

Московский государственный технический университет им. Н.Э. Баумана

Кафедра «Автоматизированные системы обработки информации и управления»

Сёмкин П.С., Сёмкин А.П.

Методические материалы к лабораторным работам

по дисциплине

«Операционные системы»

Лабораторная работа № 1

**«ОС Alt Linux. Установка операционной системы Альт Рабочая станция.
Интерфейс пользователя»**

Москва

2024 г.

ОГЛАВЛЕНИЕ

1	ЦЕЛЬ РАБОТЫ	3
2	ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	3
2.1	Физические и виртуальные ресурсы операционных систем	3
2.2	Программа виртуализации Oracle VM VirtualBox	5
2.3	Дистрибутив ОС Альт Рабочая станция	6
2.4	Установка операционной системы Альт Рабочая станция	6
2.5	Загрузка операционной системы. Уровни выполнения.....	6
2.6	Графический режим выполнения Alt Linux	9
2.7	Консольный режим выполнения Alt Linux.....	9
2.7.1	Виртуальные консоли	9
2.7.2	Командные процессоры (интерпретаторы команд)	10
2.7.3	Внутренние и внешние команды интерпретатора команд	11
2.7.4	История выполнения команд интерпретатора	11
3	ВЫПОЛНЕНИЕ РАБОТЫ	12
3.1	Задание.....	12
3.2	Порядок выполнения работы.....	12
3.2.1	Создание виртуальной машины alt-10.....	13
3.2.2	Настройка виртуальной машины Alt-10.....	13
3.2.3	Установка операционной системы Альт Рабочая станция	13
3.2.4	Работа с ОС Альт Рабочая станция в графическом режиме	13
3.2.5	Работа с ОС Альт Рабочая станция в консольном режиме	14
4	КОНТРОЛЬНЫЕ ВОПРОСЫ	14
5	ЛИТЕРАТУРА	15
6	ПРИЛОЖЕНИЕ	15
6.1	Создание виртуальных машин и установка гостевых ОС в программе виртуализации VirtualBox	15
6.1.1	Создание виртуальной машины и установка гостевой ОС с использованием дистрибутива	15
6.1.2	Создание виртуальной машины и установка гостевой ОС путем импорта готовой конфигурации	16
6.2	Установка операционной системы Alt Linux.....	16
6.3	Команды переключение между виртуальными консолями	20
6.4	Командный интерпретатор BASH	20
6.4.1	Запуск интерпретатора команд	20
6.4.2	Получение информации о внутренних и внешних командах	21
6.4.3	Команды получения общей информации о системе	21
6.4.4	Переменные среды окружения интерпретатора команд.....	22
6.5	Изменение уровней выполнения ОС в системе инициализации systemd.....	24
6.5.1	Уровни выполнения операционной системы.....	24
6.5.2	Изменение уровня выполнения системы во время сеанса.....	27
6.5.3	Установка уровня выполнения системы по умолчанию при загрузке системы	27

1 Цель работы

Целью работы является:

- знакомство с концепцией виртуализации;
- изучение возможностей и интерфейса программы виртуализации **VirtualBox**;
- создание виртуальной машины и установка гостевой операционной системы **Альт Рабочая станция**;
- знакомство и работа в графической оболочке **MATE ОС Alt Linux**;
- знакомство и работа с командным интерпретатором **bash ОС Alt Linux**.

2 Теоретическая часть

2.1 Физические и виртуальные ресурсы операционных систем

Физический ресурс - реально существует и при распределении его между процессами используются все его физические характеристики.

Примеры физических ресурсов: оперативная память (характеристики: объём ОП, время доступа, центральный процессор (быстродействие, число ядер и т.д.).

Виртуальный ресурс – некоторая программная модель, построенная на основе одного или нескольких физических или виртуальных ресурсов, и обладающая характеристиками, отличными от характеристик ресурсов, на основе которых она построена. Пример виртуального ресурса: виртуальная память, которая представляется имеющей гораздо больший объём, чем физическая память, установленная на компьютере. Программная модель виртуальной памяти строится на основе двух физических ресурсов: оперативной и внешней памяти.

