Отчёт по лабораторной работе №10

Дисциплина: Операционные системы

Рыжкова Ульяна Валерьевна

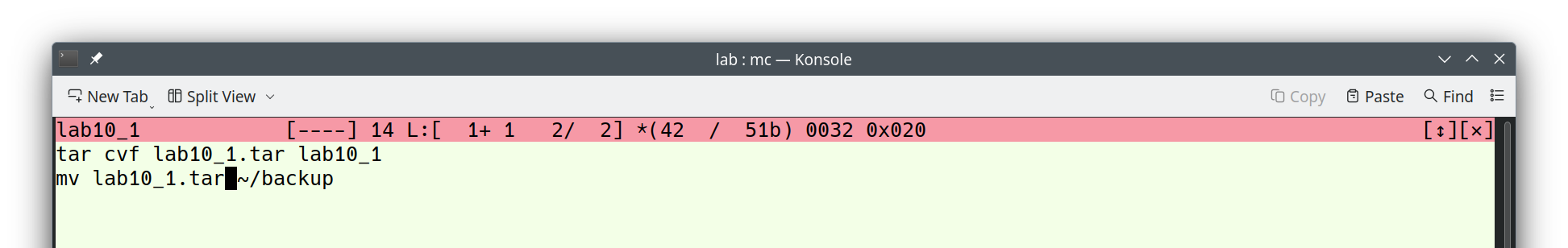
Содержание

# 1 Цель работы

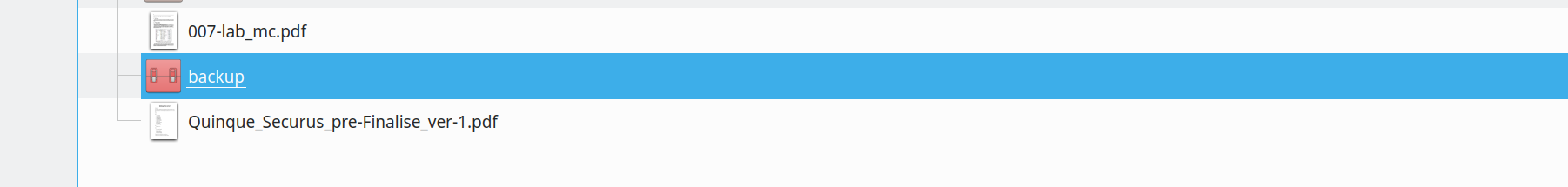
Изучить основы программирования в оболочке OC UNIX. НАучиться писать небольшие командные файлы.

# 2 Выполнение лабораторной работы

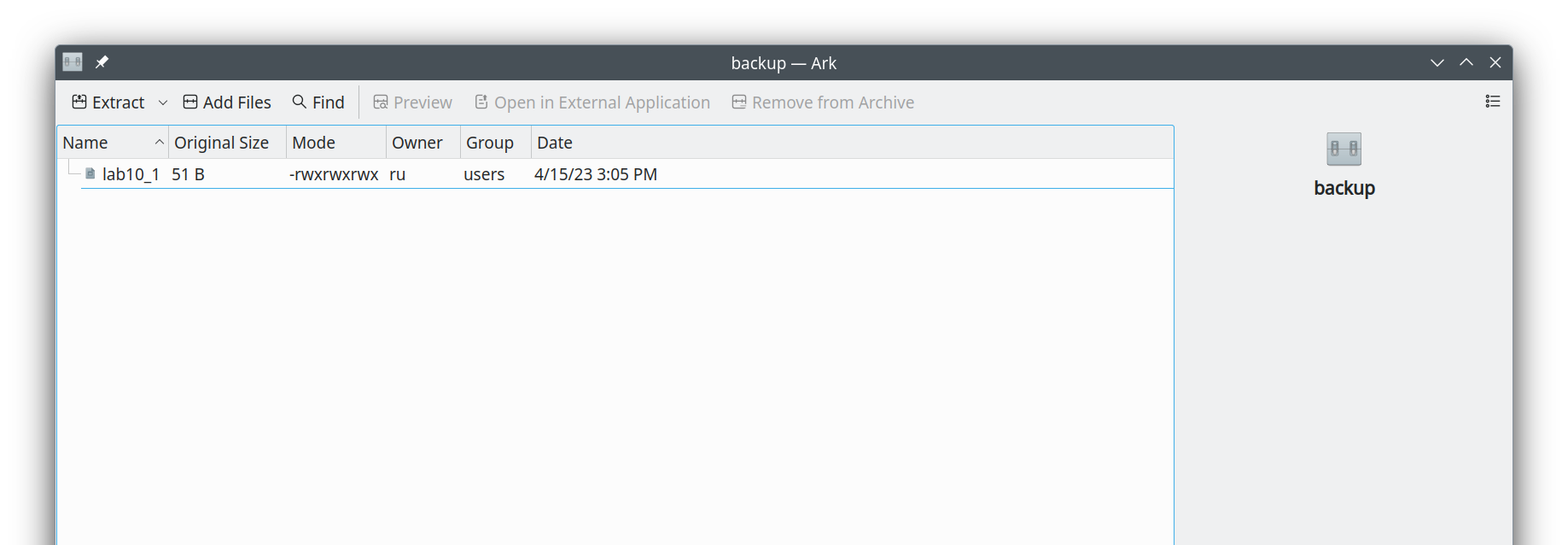
1. Пишем скрипт, который архивирует сам себя и перемещает в директорию backup в домашнем каталоге



