



Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

Операційні системи

Лабораторна №4

Виконав:
Студент групи ФБ-82
Козачок Вячеслав
Перевірив:
Кіреєнко О.В.

Варіант 3

1. Написати сценарій для оболонки `bash` згідно таких вимог: Сценарій приймає 3 параметри командного рядка. Перший параметр — ім'я каталогу, в якому (і в підкаталогах якого рекурсивно) треба здійснювати пошук. Другий параметр — шаблон пошуку. За відсутності двох параметрів сценарій повинен виводити рядок, що описує коректний формат виклику сценарію. Третій параметр — необов'язковий — це ім'я файлу, що буде створений.
2. Сценарій шукає у заданому каталозі і його підкаталогах усі файли, які містять послідовність байт, що відповідає шаблону пошуку, і формує їх список. Список повинен містити ім'я файлу, повний шлях до каталогу, тип файлу (див. команду `file`), розмір файлу у байтах, і рядок, що відповідає шаблону.
3. Якщо третій параметр виклику сценарію (ім'я файлу) був заданий, то список має бути виведений у цей файл. Якщо параметр заданий не був — на цьому етапі сценарій повинен запросити ввести його. В обох випадках якщо такий файл для виводу вже існує, сценарій повинен видати запит, що робити далі — переписати вміст файлу, дописати нові дані у кінець файлу, або скасувати операцію. 38Операційні системи – комп'ютерний практикум
4. Сценарій повинен коректно відпрацьовувати помилки, такі як некоректні імена файлів і шаблон пошуку, відсутність заданого каталогу, помилки доступу (зокрема, відсутність права доступу до певних файлів), відсутність жодних файлів, що відповідають заданому шаблону. При цьому сценарій повинен видавати діагностику помилок.

Робота

```
1 #!/bin/bash
2 usage="script -d directory -t template -f filename"
3
4 checkargs() {
5     if [[ $OPTARG =~ ^-[d,t,f]$ ]]
6     then
7         echo "Unknown argument $OPTARG for option $option"
8         exit 1
9     else
10        echo "lol"
11    fi
12 }
13
14 directory=
15 template=
16 filename=
17 while getopts "d:t:f:" option
18 do
19     case "$option" in
20         d) checkargs
21            directory=${OPTARG};;
22         t) checkargs
23            template=${OPTARG};;
24         f) checkargs
25            filename=${OPTARG};;
26     esac
27 done
28
29 if [ -d "$directory" ];
30 then
31     echo "$directory valid"
32 else
33     echo "Such directory does not exists."
34     exit 1
35 fi
36
37 if [[ $directory == "" || $template == "" ]];
38 then
39     echo " [!] Usage: $usage"
```

```

41     exit 1
42 fi
43
44 if [[ $filename == "" ]];
45 then
46     read -p "Please enter the name of the file: "
47     filename=$REPLY
48 fi
49
50 if [[ -f $filename ]];
51 then
52     printf "This file: $filename already exists.\n What do you want to do?\n  [1] Rewrite
53     file\n  [2] Append to file\n  [3] exit\n  Choice:"
54     read
55     if [[ $REPLY == "1" ]]; then
56         rm $filename
57         touch $filename
58     fi
59     if [[ $REPLY == "3" ]]; then
60         exit 1
61     fi
62     touch /tmp/$filename.err
63     touch /tmp/$filename.bin
64 fi
65
66 printf '\n'
67
68 grep -nrwi $directory -e "$template" 2> /tmp/$filename.err > "/tmp/$filename.tmp"
69
70 while IFS="" read -r p || [ -n "$p" ]
71 do
72     if [[ "$p" =~ .*"matches"*. ]];
73     then
74         echo "$p" > "/tmp/$filename.bin"
75     fi
76 done < /tmp/$filename.tmp
77
78 sed -i -e '/matches/d' /tmp/$filename.tmp
79
80 while IFS="" read -r p || [ -n "$p" ]
81 do
82     name=$(cut -d: -f1 <<< "$p")
83     text=$(cut -d: -f3 <<< "$p")
84     echo "$((basename "$name")):$((realpath $name)):$((file $name | cut -d: -f2):bytes $(wc -c
85     $name | cut -d" " -f1):text $text" >> $filename
86 done < "/tmp/$filename.tmp"
87
88 while IFS="" read -r p || [ -n "$p" ]
89 do
90     name=$(cut -d" " -f3 <<< "$p")
91     echo "$((basename "$name")):$((realpath $name)):$((file $name | cut -d: -f2):bytes $(wc -c
92     $name | cut -d" " -f1):binary file" >> $filename
93 done < /tmp/$filename.bin
94
95 rm "/tmp/$filename.tmp"
96 rm "/tmp/$filename.err"
97 rm "/tmp/$filename.bin"

```

Висновки

Навчився писати скрипти для bash. Оволодів практичними навичками професійної роботи з командною оболонкою shell – використанням змінних і створенням командних файлів.