Виртуализация предоставляет дополнительные возможности по использованию ресурсов, предоставляя расширенный интерфейс и скрывая сложные архитектуры аппаратных средств

Виртуальные машины

Расширением идей виртуализации является предоставление пользователю виртуальной вычислительной машины.

Виртуальная машина — это программная модель реальной машины, созданная с использованием программ виртуализации и средств операционной системы.

С точки зрения пользователя, виртуальные машины выглядят как реально существующие машины. На одной реальной машине может быть установлено несколько виртуальных, на каждой из которых может быть установлено собственное программное обеспечение. Согласно такой концепции было создано множество программных систем виртуализации.

Каждая виртуальная машина содержит собственную операционную систему и множество виртуальных аппаратных ресурсов.

Мультипрограммные операционные системы разделяют ресурсы системы между несколькими процессами. Каждому из этих процессов выделяется порция ресурсов реальной машины. Каждый процесс видит менее мощную машину с меньшими возможностями, чем машина, на которой действительно выполняется процесс.

Системы виртуальных машин разделяют ресурсы единственной машины по-другому. Они создают иллюзию того, что одна реальная машина - это на самом деле несколько машин. Они создают виртуальные процессоры, устройства хранения и ввода/вывода, возможно, с куда большими объёмами и возможностями, чем таковые у реальной машины.

ОС реальной машины создает среду, в которой могут работать виртуальные машины. Она обеспечивает поддержку различных операционных систем и управляет реальной машиной, лежащей в основе виртуальной машины. Она предоставляет каждому пользователю доступ к устройствам реальной машины: процессору, оперативной памяти накопителям и устройствам ввода-вывода. ОС управляет одновременно

несколькими виртуальными машинами, а не отдельными задачами или процессами.

Виртуальные машины, работающие под управлением ОС, работают почти так же, как и реальные их аналоги, но медленнее, поскольку ОС выполняет операции одновременно множества виртуальных машин.

Способность одновременно выполнять несколько операционных систем имеет множество применений. Во-первых, это облегчает переход между различными операционными системами или различными версиями одной системы. Она позволяет пользователям обучаться одновременно с выполнением рабочих задач. Клиенты могут исполнять одновременно разные операционные системы.

Процессы, выполняющиеся на виртуальных машинах, управляются не ОС, а операционными системами, работающими на виртуальных машинах. Эти операционные системы выполняют все свои обычные функции, включая управление устройствами хранения, планирование загрузки процессора, управление вводом/выводом, защиту пользователей друг от друга, защиту операционной системы от пользователей, мультипрограммирование, управление процессами и задачами, обработку ошибок и так далее.

Пользователь виртуальной машины оперирует эквивалентом целой реальной машины, а не набором ресурсов, который он разделяет с множеством других пользователей.

2.2 Программа виртуализации Oracle VM VirtualBox

Данная программа виртуализации фирмы Oracle предназначена для создания виртуальных машин, установки на этих машинах операционных систем и работы в среде виртуальных ОС.

Гостевая операционная система в VirtualBox может быть установлена либо с использованием дистрибутива, либо импортом готовой конфигурации, сгенерированной ранее операционной системы.

2.3 Дистрибутив ОС Альт Рабочая станция

Альт Рабочая станция – дистрибутив на базе ядра Linux, включающий в себя операционную систему и набор приложений.

Альт Рабочая станция предназначена для установки как на реальные, так и на виртуальные машины.

Дистрибутив выпускается в нескольких вариантах исполнения:

- для процессоров x86-64(Intel 32 и 64-битных);
- для процессоров ARM64(Huawei Kunpeng Desktop, NVIDIA Jetson Nano, Raspberry Pi 3/Pi 4, mini-ITX на процессоре «Байкал-М»)
- процессоров e2k/e2kv4(«Эльбрус»)

Альт Рабочая станция может быть использована в качестве тонкого клиента в Alt-домене.