код

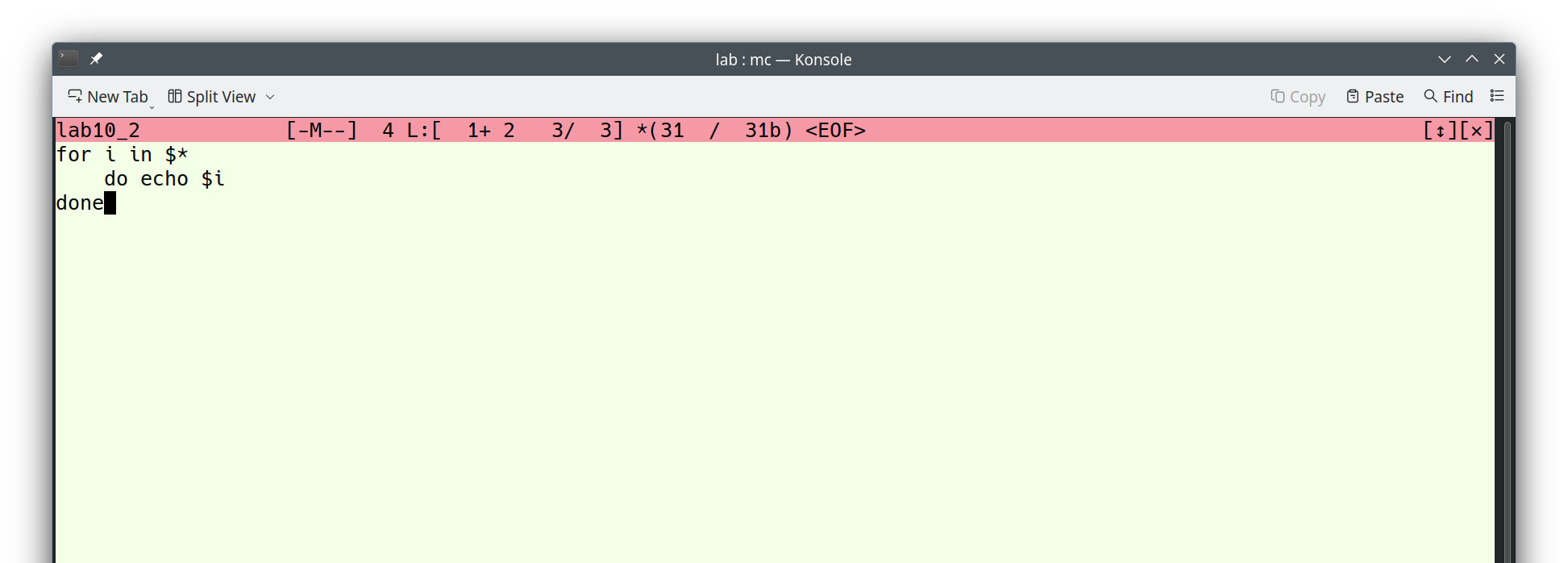


backup в домашнем каталоге

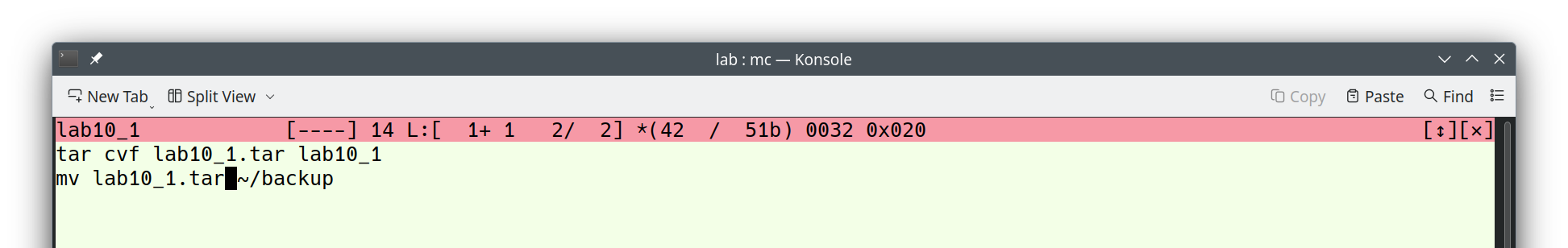


содержание backup

1. Пишем скрипт, обрабатывающий произвольное число аргументов (в том числе превышающее 10)

