

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ по лабораторной работе № 10

дисциплина:Операционные системы

- Студент: Гань Чжаолун
- Группа:НФИБД-01-21
- № ст. билета: 1032198038

МОСКВА 2022 г.

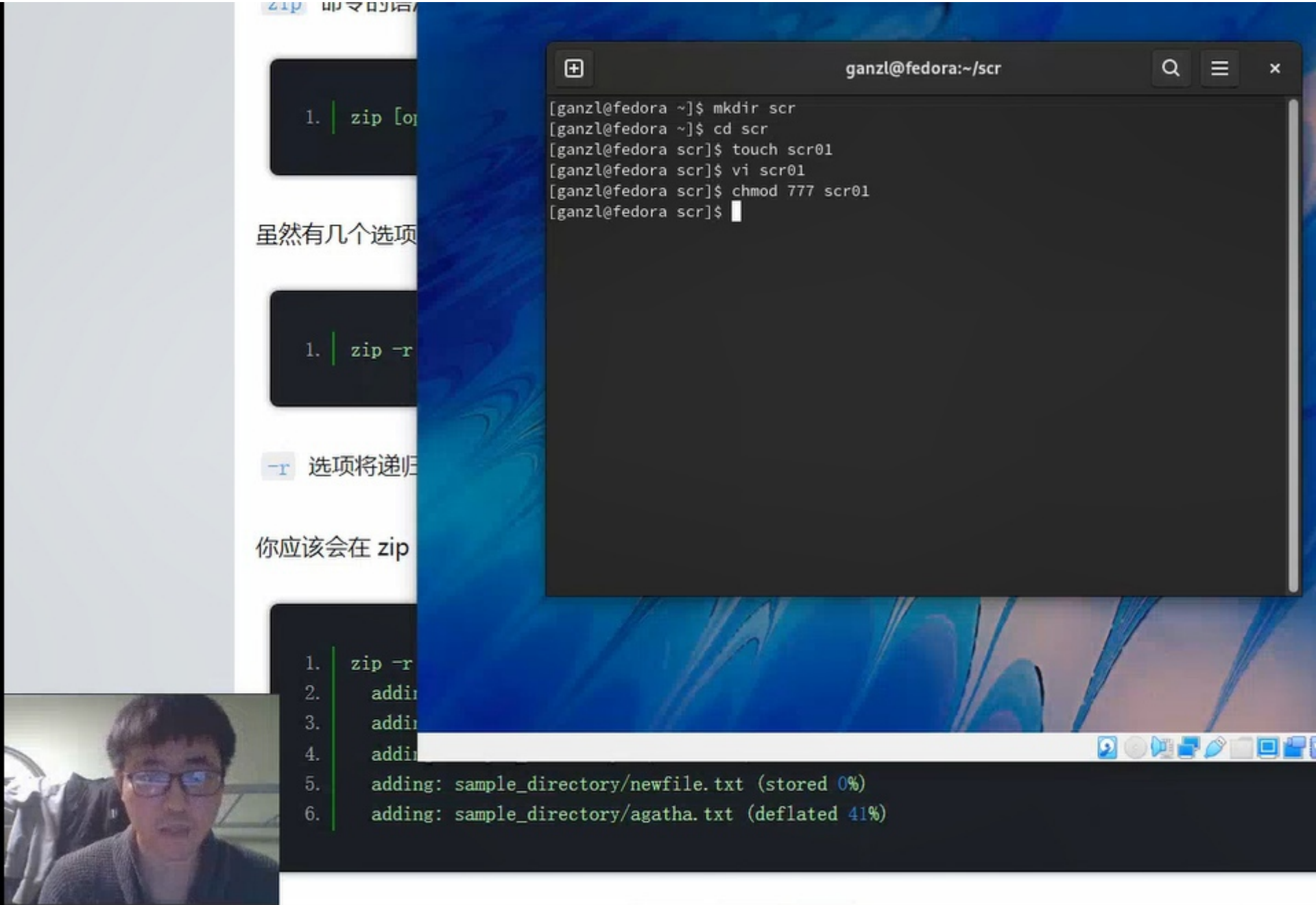
## Цель работы:

Изучить основы программирования в оболочке ОС UNIX/Linux. Научиться писать небольшие командные файлы.

