

Отчёт по лабораторной работе №10

Дисциплина: Операционные системы

Рыжкова Ульяна Валерьевна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Ответы на контрольные вопросы	10
4	Выводы	14

Список иллюстраций

2.1	код	6
2.2	backur в домашнем каталоге	6
2.3	содержание backur	7
2.4	код	7
2.5	результат вывода 11 аргументов	7
2.6	код	8
2.7	вывод информации о файлах	8
2.8	код	9
2.9	вывод информации о каталоге	9

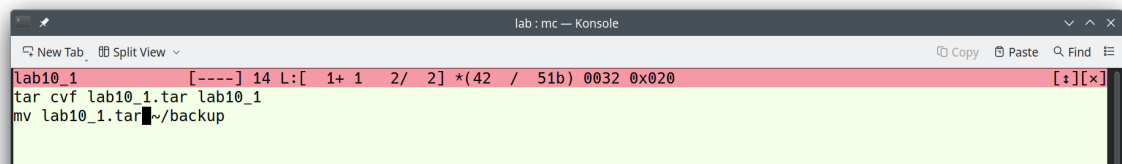
Список таблиц

1 Цель работы

Изучить основы программирования в оболочке ОС 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



```
lab10_1 [----] 14 L: [ 1+ 1 2/ 2] *(42 / 51b) 0032 0x020 [!][x]
tar cvf lab10_1.tar lab10_1
mv lab10_1.tar ~/backup
```