Дистрибутив Альт Рабочая станция включён в Единый реестр российских программ для электронных вычислительных и баз данных.

2.4 Установка операционной системы Альт Рабочая станция

Для установки системы необходимо воспользоваться носителем, содержащим дистрибутив операционной системы.

Одним из способов запустить установку ОС - загрузить компьютер с носителя DVD с дистрибутивом.

После загрузки будет произведена установка базовой системы на жёсткий диск. При этом программа установки работает с образом системы, загруженным в оперативную память компьютера.

После инициализации оборудования, будет запущен графический интерфейс программы-установщика.

Процесс установки разделён на отдельные шаги.

2.5 Загрузка операционной системы. Уровни выполнения

Загрузка установленной операционной системы производится с жёсткого диска после запуска компьютера и отработки программ BIOS.

Основные шаги загрузки операционной системы

- **загрузка ядра**
- **подключение (монтирование) файловых систем**
- **запуск системных служб**

После загрузки ядра системы и монтирования корневой файловой системы, ОС выполняет первую программу - **init**. Эта программа отвечает за запуск системных стартовых сценариев (скриптов) и за изменение уровней выполнения, а также устанавливает другие стартовые и заключительные процедуры.

Уровни выполнения (**runlevel**) определяют режим, в котором выполняется ОС.

Каждый из уровней определяет процессы, которые будут включены или выключены в зависимости от запущенного уровня выполнения.

С момента загрузки ОС выполняется в одном из режимов, но есть возможность переключаться между уровнями во время работы ОС.

В зависимости от того, какие службы нужно включить, а какие выключить, ОС меняет уровни выполнения.

Уровни выполнения дают администраторам повышенный контроль над системой.

Режим, в котором работает система, может быть изменен, как и сервисы, которые выполняются в этом режиме. Это позволяет полностью контролировать, к каким службам система будет иметь доступ в данный момент.

В ОС **Alt Linux** семь уровней выполнения, которые нумеруются от 0 до 6:

- **runlevel 0 (poweroff)** завершение работы и отключение системы

- **runlevel 1 (rescue)** однопользовательский режим работы. Чаще всего используется в целях обслуживания и выполнения других административных задач (оболочка восстановления).
- **runlevel 2 (multi-user)** многопользовательский режим работы без поддержки сетевых служб (демонов).
- **runlevel 3 (multi-user)** многопользовательский режим с поддержкой сети, но без графического интерфейса.
- **runlevel 4 (multi-user)** многопользовательский режим с поддержкой сети
- **runlevel 5 (graphical)** этот режим схож с уровнем 3, но тут еще запускается графический интерфейс.
- **runlevel 6 (reboot)** перезагрузка операционной системы.

Выбор режима выполнения системы зависит от задачи, выполняемой операционной системой.

В основном используется один из двух режимов выполнения: или **runlevel 3**, или **runlevel 5**.

Если необходимо запустить систему в режиме обслуживания, то используется **уровень 1**. В этом режиме в системе только один пользователь, сетевые службы не запущены. Это позволит выполнить обслуживания без сбоя.

В редких случаях появляется необходимость использовать уровень выполнения 4. Это может быть только в том случае, если администратору нужен уровень выполнения для особых задач.

Запускать систему с уровнем 0 и 6 невозможно, но можно переключаться на них если нужно выключить или перезагрузить систему. Но в этом нет особой необходимости, так как есть команды, которые выполняют эти операции.

Загрузка системы заканчивается тем, что программа **init** запускает копии программы **getty**, которые в свою очередь выдают приглашения **login** для всех виртуальных консолей.

Выполнив действия по инициализации системы программа **init** переходит в фоновый режим

После загрузки ОС Альт Рабочая станция 10.0 по умолчанию переходит на уровень выполнения **runlevel 5 (graphical)**

2.6 Графический режим выполнения Alt Linux

В состав ОС Альт Рабочая станция входит графическая оболочка **MATE**. Графическая оболочка включает в себя

- графический интерфейс программ, используемых для управления настройками ОС, установки и удаления приложений;
- графический интерфейс установленных в ОС приложений.