## Выполнение работы:

**1. Написать скрипт, который при запуске будет делать резервную копию самого себя (то есть файла, в котором содержится его исходный код) в другую директорию *backup* в вашем домашнем каталоге. При этом файл должен архивироваться одним из архиваторов на выбор *zip*, *bzip2* или *tar*. Способ использования команд архивации необходимо узнать, изучив справку.**

Сначала я создаю каталог *scr*, создаю файл *scr01* в этом каталоге и устанавливаю разрешения на выполнение



Затем я использую команду vi для редактирования файла, как показано ниже

1. | zip [o

虽然有几个选项

1. | zip -r

-r 选项将递归

你应该会在 zip

1. | zip -r

2. | addi

3. | addi

4. | addi

5. | adding: sample\_directory/newfile.txt (stored 0%)

6. | adding: sample\_directory/agatha.txt (deflated 41%)

ganzl@fedora:~/scr — /usr/bin/vim scr01

zip scr01.zip scr01  
mv scr01.zip /home/ganzl/scr/backup

-- INSERT --

2,36 All

Наконец, я дал разрешение на выполнение файла, выполнил его, а затем проверил, был ли он успешно выполнен.

zip 命令的语

虽然有几个选项

1. | zip -r

-r 选项将递归

你应该会在 zip

1. | zip -r

2. | addi

3. | addi

4. | addi

5. | adding: sample\_directory/newfile.txt (stored 0%)

6. | adding: sample\_directory/agatha.txt (deflated 41%)

Activities Terminal May 20 21:37

ganzl@fedora:~/scr/backup

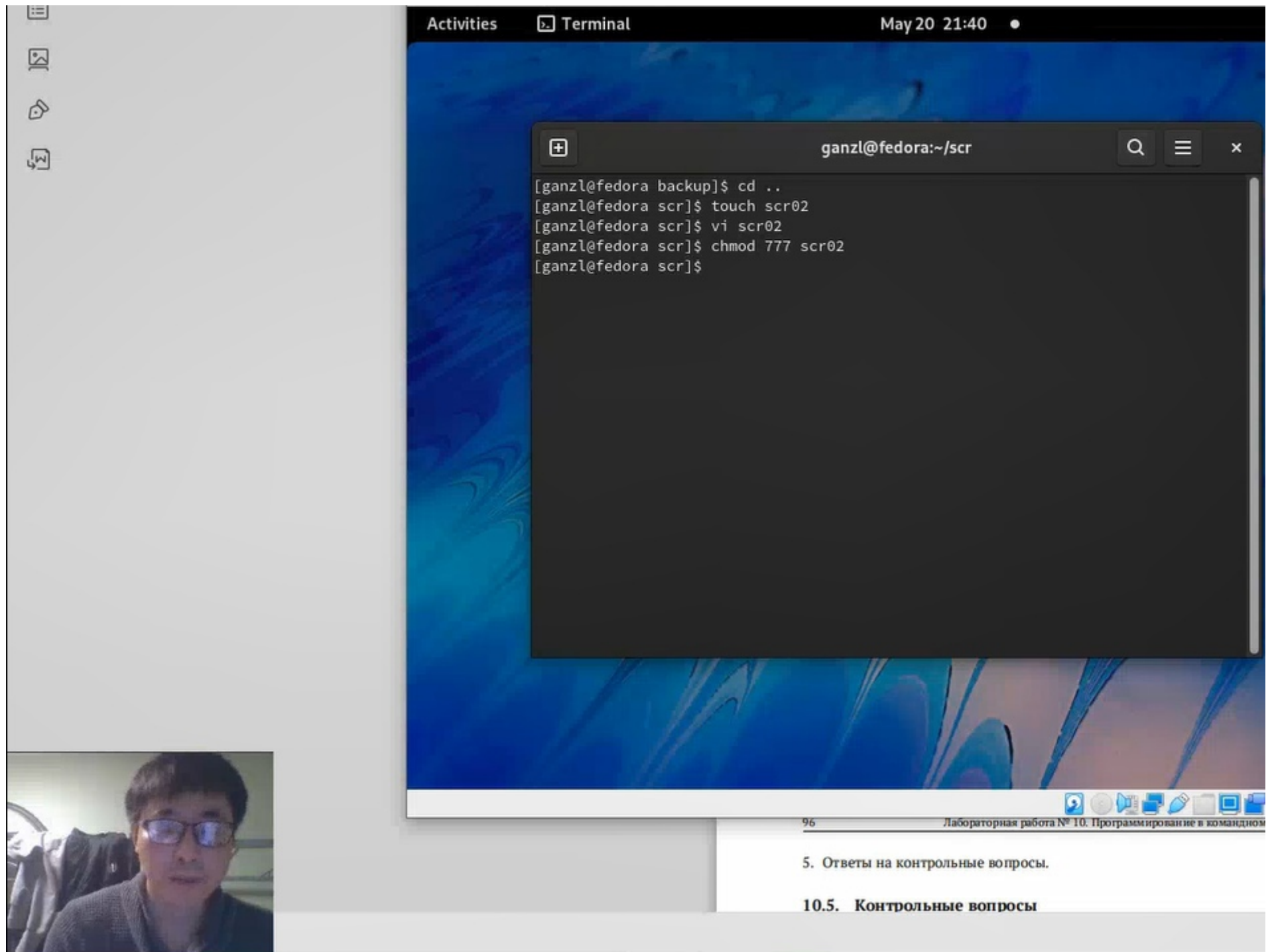
[ganzl@fedora ~]\$ mkdir scr  
[ganzl@fedora ~]\$ cd scr  
[ganzl@fedora scr]\$ touch scr01  
[ganzl@fedora scr]\$ vi scr01  
[ganzl@fedora scr]\$ chmod 777 scr01  
[ganzl@fedora scr]\$ mkdir backup  
[ganzl@fedora scr]\$ ./scr01  
adding: scr01 (deflated 27%)  
[ganzl@fedora scr]\$ ls  
backup scr01  
[ganzl@fedora scr]\$ cd backup  
[ganzl@fedora backup]\$ ls  
scr01.zip  
[ganzl@fedora backup]\$



