

Российский университет дружбы народов

Факультет физико-математических и естественных наук

Лабораторная работа № 11. Программирование в командном процессоре ОС UNIX. Ветвления и циклы

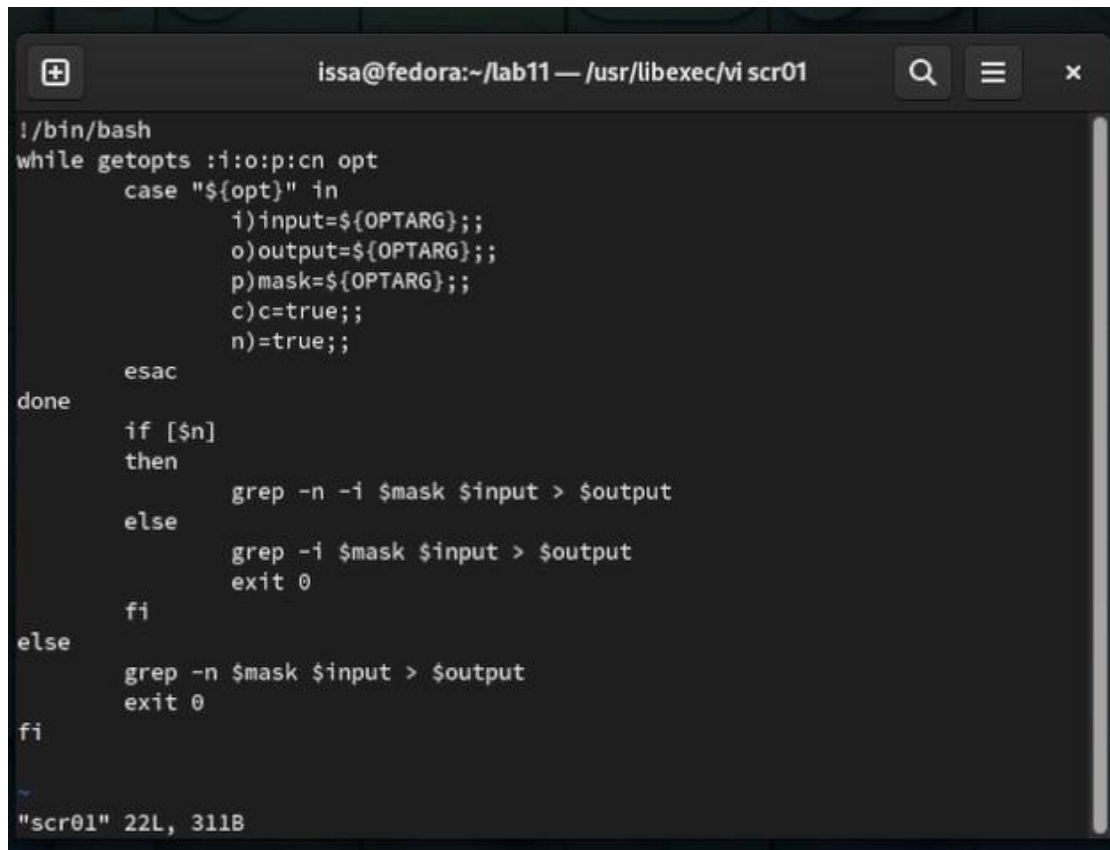
-
- Имя : исса гадир
 - Студенческий билет : 1032218267
 - Группа : нфибд-01-21

Цель работы

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

Чтение и вывод файлов

Сначала установите код условного выполнения , а затем напишите оператор условного выполнения , чтобы он мог выполнять различные потребности , требуемые пользователем при различных условиях