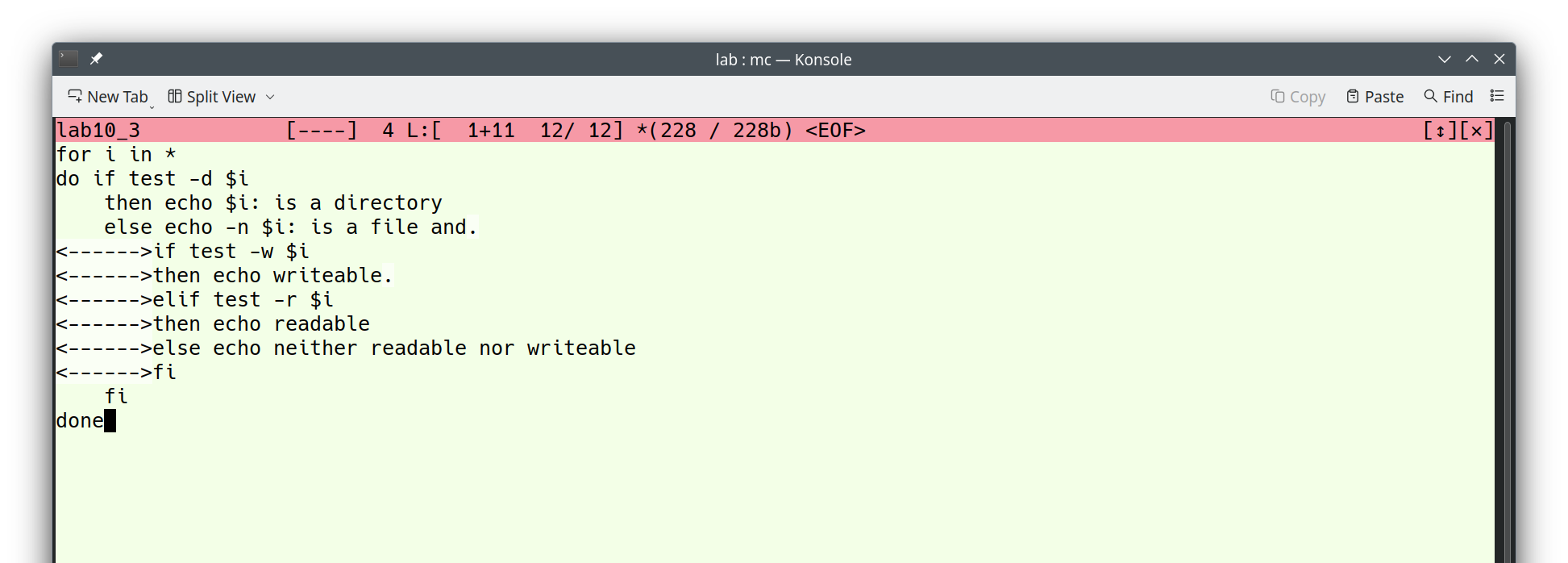


код

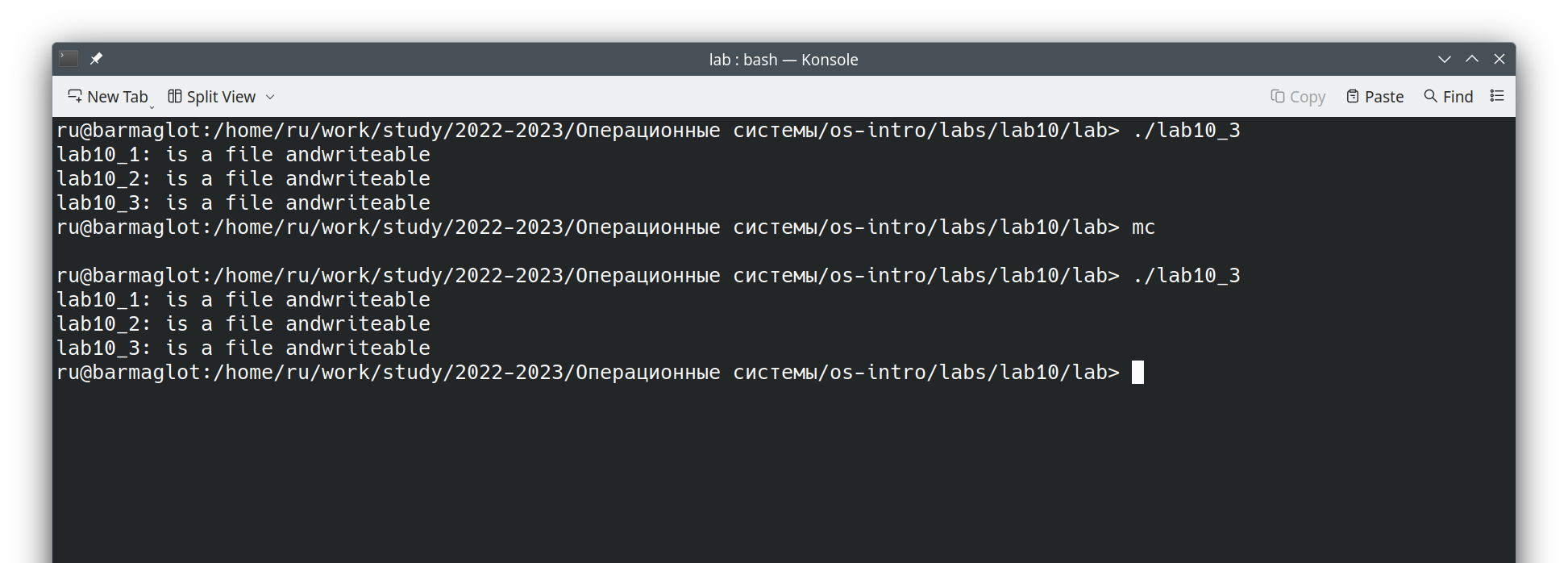


результат вывода 11 аргументов

1. Пишем скрипт, являющийся аналогом команды ls