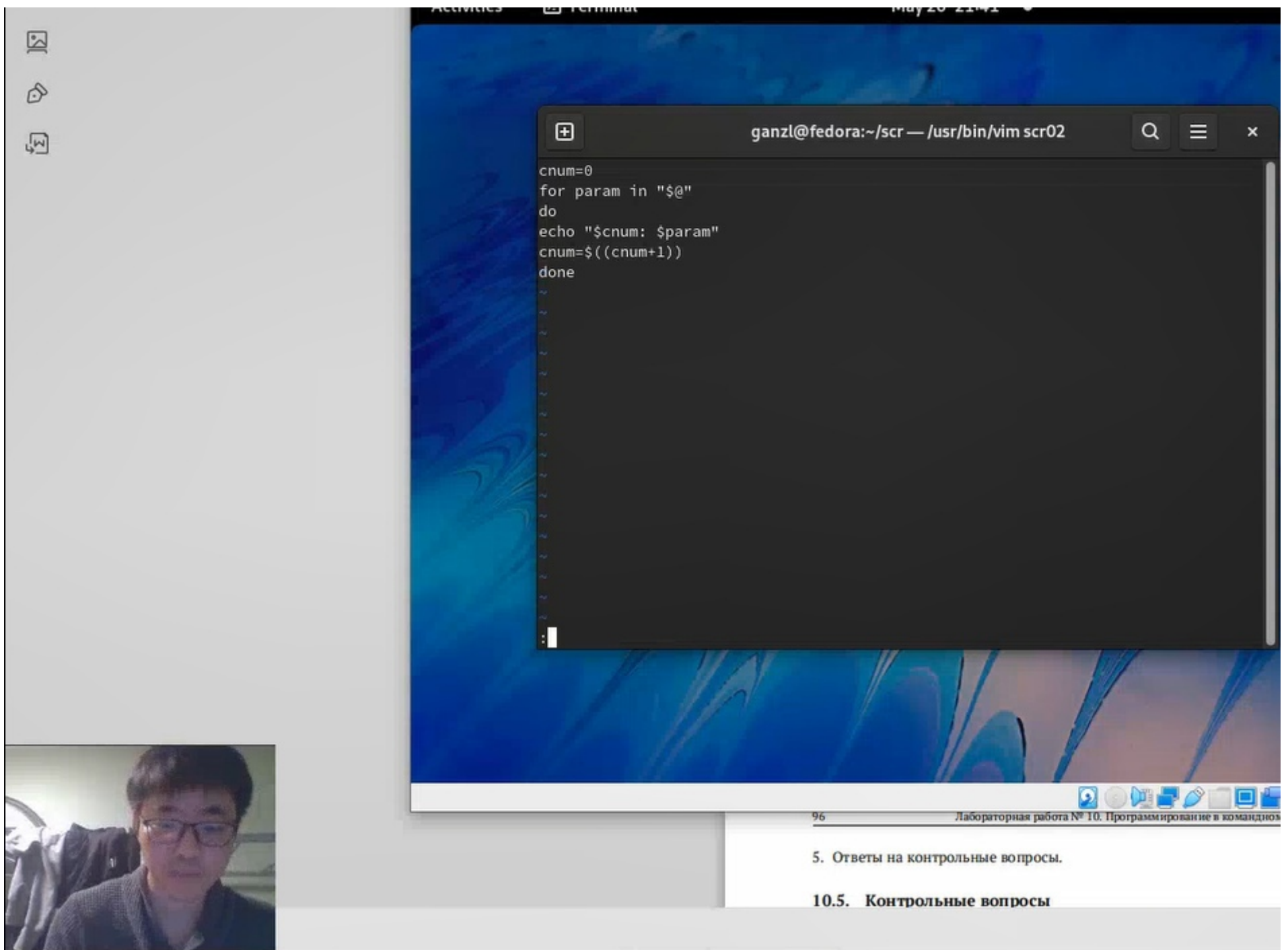
**2. Написать пример командного файла, обрабатывающего любое произвольное число аргументов командной строки, в том числе превышающее десять. Например, скрипт может последовательно распечатывать значения всех переданных аргументов.**