При загрузке в графическом режиме работа загрузчика ОС заканчивается переходом к окну входа в систему.

Для регистрации в системе необходимо выбрать имя пользователя ввести пароль, затем нажать **Enter** или щелкнуть на кнопке **Войти**. После этого запустится графическая оболочка операционной системы.

Так как работа в системе с использованием учётной записи администратора системы небезопасна, вход в систему в графическом режиме для суперпользователя root запрещён.

2.7 Консольный режим выполнения Alt Linux

2.7.1 Виртуальные консоли

При загрузке ОС в многопользовательском режиме (уровень выполнения 2,3 или 5) инициализируются шесть виртуальных консолей (**tty1** – **tty6**) с входом в систему в консольном режиме.

Каждая консоль позволяет входить в систему с использованием разных учётных записей.

Двенадцатая виртуальная консоль (**tty12**) выполняет функцию системной консоли – на нее выводятся сообщения о происходящих в системе событиях.

Каждой виртуальной консолю управляют отдельные процессы **getty**.

После входа пользователя в консоль, процесс **getty** обрабатывает вход в систему и запускает интерпретатор **bash**.

2.7.2 Командные процессоры (интерпретаторы команд)

Консольный режим операционной системы включает в себя взаимодействие с операционной системой с помощью интерфейса командной строки.

Для интерпретации и выполнения команд пользователя в интерфейсе командной строки предназначены командные процессоры (интерпретаторы команд).

В **Unix** и **Linux** - системах существует несколько командных процессоров. В ОС **Alt Linux** интерпретатором команд по умолчанию является командный процессор **bash** (Bourne Again Shell), наиболее распространённый интерпретатор, имеющий удобный интерфейс.

Интерпретатор команд, выполняя команду из командной строки, запускает нужную утилиту в качестве дочернего процесса, дожидается окончания её работы анализирует результат и переходит в режим ожидания следующей команды.

Запущенная утилита получает от родительского процесса (интерпретатора команд) информацию двух типов:

- **параметры командной строки**

- **переменные среды окружения** (переменные и их значения, которые были определены в окружении родительского процесса – интерпретатора команд).

2.7.3 Внутренние и внешние команды интерпретатора команд

Внутренние команды – это команды, встроенные непосредственно в программу интерпретатор команд. Выполнение таких команд происходит быстрее, так как у интерпретатора нет необходимости искать путь к команде в переменной среды окружения PATH, а также не порождается новый процесс.

Внешние команды – команды, которые не встроены в интерпретатор, а их исполняемые модули находятся в одном из системных каталогов. Когда должна быть выполнена внешняя команда, то интерпретатор ищет ее путь, указанный в переменной PATH, порождается новый процесс и происходит выполнение команды.

Обычно внешние команды расположены в каталогах **/bin** или **/usr/bin**.

Большинство команд интерпретатора являются внешними

Интерпретатор команд - это тоже внешняя команда, но обладающая собственным набором внутренних команд.

Если команда существует и как внутренняя команда интерпретатора, и как внешняя, интерпретатор будет присваивать высший приоритет своей собственной внутренней команде с тем же именем.

2.7.4 История выполнения команд интерпретатора

В **bash**, как и в других интерпретаторах команд, имеется такая встроенная функция, как история, позволяющая просматривать, изменять и повторно использовать команды, которые выполнялись ранее.

Она может оказаться очень полезной, поскольку многие команды **Linux** длинные и сложные.

Чтобы просмотреть всю историю, необходимо ввести команду **history**. Для отображения определенного количества предыдущих команд в истории

необходимо указать после **history** нужное число.

Для перемещения среди команд в истории используются клавиши «**вверх**» и «**вниз**».

Как только команда отобразится на экране, можно воспользоваться клавиатурой для редактирования текущей команды так же, как и любой другой, посредством клавиш «влево», «вправо», **Delete**, **Backspace** и т. д.