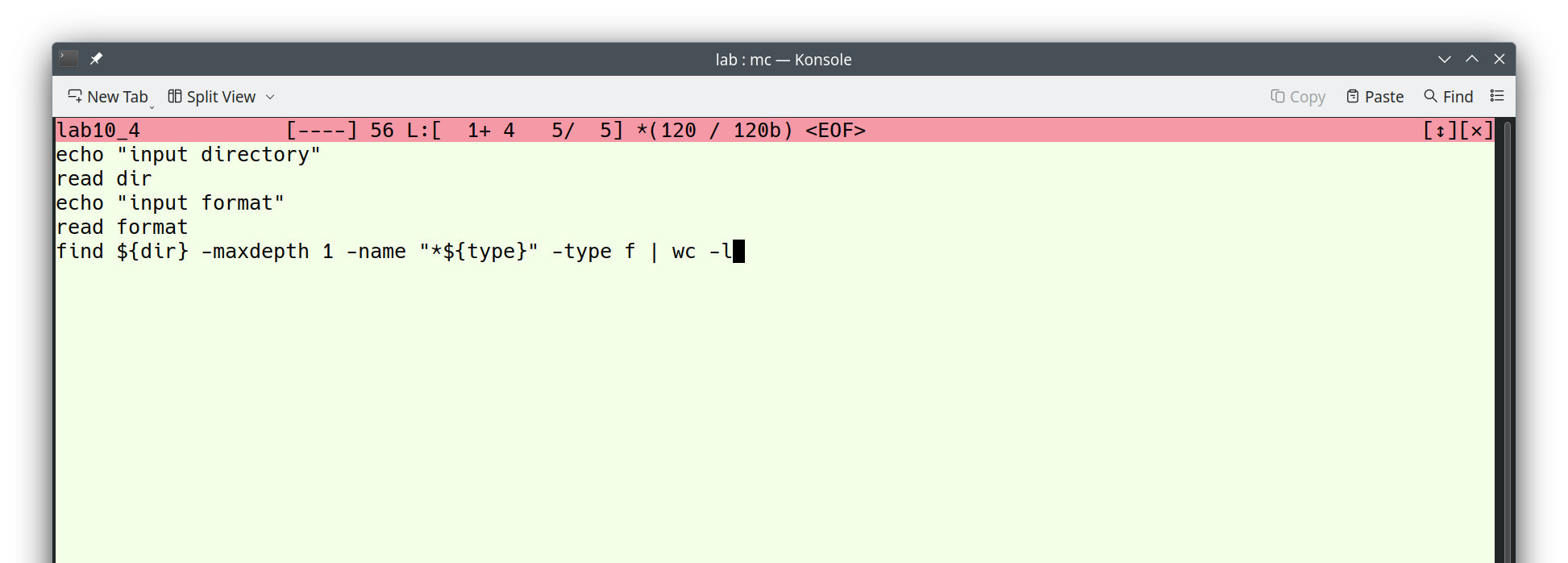


код

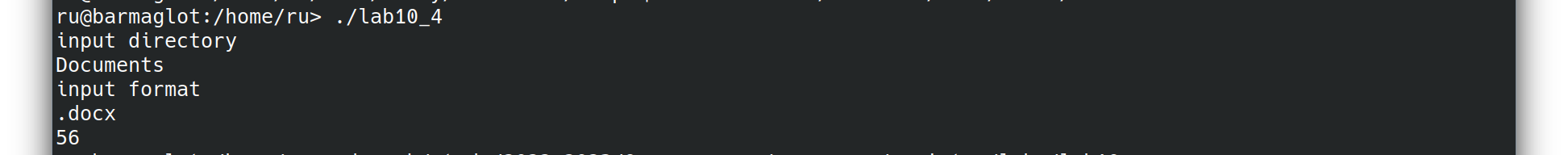


вывод информации о файлах

1. Пишем скрипт, который вычисляет количество файлов заданного формата в указанной директории



код



вывод информации о каталоге

# 3 Ответы на контрольные вопросы

1. Объясните понятие командной оболочки. Приведите примеры командных оболочек. Чем они отличаются?