Рис. 2.1: код



Рис. 2.2: backup в домашнем каталоге

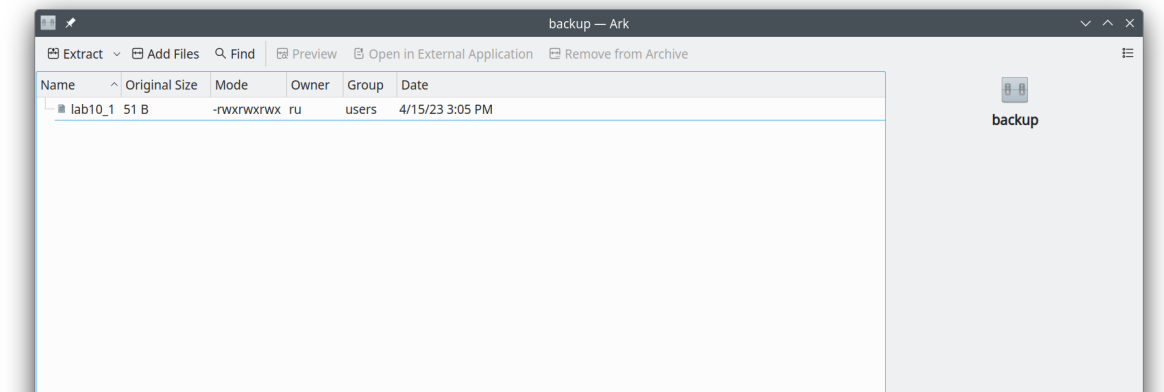


Рис. 2.3: содержание backup

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

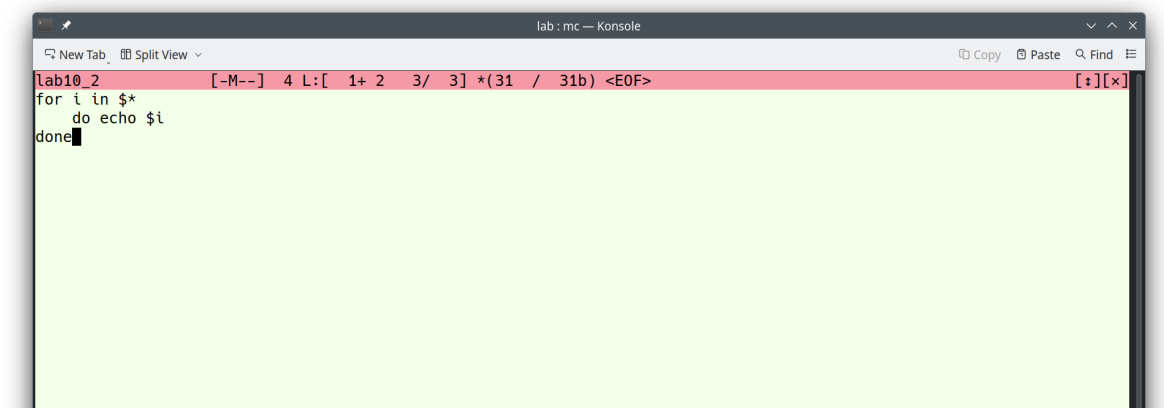


Рис. 2.4: код

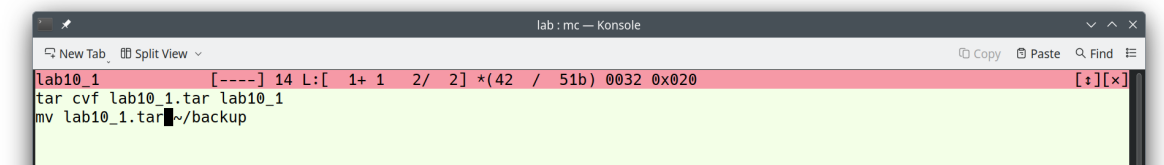
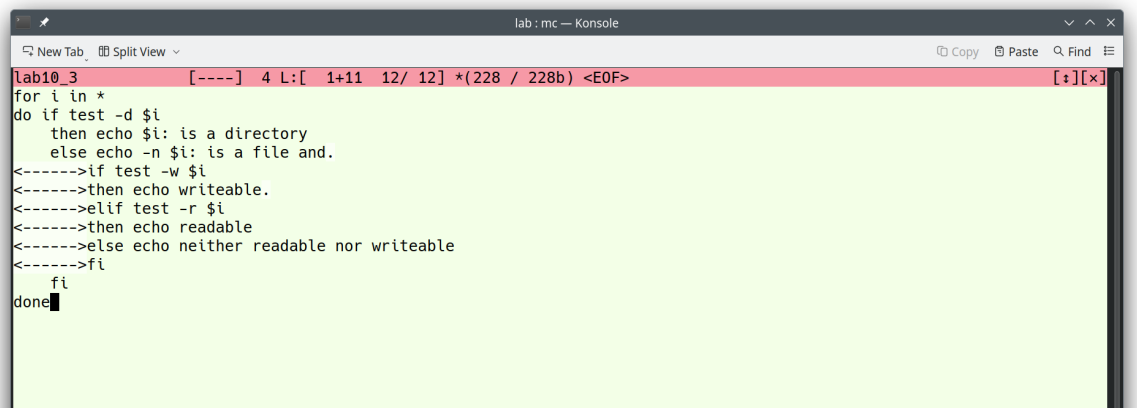


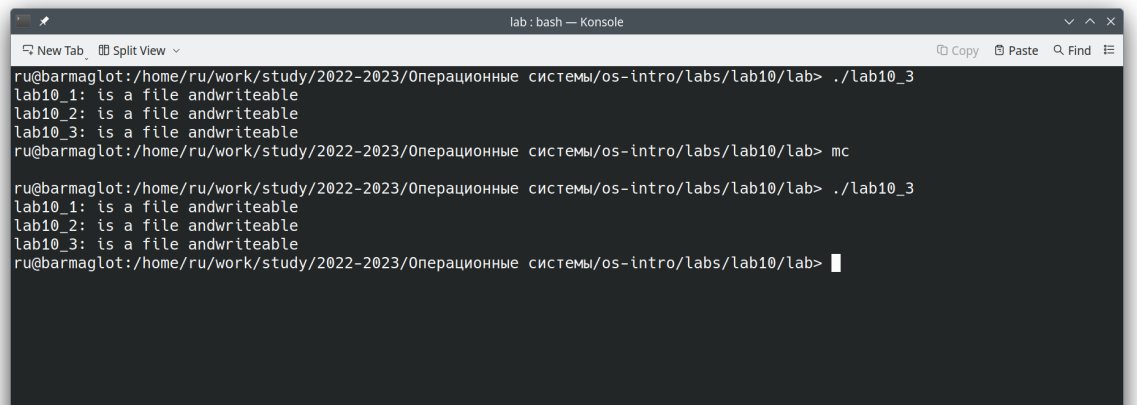
Рис. 2.5: результат вывода 11 аргументов

3. Пишем скрипт, являющийся аналогом команды ls



```
lab10_3 [----] 4 L: [ 1+11 12/ 12] *(228 / 228b) <EOF> [!][x]
for i in *
do if test -d $i
then echo $i: is a directory
else echo -n $i: is a file and.
<----->if test -w $i
<----->then echo writeable.
<----->elif test -r $i
<----->then echo readable
<----->else echo neither readable nor writeable
<----->fi
fi
done
```