---

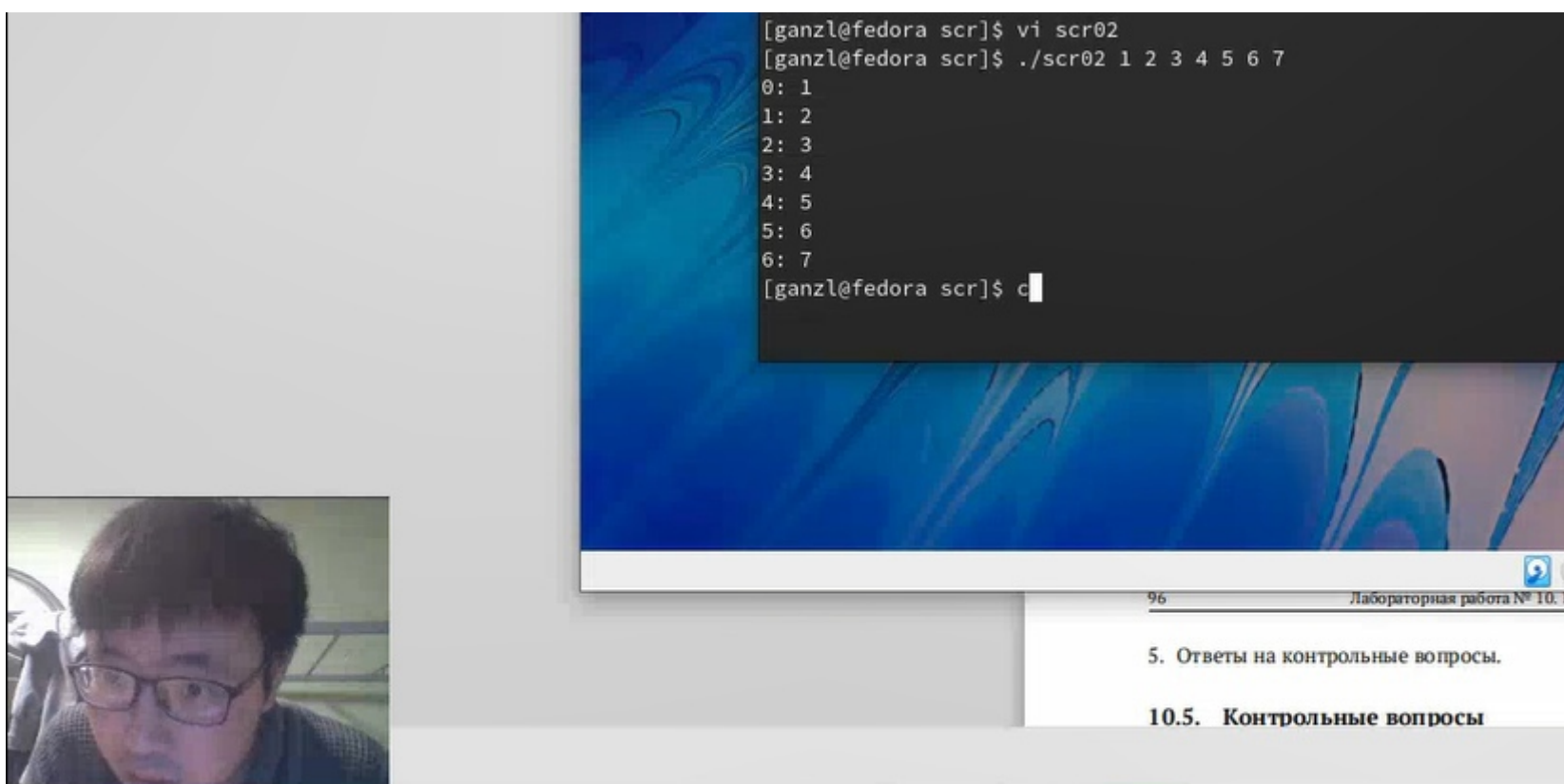
Сначала я создаю файл scr02 и устанавливаю разрешения на выполнение