3 Выполнение работы

3.1 Задание

1. Создать виртуальную машину в среде менеджера виртуальных машин **Oracle VM VirtualBox**.
2. Установить гостевую операционную систему **Alt Linux**
3. Ознакомиться с основными возможностями графической оболочки **MATE** операционной системы **Alt Linux**. Выполнить операции в соответствии с заданием.
4. Ознакомиться с консольным режимом выполнения и интерфейсом командного интерпретатора **bash** операционной системы **Alt Linux**.
5. Выполнить команды интерпретатора **bash** в соответствии с заданием.

3.2 Порядок выполнения работы

- Войти в систему под учётной записью **stud_XX**, где **XX** - индекс группы.
Пароль **studXX**
- Запустить программу менеджера виртуальных машин **Oracle VM VirtualBox**.
- Выполнить настройку папки для установки виртуальных машин:
d:\Users\studXX\VirtualBox VMs

3.2.1 Создание виртуальной машины *alt-10*

Процесс создания виртуальных машин и установки гостевых ОС описан в пункте 6.1 приложения

Имя: **alt-10**

Тип: **Linux**

Версия: **Linux 2.6 / 3.x / 4.x(64-bit)**

Объём памяти **2048 МБ**

Создать новый виртуальный жёсткий диск

Тип **VDI (VirtualBox Disk Image)**

Формат хранения **Динамический виртуальный жёсткий диск**

Размер файла **25,00 ГБ**

3.2.2 Настройка виртуальной машины *Alt-10*

1. Подключить образ оптического диска с дистрибутивом операционной системы:

- Контроллер **IDE**

- Привод: **Первичный мастер IDE**

Выбрать файл образа **D:\ОС\Дистрибутивы\alt-workstation-10.0-x86_64.iso**

2. Установить порядок загрузки системы

Порядок загрузки: **CD/DVD**

Жёсткий диск

3.2.3 Установка операционной системы *Альт Рабочая станция*

1. Запустить виртуальную машину **alt-10**

2. Установить операционную систему **Альт Рабочая станция** с дистрибутива

3.2.4 Работа с ОС *Альт Рабочая станция* в графическом режиме

1. Войти в систему под именем **admin_kaf**, пароль **adminkaf**

2. Ознакомиться с основными элементами графической оболочки **MATE** и освоить основные приёмы запуска программ.

3. Установить уровень выполнения при загрузке системы по умолчанию равным **3 (multi-user)**:

- открыть программу «Терминал»
- перейти в режим администратора системы **root**
 - переключиться на консоль **tty2**
 - ввести login и Password: **root/adminroot**
 - выполнить команду для установки уровня выполнения **3 (multi-user - многопользовательский режим с поддержкой сети без графического интерфейса)** при загрузке системы
 - перезагрузить систему

После перезагрузки система по умолчанию перейдёт в консольный режим выполнения

3.2.5 Работа с ОС *Альт Рабочая станция* в консольном режиме

1. Войти в систему, используя виртуальные консоли:

- Виртуальная консоль **tty1** – администратор системы **root/adminroot**
- Виртуальная консоль **tty2** – системный пользователь **admin_kaf/adminkaf**

2. Отобразить все процессы, связанные с консолями

3. Отобразить информацию о процессах системы

4 Контрольные вопросы

1. В чём различие между физическими и виртуальными ресурсами?
2. Что такое виртуальная машина?
3. В чём отличие мультипрограммных систем и систем виртуальных машин?
4. Для чего предназначена программа VirtualBox?