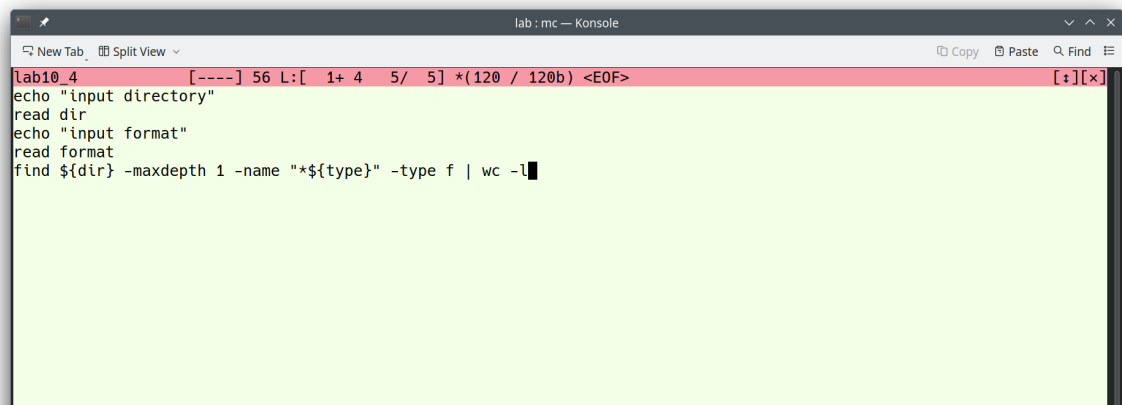
Рис. 2.6: код



```
ru@barmaglot:/home/ru/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/lab> ./lab10_3
lab10_1: is a file andwriteable
lab10_2: is a file andwriteable
lab10_3: is a file andwriteable
ru@barmaglot:/home/ru/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/lab> mc
ru@barmaglot:/home/ru/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/lab> ./lab10_3
lab10_1: is a file andwriteable
lab10_2: is a file andwriteable
lab10_3: is a file andwriteable
ru@barmaglot:/home/ru/work/study/2022-2023/Операционные системы/os-intro/labs/lab10/lab> █
```

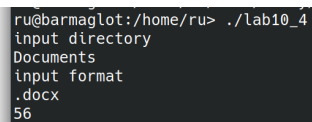
Рис. 2.7: вывод информации о файлах

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



```
lab : mc — Konsole
New Tab Split View
Copy Paste Find
lab10_4 [----] 56 L: [ 1+ 4 5/ 5] *(120 / 120b) <EOF> [!][x]
echo "input directory"
read dir
echo "input format"
read format
find ${dir} -maxdepth 1 -name "*${type}" -type f | wc -l
```

Рис. 2.8: код



```
ru@barmaglot:/home/ru> ./lab10_4
input directory
Documents
input format
.docx
56
```

Рис. 2.9: 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; * C-оболочка (или *csh*) — надстройка на оболочке Борна, использующая C-подобный синтаксис команд с возможностью сохранения истории выполнения команд; * оболочка Корна (или *ksh*) — напоминает оболочку C, но операторы управления программой совместимы с операторами оболочки Борна; * *BASH* — сокращение от Bourne Again Shell (опять оболочка Борна), в основе своей совмещает свойства оболочек C и Корна (разработка компании Free Software Foundation).

2. Что такое POSIX?

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

3. 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 `-A`.

4. 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.

5. Какие арифметические операции можно применять в языке программирования `bash`?

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

6. Что означает операция `(())`?

В двойные скобки можно записывать выражения для вычисления.

7. Какие стандартные имена переменных Вам известны?

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

8. Что такое метасимволы?

Такие символы, как `' < > * ? | " &`, являются метасимволами и имеют для командного процессора специальный смысл.

9. Как экранировать метасимволы?

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

10. 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

11. 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` с флагом `-f`.

12. 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.

13. 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 иницирует 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.

14. 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.

15. Назовите специальные переменные языка `bash` и их назначение

`OPTARG` и `OPTIND`. Если ожидается дополнительное значение, то `OPTARG` устанавливается в значение этого аргумента (будет равно `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