Отчёт по лабораторной работе №14

Операционные системы

Пономарева Татьяна Александровна

Содержание

# 1 Цель работы

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

# 2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

# 3 Теоретическое введение

Здесь описываются теоретические аспекты, связанные с выполнением работы.

Например, в табл. 1 приведено краткое описание стандартных каталогов Unix.

Таблица 1: Описание некоторых каталогов файловой системы GNU Linux

| Имя каталога | Описание каталога |
| --- | --- |
| / | Корневая директория, содержащая всю файловую |
| /bin | Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям |
| /etc | Общесистемные конфигурационные файлы и файлы конфигурации установленных программ |
| /home | Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя |
| /media | Точки монтирования для сменных носителей |
| /root | Домашняя директория пользователя root |
| /tmp | Временные файлы |
| /usr | Вторичная иерархия для данных пользователя |

Более подробно про Unix см. в [1–4].

# 4 Выполнение лабораторной работы

Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию (рис. 1).



Рис. 1: Название рисунка

# 5 Ответы на контрольные вопросы

1. Найдите синтаксическую ошибку в следующей строке: while [$1 != “exit”]

Ошибка: Пропущены пробелы вокруг оператора сравнения. Правильный синтаксис:

while [ “$1” != “exit” ]

1. Как объединить (конкатенация) несколько строк в одну? В Bash можно объединить строки несколькими способами:

Использовать символ + (внутри кавычек):

str1=“Hello” str2=“world” result=“$str1 $str2"
echo "$result” # Вывод: Hello world Использовать команду echo с конкатенацией:

echo “Hello” “world” # Вывод: Hello world

1. Найдите информацию об утилите seq. Какими иными способами можно реализовать её функционал при программировании на bash? Утилита seq используется для генерации последовательностей чисел, например:

seq 1 5 # Выводит: 1 2 3 4 5 Иными способами можно сгенерировать такую последовательность с помощью цикла for:

for i in {1..5}; do echo $i; done # Выводит: 1 2 3 4 5

1. Какой результат даст вычисление выражения $( (10/3))? Результат будет равен 3, так как это целочисленное деление в Bash. При выполнении деления целых чисел в Bash результат округляется в меньшую сторону.
2. Укажите кратко основные отличия командной оболочки zsh от bash.

zsh обладает более мощными возможностями автодополнения команд и путей, чем bash.

В zsh есть поддержка тем и плагинов, что делает её более настраиваемой.

zsh предлагает расширенные возможности работы с массивами и строками.

В zsh присутствует более гибкое управление историей команд.

В отличие от bash, zsh поддерживает “glob” с возможностью использования регулярных выражений для сопоставления файлов.

1. Проверьте, верен ли синтаксис данной конструкции:

for ((a=1; a <= LIMIT; a++)) Ответ: Синтаксис верен при условии, что LIMIT является заранее определённой переменной с числовым значением. Важно, чтобы переменная LIMIT была инициализирована до начала цикла.

1. Сравните язык bash с какими-либо языками программирования. Какие преимущества у bash по сравнению с ними? Какие недостатки?

Преимущества bash:

Простота в написании сценариев для автоматизации задач в ОС UNIX/Linux.

Поддержка выполнения команд непосредственно в оболочке и взаимодействие с операционной системой.

Лёгкость в использовании для написания одноразовых скриптов и утилит для системного администрирования.

Недостатки bash:

Отсутствие строгой типизации данных.

Ограниченные возможности для сложных вычислений и работы с большими объёмами данных по сравнению с полноценными языками программирования.

Сложности при отладке, так как ошибки часто трудно выявить в больших скриптах.

# 6 Выводы

Были изучены основы программирования в оболочке ОС UNIX. Были получены знания о написании более сложных командных файлов с использованием логических управляющих конструкций и циклов.

# Список литературы

1. [Курс на ТУИС](https://esystem.rudn.ru/course/view.php?id=113)

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.

2. Robbins A. Bash Pocket Reference. O’Reilly Media, 2016. 156 с.

3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.

4. Newham C. [Learning the bash Shell: Unix Shell Programming](http://www.amazon.com/Learning-bash-Shell-Programming-Nutshell/dp/0596009658). O’Reilly Media, 2005. 354 с.