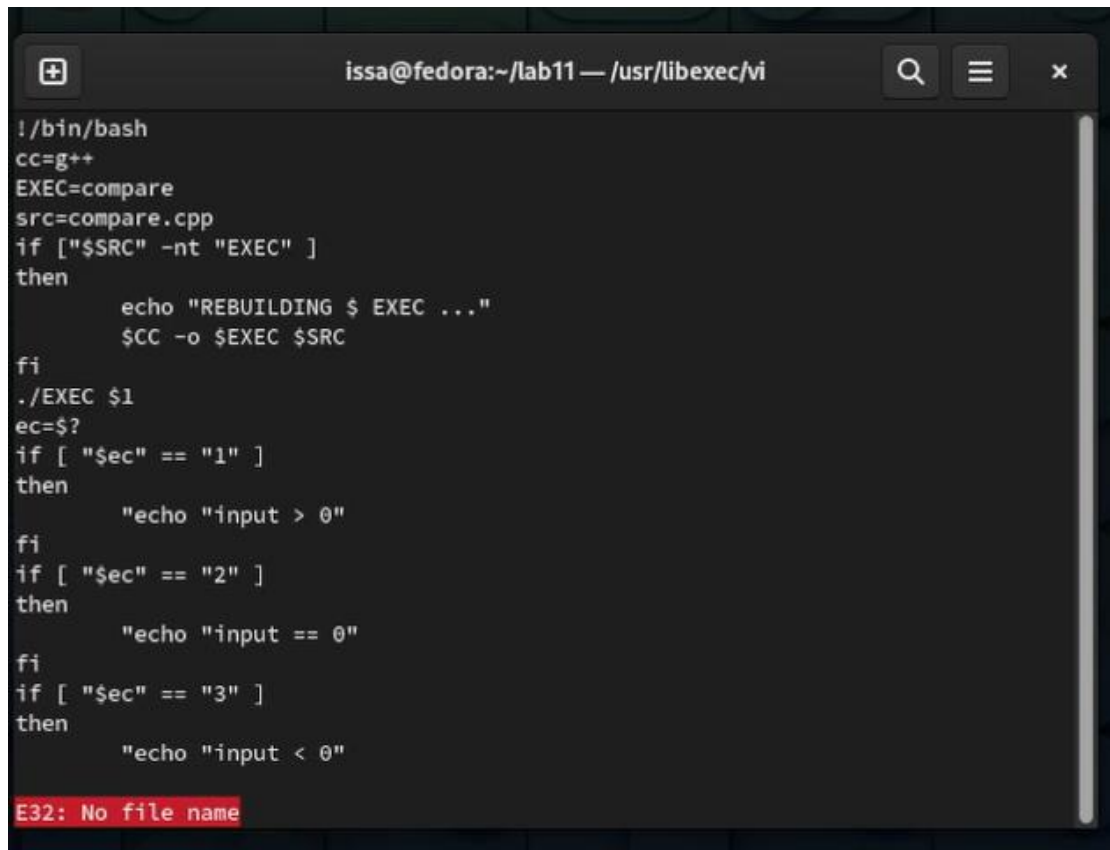
A screenshot of a terminal window with a dark background. The window title bar shows the user 'issa' at 'fedora' in the directory '~/lab11', editing a file named 'scr01' with the 'vi' editor. The terminal content shows a shell script starting with '!/bin/bash'. It uses a 'while' loop with 'getopts' to parse options: '-i' for input, '-o' for output, '-p' for mask, '-c' for a flag, and '-n' for another flag. Inside the loop, a 'case' statement handles each option by assigning values to variables or setting flags. After the loop, an 'if' statement checks if the '-n' flag was set. If so, it runs 'grep -n -i \$mask \$input > \$output'. Otherwise, it runs 'grep -i \$mask \$input > \$output' and then 'exit 0'. The script ends with 'fi' and 'exit 0'. At the bottom, the terminal status shows '"scr01" 22L, 311B'.

```
!/bin/bash
while getopts :i:o:p:cn opt
do
    case "${opt}" in
        i)input=${OPTARG};;
        o)output=${OPTARG};;
        p)mask=${OPTARG};;
        c)c=true;;
        n)=true;;
    esac
done
if [$n]
then
    grep -n -i $mask $input > $output
else
    grep -i $mask $input > $output
    exit 0
fi
else
    grep -n $mask $input > $output
    exit 0
fi

"scr01" 22L, 311B
```