5. Как может быть установлена гостевая операционная система?
6. Учётные записи каких пользователей создаются при установке операционной системы Alt Linux&

5 ЛИТЕРАТУРА

1. Х.М. Дейтел, П. Дж. Дейтел, Д.Р. Чофнес Операционные системы. Часть 1. Основы и принципы: Третье издание. Пер. с англ. – М.: ООО «Бином-Пресс», 2011 г. – 1024 с.: ил.
2. Робачевский А.М. Операционная система UNIX.-СПб.: БХВ-Петербург, 2001. – 528 с.:ил.
3. Семкин П.С., Семкин А.П., Горячкин Б.С. Лабораторный практикум по дисциплине «Операционные системы». Часть 1. ОС Alt Linux. Управление пользователями. Администрирование дисковой подсистемы: Учебно-методическое пособие. –М.: Издательство «Спутник+», 2023. -78 с.
4. Документация для ОС «Альт Рабочая станция». Режим доступа: <https://www.basealt.ru/alt-workstation/docs>

6 Приложение

6.1 Создание виртуальных машин и установка гостевых ОС в программе виртуализации VirtualBox

6.1.1 Создание виртуальной машины и установка гостевой ОС с использованием дистрибутива.

Выбрать пункт меню «Создать»

Задать:

- **имя** новой виртуальной машины
- **тип** устанавливаемой гостевой системы;
- определить количество выделяемой ей **оперативной памяти**;

- создать **виртуальный диск** (фиксированного размера или динамически расширяющийся).

После создания виртуальной машины производится обычная установка гостевой операционной с **установочного диска** или образа **ISO дистрибутива**.

6.1.2 Создание виртуальной машины и установка гостевой ОС путем импорта готовой конфигурации

Выбрать пункт меню **«Импорт»**

Выбрать устанавливаемую конфигурацию в открытых форматах виртуализации **OVA** или **OVF**.

Открывается окно **«Укажите параметры импорта»**

Перечисляются устройства импортируемой конфигурации. Некоторые параметры можно изменить или отключить

После этого производится создание новой виртуальной машины и импорт выбранной конфигурации

6.2 Установка операционной системы Alt Linux

Установка начинается с меню, в котором перечислены несколько вариантов установки.

Выбрать **Установка**

1/12: Язык

Выберите язык:

Русский

Выберите вариант переключения раскладки клавиатуры

➤ **Далее**

2/12: Лицензионный договор

Лицензионный договор

✓ Да, я согласен с условиями

➤ **Далее**

3/12: Дата и время

Выберите страну

Выберите часовой пояс

➤ **Далее**

4/12: Подготовка диска

Разделы

Имя устройства	Файловая система	Размер[свободно]
sda	Unknown	25 GB[25 GB]

Выберите метод установки

- Использовать неразмеченное пространство
 - Удалить все разделы и создать разделы автоматически
 - Подготовить разделы вручную

➤ **Далее**

5/12: Установка системы

Дополнительные приложения:

✓ **Офис**

- ✓ Антивирус ClamAV
 - Управление проектами (диаграммы Гантта)
- ✓ Словари
- ✓ Офисный пакет (LibreOffice)
 - Поддержка сканеров
 - Поддержка смарт-карт

✓ **Интернет/сети**

- Поддержка сторонних приложений (1С, CryptoPro...)
- ✓ Инструменты управления групповыми политиками
- ✓ Агенты инфраструктуры (Zabbix, Bacula...)
- ✓ Клиент облачного хранилища (Owncloud, Seafiler)
- ✓ Клиент домена FreeIPA
- ✓ Файлообменные сети

- ✓ Общий доступ к папкам
- ✓ Обмен мгновенными сообщениями (Pidgin)
- ✓ Электронная почта (Thunderbird)
- ✓ Интернет-телефония (Linphone)

Мультимедиа

✓ Виртуализация

- ✓ Запуск программ Windows/DOS
- ✓ Сервер виртуализации (KVM)
- ✓ Доступ к удалённым рабочим столам
- ✓ Настольная виртуализация (VirtualBox)

Издательство

Требуемое место на диске **9548 Мб**

➤ **Далее**

5/12: Установка системы

Установка базовой системы

Установка программного обеспечения

Установка дополнительного программного обеспечения

➤ **Далее**

6/12 Сохранение настроек

➤ **Далее**

7/12 Установка загрузчика

Устройство **Жёсткий диск sda, ATA VBOX HARDDISK, 25.00 Gb**

Пароль на загрузчик (имя пользователя: boot)