Затем я использую команду vi для редактирования файла, как показано ниже



Наконец я запускаю файл и проверяю, правильно ли это сделано

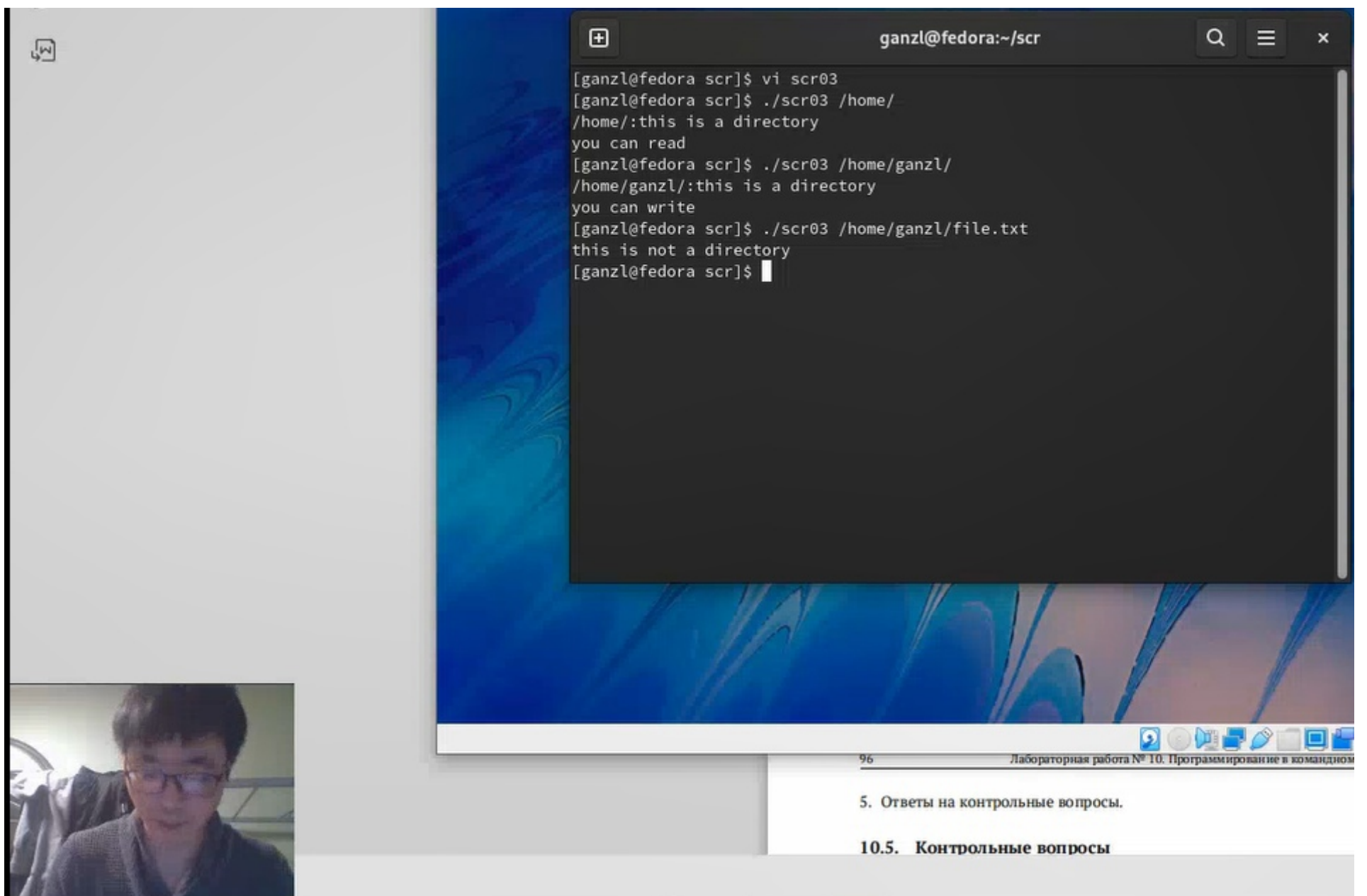


**3. Написать командный файл — аналог команды *ls* (без использования самой этой команды и команды *dir*). Требуется, чтобы он выдавал информацию о нужном каталоге и выводил информацию о возможностях доступа к файлам этого каталога.**

Сначала я создаю файл scr03 и устанавливаю разрешения на выполнение







The screenshot shows a video lecture interface. On the left is a small video feed of a man with glasses. The main area displays a terminal window titled 'ganzl@fedora:~/scr' with the following commands and output:

```
[ganzl@fedora scr]$ vi scr03
[ganzl@fedora scr]$ ./scr03 /home/
/home/:this is a directory
you can read
[ganzl@fedora scr]$ ./scr03 /home/ganzl/
/home/ganzl/:this is a directory
you can write
[ganzl@fedora scr]$ ./scr03 /home/ganzl/file.txt
this is not a directory
[ganzl@fedora scr]$
```

Below the terminal, a document titled 'Лабораторная работа № 10. Программирование в командном процессоре' is visible, showing the text:

