

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 12__

дисциплина: *Операционные системы*

Студент: Ниemek Яи Жак

Группа: НММБд-04-24

МОСКВА

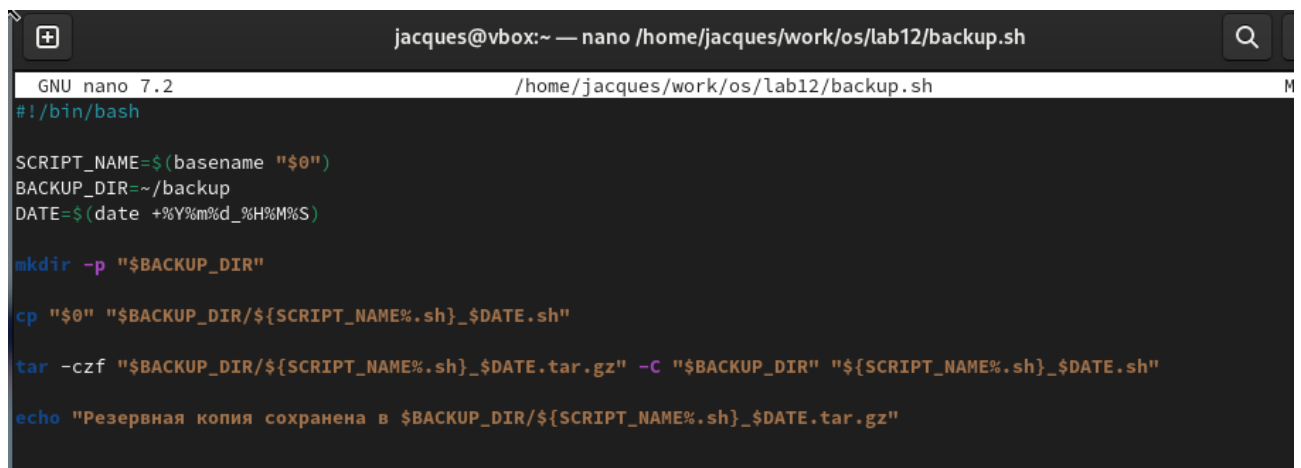
2025__ г.

Цель работы

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

Последовательность выполнения 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.
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.
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.
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



```
jacques@vbox:~ — nano /home/jacques/work/os/lab12/backup.sh
GNU nano 7.2 /home/jacques/work/os/lab12/backup.sh
#!/bin/bash

SCRIPT_NAME=$(basename "$0")
BACKUP_DIR=~/.backup
DATE=$(date +%Y%m%d_%H%M%S)

mkdir -p "$BACKUP_DIR"

cp "$0" "$BACKUP_DIR/${SCRIPT_NAME%.sh}_${DATE}.sh"

tar -czf "$BACKUP_DIR/${SCRIPT_NAME%.sh}_${DATE}.tar.gz" -C "$BACKUP_DIR" "${SCRIPT_NAME%.sh}_${DATE}.sh"

echo "Резервная копия сохранена в $BACKUP_DIR/${SCRIPT_NAME%.sh}_${DATE}.tar.gz"
```

```
jacques@vbox:~/work/os$ cd lab12
jacques@vbox:~/work/os/lab12$ touch backup.sh
jacques@vbox:~/work/os/lab12$ nano backup.sh
jacques@vbox:~/work/os/lab12$ chmod +x backup.sh
```

```
jacques@vbox:~/work/os/lab12$ ls
backup.sh
jacques@vbox:~/work/os/lab12$ ./backup.sh
tar: backup_DATE.sh : stat impossible: Aucun fichier ou dossier de ce type
tar: Arrêt avec code d'échec à cause des erreurs précédentes
резервная копия сохранена в /home/jacques/backup/backup_20250829_103620.tar.gz
jacques@vbox:~/work/os/lab12$
```

1. Командная оболочка (shell)

- * Программа, обеспечивающая взаимодействие пользователя с ОС через команды.
- * Примеры: bash, sh, zsh, csh.
- * Отличия: синтаксис команд, поддержка скриптов, возможности автодополнения и истории команд.

2. POSIX

- * POSIX (Portable Operating System Interface) — стандарт для совместимости операционных систем UNIX-подобных.
- * Определяет набор команд, системных вызовов и стандартов для скриптов.

3. Переменные и массивы в Bash

- * Переменная:

```
VAR=value
echo $VAR
```

- * Массив:

```
ARR=(one two three)
echo ${ARR[1]} # "two"
```

4. Операторы let и read

* let – арифметические вычисления:

```
let "a = 5 + 3"
```

* read – считывание ввода пользователя:

```
read name  
echo "Hello, $name"
```

5. Арифметические операции в Bash

* Сложение +, вычитание -, умножение *, деление /, остаток %, возведение в степень **.

6. Операция (())

* Арифметическая оценка выражений.

```
((a = 5 + 3))
```

7. Стандартные имена переменных

* HOME, PATH, USER, PWD, SHELL, UID, RANDOM, BASH_VERSION

8. Метасимволы

* Символы с особым значением для оболочки:

```
* ? $ & ; | > < \ [ ] { } ! ~
```

9. Экранирование метасимволов

* С помощью обратного слеша `\`` или кавычек:`

```
echo "Hello $USER"  
echo \$USER
```

10. Создание и запуск командных файлов

- * Создать файл с расширением .sh и написать команды.
- * Сделать исполняемым:

```
chmod +x file.sh
./file.sh
```

11. Определение функций в Bash

```
myfunc() {
    echo "Hello"
}
myfunc
```

12. Проверка типа файла

- * Каталог: -d
 - * Файл: -f
- ```
if [-d "$FILE"]; then echo "Directory"; fi
```

---

### ### 13. Команды set, typeset, unset

- \* set – установка опций оболочки и просмотр переменных.
- \* typeset – объявление переменных с типом.
- \* unset – удаление переменной.

---

### ### 14. Передача параметров в скрипты

- \* \$1, \$2, ..., \$n – позиционные параметры
- \* @\$ или \$\* – все параметры
- \* \$# – количество параметров

---

### ### 15. Специальные переменные Bash

- \* \$0 – имя скрипта
- \* \$? – код завершения последней команды
- \* \$\$ – PID текущего процесса
- \* \$! – PID последнего фонового процесса

\* \$# – количество аргументов

\* \$\* / @\$ – все аргументы