*Командная оболочка* - это программа, позволяющая пользователю взаимодействовать с операционной системой компьютера. В операционных системах типа UNIX/Linux наиболее часто используются следующие реализации командных оболочек: \* *оболочка Борна (Bourne shell или sh)* — стандартная командная оболочка UNIX/Linux, содержащая базовый, но при этом полный набор функций; \* *С-оболочка (или csh)* — надстройка на оболочкой Борна, использующая С-подобный синтаксис команд с возможностью сохранения истории выполнения команд; \* *оболочка Корна (или ksh)* — напоминает оболочку С, но операторы управления программой совместимы с операторами оболочки Борна; \* *BASH* — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек С и Корна (разработка компании Free Software Foundation).

1. Что такое POSIX?

*POSIX (Portable Operating System Interface for Computer Environments)* — набор стандартов описания интерфейсов взаимодействия операционной системы и прикладных программ.

1. Как определяются переменные и массивы в языке программирования bash?

Команда mark=/usr/andy/bin присваивает значение строки символов /usr/andy/bin переменной mark типа строка символов, массивы задают командой set с флагом -А.

1. Каково назначение операторов let и read?

Команда let берет два операнда и присваивает их переменной, а команда read позволяет читать значения переменных со стандартного ввода.

1. Какие арифметические операции можно применять в языке программирования bash?

C помощью команды let можно производить базовые арифметические операции: сложение/вычитание, умножение/деление, логические операции.

1. Что означает операция (( ))?

В двойные скобки можно записывать выражения для вычисления.

1. Какие стандартные имена переменных Вам известны?

PATH, HOME, IFS, MAIL, TERM, LOGNAME. Остальные переменные можно посмотреть с помощью команды set

1. Что такое метасимволы?

Такие символы, как ’ < > \* ? |  ” &, являются метасимволами и имеют для командного процессора специальный смысл.

1. Как экранировать метасимволы?

Экранирование может быть осуществлено с помощью предшествующего метасимволу символа , который, в свою очередь, является метасимволом.

1. Как создавать и запускать командные файлы?

Создаём файлы командой touch (как и любые другие), запускаем как исполняемые (с комбинацией ./имя\_файла), предварительно сделав их исполняемыми

1. Как определяются функции в языке программирования bash?

Существует ключевое слово function, после которого следует имя функции и список команд, заключённых в фигурные скобки. Удалить функцию можно с помощью команды unset c флагом -f.

1. Каким образом можно выяснить, является файл каталогом или обычным файлом?

test -d FILE - истина, если FILE является каталогом. test -e FILE - истина, если FILE является файлом.

1. Каково назначение команд set, typeset и unset?

* команда set - вывод списка переменных окружения
* команда unset - удаление переменной из окружения командной оболочки

Команда typeset имеет четыре опции для работы с функциями: \* -f — перечисляет определённые на текущий момент функции; \* -ft — при последующем вызове функции инициирует её трассировку; \* -fx — экспортирует все перечисленные функции в любые дочерние программы оболочек; \* -fu — обозначает указанные функции как автоматически загружаемые. Автоматически загружаемые функции хранятся в командных файлах, а при их вызове оболочка просматривает переменную FPATH, отыскивая файл с одноимёнными именами функций, загружает его и вызывает эти функции.

1. Как передаются параметры в командные файлы?

При вызове командного файла на выполнение параметры ему могут быть переданы точно таким же образом, как и выполняемой программе. С точки зрения командного файла эти параметры являются позиционными. Символ $ является метасимволом командного процессора. Он используется, в частности, для ссылки на параметры, точнее, для получения их значений в командном файле. В командный файл можно передать до девяти параметров. При использовании где-либо в командном файле комбинации символов $i, где 0 < 𝑖 < 10, вместо неё будет осуществлена подстановка значения параметра с порядковым номером i, т.е. аргумента командного файла с порядковым номером i. Использование комбинации символов $0 приводит к подстановке вместо неё имени данного командного файла.

1. Назовите специальные переменные языка bash и их назначение

OPTARG и OPTIND. Если ожидается дополнительное значение, то OPTARG устанавливается в значение этого аргумента (будет равна file\_in.txt для опции i и file\_out.doc для опции o. OPTIND является числовым индексом на упомянутый аргумент.

# 4 Выводы

Я освоила базовые команды для программирования в оболочке ОС UNIX