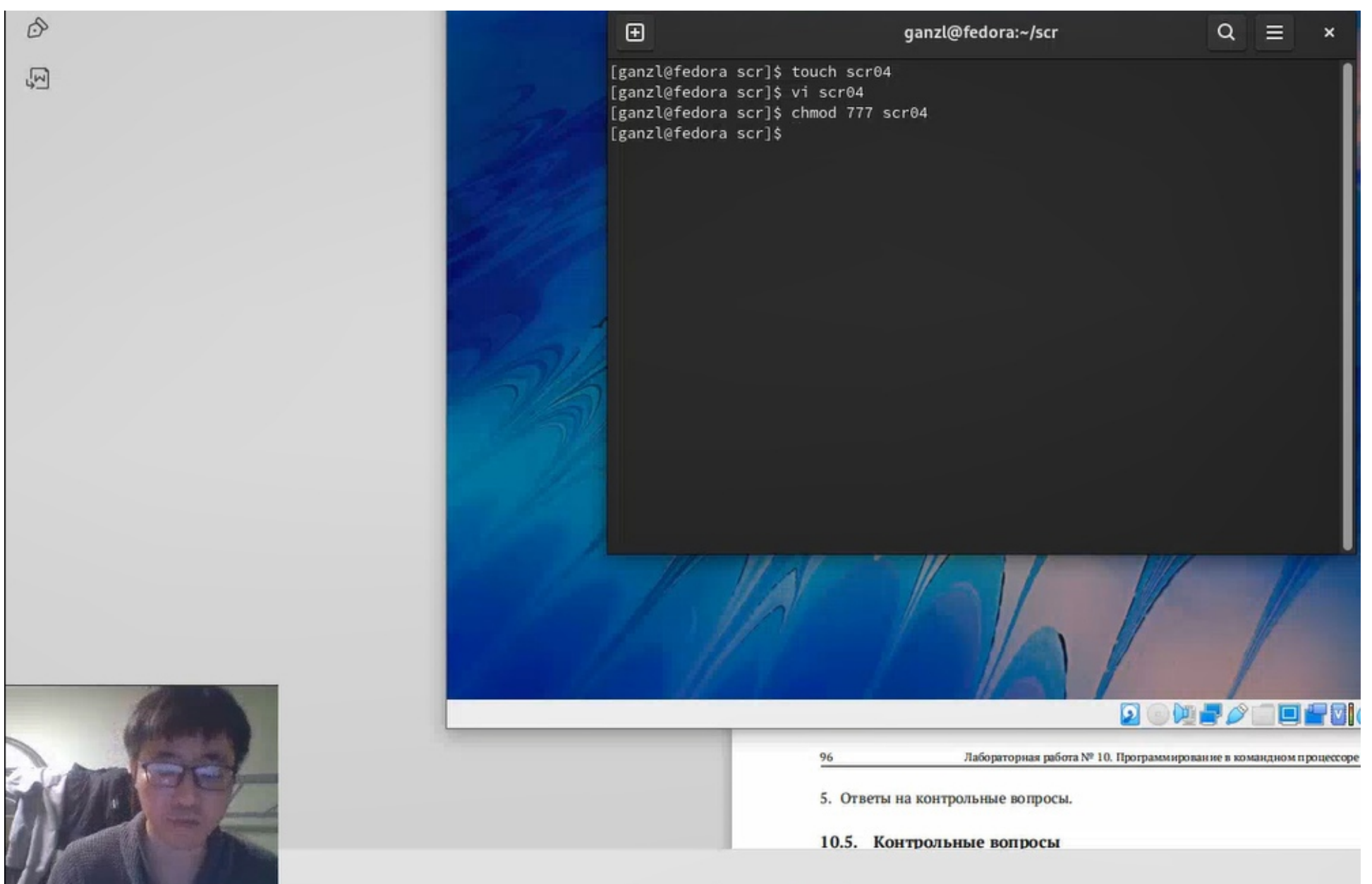
96

5. Ответы на контрольные вопросы.

10.5. Контрольные вопросы

**4. Написать командный файл, который получает в качестве аргумента командной строки формат файла (.txt, .doc, .jpg, .pdf и т.д.) и вычисляет количество таких файлов в указанной директории. Путь к директории также передаётся в виде аргумента командной строки.**

Сначала я создаю файл scr04 и устанавливаю разрешения на выполнение



The screenshot shows a video lecture interface. On the left is a small video feed of a man with glasses. The main area displays a terminal window titled 'ganzl@fedora:~/scr' with the following commands and output:

```
[ganzl@fedora scr]$ touch scr04
[ganzl@fedora scr]$ vi scr04
[ganzl@fedora scr]$ chmod 777 scr04
[ganzl@fedora scr]$
```

