синтаксису, навчитися використовувати вбудовані команди, конвеєри.

Хід виконання роботи

Створимо файл pz4.sh, зробимо його виконуваним та додамо директиву
#!/bin/bash

В якості прикладу, реалізуємо скрипт, який збирає інформацію про заданий користувач тип файлів, та агрегує зміст файлів у один.

Зчитуємо формат файлу та файл, у який треба записати звіт:

```
echo "Enter file extension:"
read extension
echo "Enter report filename:"
read report
```

Якщо файлів такого формату не знайдено, звіт створювати не треба:

```
num=$(ls -a | grep .${extension}\$ | wc -l)
if [[ $num -eq 0 ]]; then
    echo "No files were found"
    exit 0
fi
```

Запишемо кількість знайдених файлів та дату формування звіту:

```
date_formatted=$(date +%D)
echo "Found $num files" > $report
echo "Report formed at ${date_formatted}" >> $report
echo "======" >> $report
```

За допомогою циклу for пройдемо по усім файлам з заданим типом. Зберемо дату останньої модифікації файла, тип файла, вміст файла, а

також дозволи на зчитування, запис та виконання за допомогою утиліти stat:

```
for f in *.{extension}; do
    echo "Processing $f file..";
    modify_date=$(date -r ${f})
    echo "$f was last modified at ${modify_date}" >> $report
    file_type=$(file ${f})
    echo "$f has type ${file_type}" >> $report
    perm=$(stat -c%a "$f")
    user=${perm:0:1}
    group=${perm:1:1}
    global=${perm:2:1}
    echo "Permissions : " >> $report
    echo "Owner Access: $(print_perm $user)" >> $report
    echo "Group Access: $(print_perm $group)" >> $report
    echo "Others Access: $(print_perm $global)" >> $report
    echo "$f content:" >> $report
    echo "======" >> $report
    cat $f >> $report
done
```

Напишемо функцію, яка форматує rwx з вигляду трьох цифр у читаємий формат:

```
print_perm() {
    case "$1" in
        0) printf "NO PERMISSIONS";;
        1) printf "Execute only";;
        2) printf "Write only";;
        3) printf "Write & execute";;
        4) printf "Read only";;
        5) printf "Read & execute";;
        6) printf "Read & write";;
        7) printf "Read & write & execute";;
    esac
}
```

Висновки: у ході виконання практичного заняття було отримано навички з написання скриптів за допомогою Bash, було написано скрипт, який агрегує дані та зміст про файли з заданим форматом.