

Лабораторная работа 11

Генералов Даниил, НПИ-01-21, 10322122800

20 May, 2022

¹RUDN University, Moscow, Russian Federation

Цель и задачи

Изучить основы программирования в оболочке ОС UNIX. Научится писать более сложные командные файлы с использованием логических управляющих конструкций и циклов.

Требуется написать 4 командных файла:

- принимает аргументы с командной строки и использует их для запуска **grep**
- запускает другую программу и получает ее код выхода
- создает файлы с названиями, имеющими номер в последовательности
- создает архив из файлов, которые обновлены меньше чем неделю назад

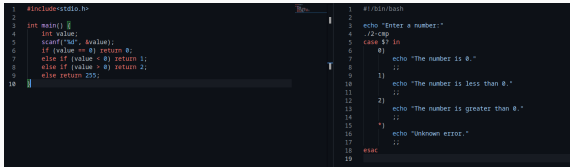
Выполнение работы

Программа для запуска grep

```
1 #!/bin/bash
2
3 while getopts "i:op:Cn" opt; do
4     case $opt in
5         i)
6             input_file=$OPTARG
7             ;;
8         o)
9             output_file=$OPTARG
10            ;;
11         p)
12             pattern=$OPTARG
13             ;;
14         C)
15             case_insensitive=1
16             ;;
17         n)
18             print_line_numbers=1
19             ;;
20         *)
21             echo "Invalid option: -$OPTARG" >&2
22             exit 1
23             ;;
24         )
25             echo "Option -$OPTARG requires an argument." >&2
26             exit 1
27             ;;
28     esac
29 done
30
31 if [ -z "$input_file" ] || [ -z "$output_file" ] || [ -z "$pattern" ]; then
32     echo "Usage: $0 -i <input_file> -o <output_file> -p <pattern> [-C] [-n]"
33     exit 1
34 fi
35
36 grep ${if [ -n "$case_insensitive" ]; then echo "-i"; fi} ${if [ -n "$print_line_numbers" ]; then printf "%s -n"; fi} "$pattern" "$input_file" > "$output_file"
```

Figure 1: Программа для запуска grep

Программа для проверки кода выхода программы



```
1 #include<stdio.h>
2
3 int main()
4 {
5     int value;
6     scanf("%d", &value);
7     if (value == 0) return 0;
8     else if (value < 0) return 1;
9     else if (value > 0) return 2;
10    else return 255;
11 }
```

```
1 #!/bin/bash
2
3 echo "Enter a number:"
4 ./2-cmp
5 case $? in
6     0) echo "The number is 0."
7         ;;
8     1) echo "The number is less than 0."
9         ;;
10    2) echo "The number is greater than 0."
11        ;;
12    *) echo "Unknown error."
13        ;;
14 esac
15
16
17
18
19
```

Figure 2: Программа для проверки кода выхода программы

Программа для создания файлов

```
1  #!/bin/bash
2
3  if [ -z "$1" ]; then
4      echo "Usage: $0 <number>"
5      exit 1
6  fi
7
8  rm -f *.tmp
9
10 for name in $(seq 1 $1); do
11     touch "$name.tmp"
12 done
13
```

Figure 3: Программа для создания файлов

Программа для создания архива

```
1  #!/bin/bash
2
3  rm -f archived.tar
4
5  for file in $(find $1 -type f -mtime -7); do
6      echo $file
7      if [ -f archived.tar ]; then
8          tar rf archived.tar "$file" 2> /dev/null
9      else
10         tar cf archived.tar "$file" 2> /dev/null
11     fi
12 done
```

Figure 4: Программа для создания архива

Заключение

Командные файлы – это удобный способ писать программы в той же самой среде, в которой работает пользователь.

Используя четыре приведенных примера, можно комбинировать и решать более сложные задачи, не используя других языков программирования.