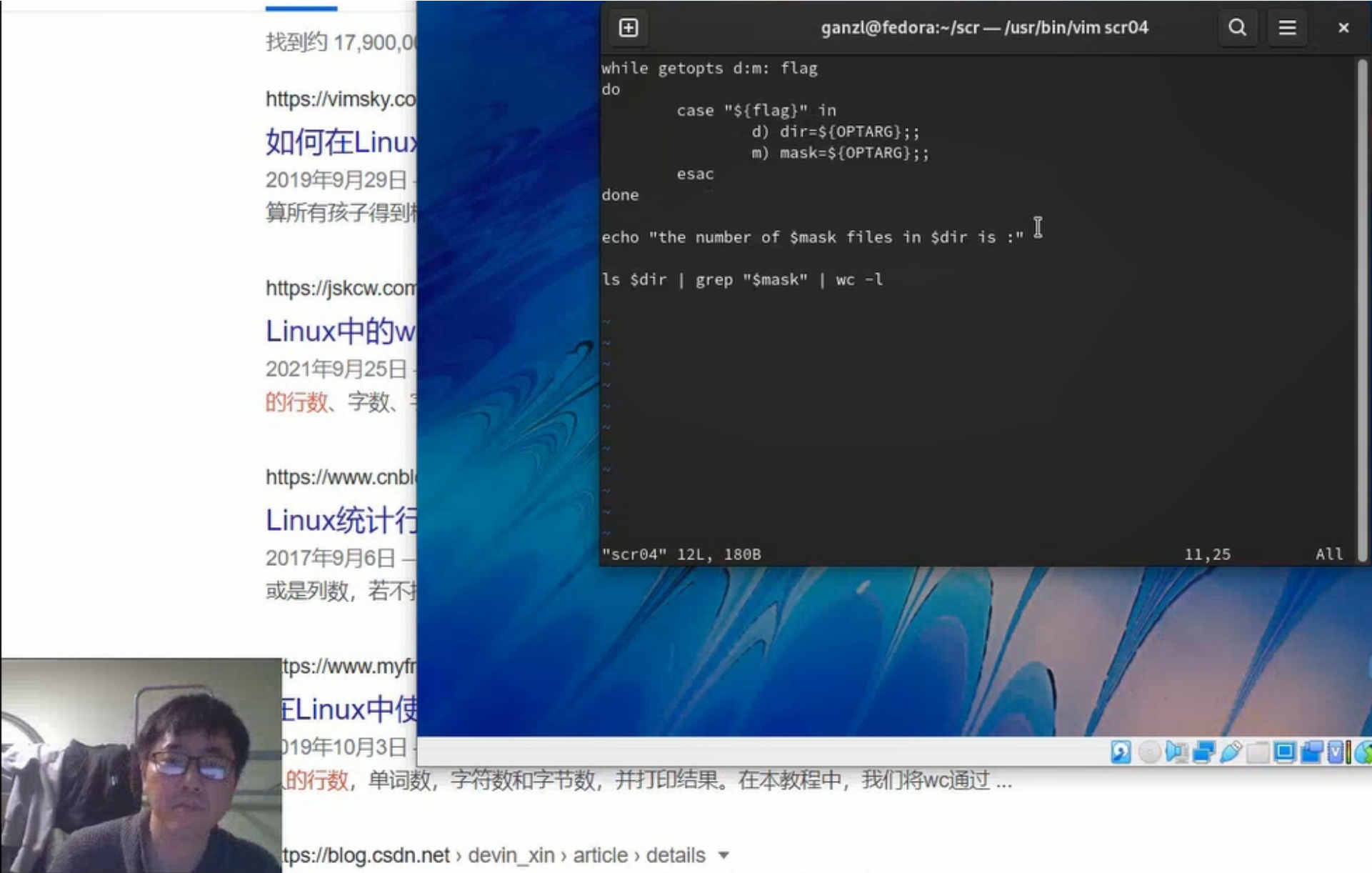
Below the terminal, a document titled 'Лабораторная работа № 10. Программирование в командном процессоре' is visible, showing the text:

96

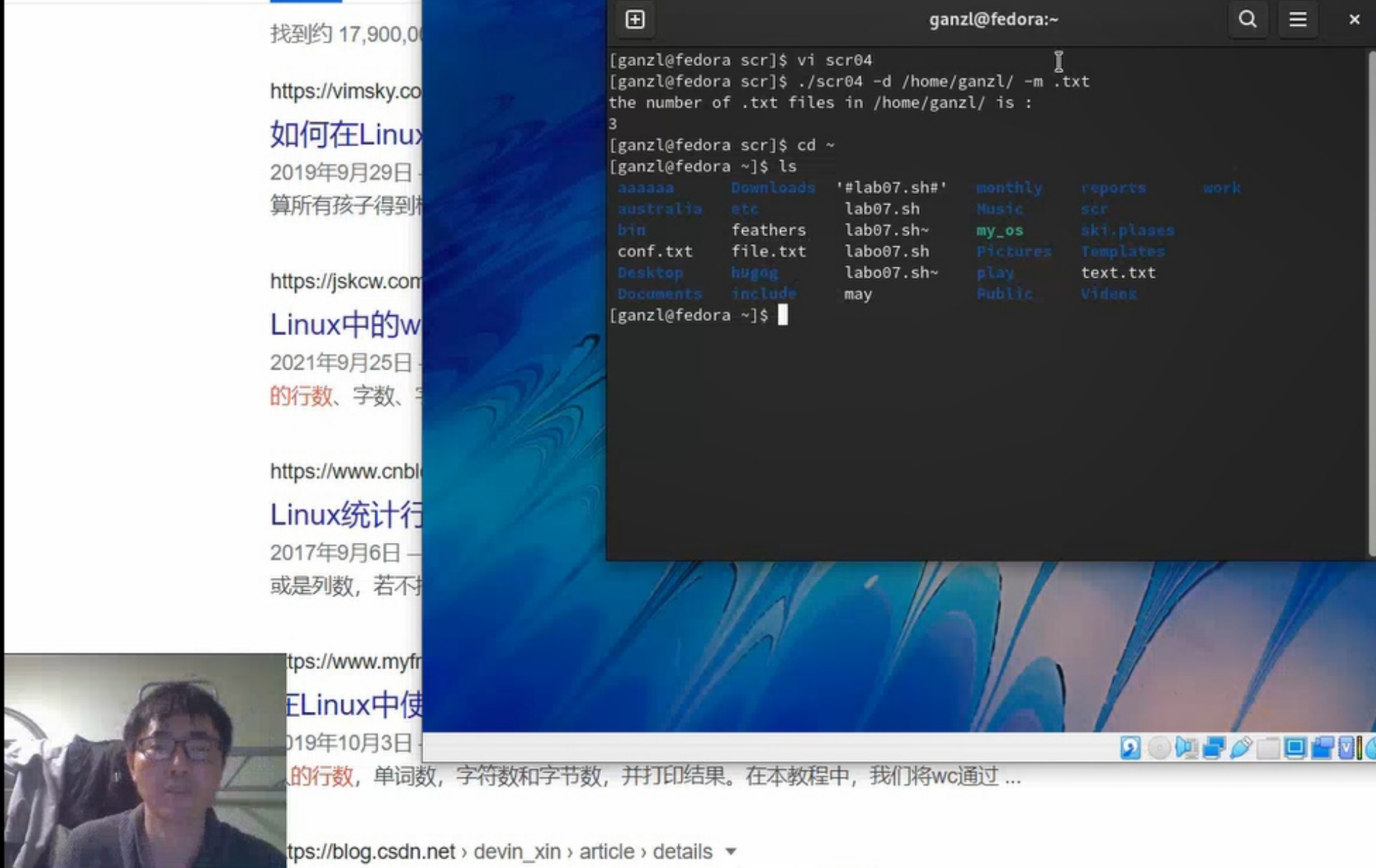
5. Ответы на контрольные вопросы.