Вызывающие функции и условное суждение

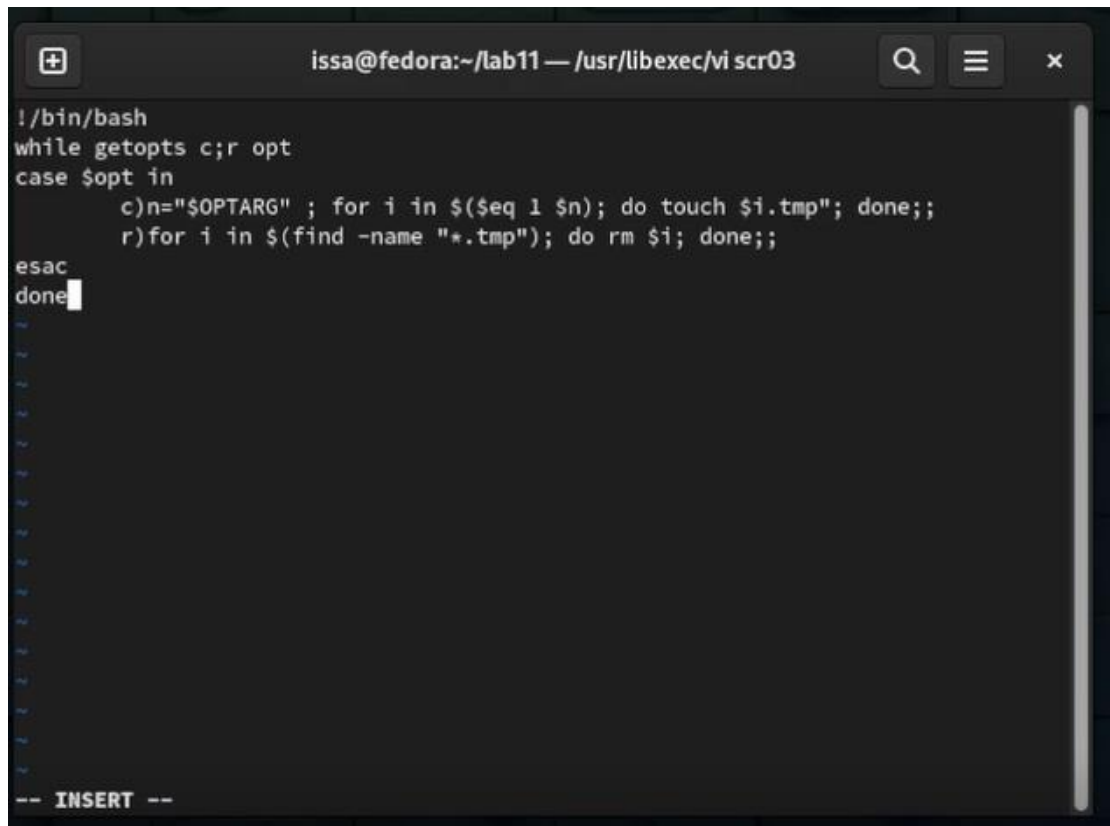
Сначала напиши.файл scrp , чтобы добиться оценки чисел , а затем использовать метод вызова функций для достижения

A terminal window with a dark background. The title bar shows the user 'issa' at 'fedora' in the directory '~/lab11', editing the file '/usr/libexec/vi'. The terminal contains a shell script with variables for compiler, source file, and executable, followed by conditional logic for rebuilding and input validation. At the bottom, a red error message 'E32: No file name' is displayed.

```
! /bin/bash
cc=g++
EXEC=compare
src=compare.cpp
if [ "$SRC" -nt "EXEC" ]
then
    echo "REBUILDING $ EXEC ..."
    $CC -o $EXEC $SRC
fi
./EXEC $1
ec=$?
if [ "$ec" == "1" ]
then
    "echo "input > 0"
fi
if [ "$ec" == "2" ]
then
    "echo "input == 0"
fi
if [ "$ec" == "3" ]
then
    "echo "input < 0"
E32: No file name
```

Пакетный файл

Я использую оператор `while`, и он может выполняться постоянно, если выполняются условия

A screenshot of a terminal window with a dark background. The window title bar shows the user 'issa' on a 'fedora' machine, in the directory '~/lab11', using the 'libexec/vi' editor, editing the file 'scr03'. The terminal content shows a shell script starting with '!/bin/bash', followed by a 'while' loop using 'getopts' to process command-line options. Inside the loop, a 'case' statement handles the options: 'c' sets a variable 'n' and runs a 'for' loop to create temporary files; 'r' runs a 'for' loop to remove them. The script ends with 'esac' and 'done'. The bottom of the screen shows the editor's status line '-- INSERT --'.

```
!/bin/bash
while getopts c;r opt
case $opt in
    c)n="$OPTARG" ; for i in $(seq 1 $n); do touch $i.tmp; done;;
    r)for i in $(find -name "*.tmp"); do rm $i; done;;
esac
done
```

команда tar

По – прежнему используйте условный оператор while , за которым следует условие выполнения tar и шаблон предложения