10.5. Контрольные вопросы

Затем я использую команду vi для редактирования файла, как показано ниже



Наконец я выполняю файл и проверяю, полностью ли он достигнут



## Контрольные вопросы:

1. Объясните понятие командной оболочки. Приведите примеры командных оболочек. Чем они отличаются?
2. Командный процессор (командная оболочка, интерпретатор команд shell) — это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера.



3. оболочка Борна (Bourne shell или sh) — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций;
4. С-оболочка (или csh) — надстройка на оболочкой Борна, использующая С-подобный синтаксис команд с возможностью сохранения истории выполнения команд;
5. оболочка Корна (или ksh) — напоминает оболочку С, но операторы управления программой совместимы с операторами оболочки Борна;
6. BASH — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек С и Корна (разработка компании Free Software Foundation).
7. Что такое POSIX?
8. POSIX (Portable Operating System Interface for Computer Environments) — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ.
9. Как определяются переменные и массивы в языке программирования bash?
10. set -А
11. Каково назначение операторов let и read?
12. Команда let берет два операнда и присваивает их переменной.
13. Команда read позволяет читать значения переменных со стандартного ввода.
14. Какие арифметические операции можно применять в языке программирования bash?
15. Простейшими математическими выражениями являются сложение (+), вычитание (-), умножение (\*), целочисленное деление (/) и целочисленный остаток от деления (%).
16. Что означает операция (( ))?
17. условия оболочки bash могут быть записаны в двойных скобках
18. Какие стандартные имена переменных Вам известны?
19. В программировании соглашение об именах — набор правил именования идентификаторов, обозначающих переменные, типы, функции и другие вещи в исходном коде и документации.
20. Что такое метасимволы?
21. Метасимвол (Meta Character) относится к символам, которые можно вводить с клавиатуры и которые имеют другие специальные значения для оболочки, которые называются метасимволами.
22. Как экранировать метасимволы?
23. \
24. Как создавать и запускать командные файлы?
25. Запустите текстовый редактор.Последовательно запишите команды, располагая каждую команду на отдельной строке. Сохраните этот файл, сделайте его исполняемым, применив команду: chmod +x имя\_файла.
26. Запустите созданный файл и проверьте правильность выполнения команд. В случае нахождения ошибки, в текстовом редакторе внесите изменения в командный файл, сохраните его и проверьте еще раз.
27. Как определяются функции в языке программирования bash?
28. `function functionname {command...}` или `functionname () {command...}`
29. Каким образом можно выяснить, является файл каталогом или обычным файлом?
30. -d FILE – Истина, если ФАЙЛ существует и является каталогом. -e FILE – Истинно, если ФАЙЛ существует и является файлом, независимо от его типа (узел, каталог, сокет и т. Д.). -f FILE – Истина, если ФАЙЛ существует и является обычным файлом (не каталогом или устройством).
31. Каково назначение команд set, typeset и unset?
32. Функция обработки
33. Как передаются параметры в командные файлы?

## - params.bat

---

## Вывод:

- Я знаком с операционной системой Linux. Приобрел практические навыки использования редактора Emacs.