Установить или сбросить пароль

➤ **Далее**

8/12: Настройка сети

Имя компьютера **alt-10**

Интерфейсы

enp0s3 Сетевая карта: Intel Corporation 82540EM Gigabit Ethernet Controller
провод подсоединён
MAC: 08:00:27:14:58:14
Интерфейс ВКЛЮЧЕН

Конфигурация Использовать DHCP
Дополнительно

Интерфейс enp0s3

Сетевая подсистема NetworkManager(etcdns)

Запускать интерфейс при загрузке системы

➤ Далее

9/12: Администратор системы

В окне необходимо ввести пароль учетной записи администратора (root).

Администратор системы отличается от всех других пользователей тем, что ему разрешены любые действия.

Укажите пароль для системного администратора

Создать автоматически

adminroot (введите фразу)

adminroot (повторите фразу)в

➤ Далее

10/12: Системный пользователь

Кроме администратора (root) в системе необходимо иметь хотя бы одного **системного пользователя**, имеющего ограниченные полномочия по сравнению с администратором, но который имеет права на выполнение некоторых системных функций.

Новая учётная запись пользователя

Имя **admin_kaf**

Комментарий

Пароль Создать автоматически

adminkaf (введите фразу)

adminkaf (повторите фразу)

Автоматический вход в систему

➤ Далее

12/12: Завершение установки

Информация о дистрибутиве

➤ Завершить

6.3 Команды переключение между виртуальными консолями

- При использовании программы «Терминал» графического режима для переключения на соответствующую консоль используется комбинация клавиш:

Ctrl+Alt+FN (F1 - F6)

- При работе в консольном режиме используется следующая комбинация:
Alt+FN (F1 - F6)

Примечание. Если используется клавиша **Fn**, то в комбинацию клавиш включается и клавиша **Fn**

6.4 Командный интерпретатор BASH

6.4.1 Запуск интерпретатора команд

При отсутствии графического интерфейса, после загрузки системы сразу попадают в командную строку консольного режима, в которой отображается строка-приглашение интерпретатора команд.

При использовании графического режима самый распространённый способ доступа к интерпретатору – использование программы «Терминал».

После запуска интерпретатора отображается строка-приглашение для ввода команды.

Строка-приглашение имеет вид

< имя текущего пользователя >@< имя компьютера > ~ \$

~ означает, что текущим является домашний каталог пользователя

\$ обозначает обычного пользователя

Пример:

admin_kaf @ alt-10 ~ \$

Приглашение для пользователя **root** имеет вид

< имя компьютера > ~ #

Пример:

alt-10 ~ #

6.4.2 Получение информации о внутренних и внешних командах

В интерпретаторе **bash**, получить список внутренних команд интерпретатора можно получить с помощью команды **help**:

\$ help

Узнать о конкретной команде, является ли она внутренней или внешней, можно с помощью команды **type [name]**:

\$ type cat

cat is /bin/cat

\$ type cd

cd is a shell builtin

6.4.3 Команды получения общей информации о системе

\$ help help – о том, как получать справку.

\$ man – (manual) вызов справки команд (**man –help** , **man <команда>**).

\$ info – информация о командах (**info <команда>**).

\$ bash – запуск нового командного интерпретатора.

\$ uname -a – показать тип ОС и характеристики (все ключи **-animporsv**).

\$ who и **\$ w** – показать подключенных пользователей

\$ env – показать переменные среды интерпретатора команд

\$ clear – очистить страницу экрана командной строки

\$ exit – завершение текущего командного интерпретатора или скрипта.

6.4.4 Переменные среды окружения интерпретатора команд

Среда окружения (environment) - пространство, в котором можно задавать переменные (**переменные окружения**), в которые можно записывать информацию (**присваивать значение переменной окружения**), а после эту информацию считывать (**подставлять значение переменной**).

Можно явно изменить окружение для запускаемого процесса, присвоив некоторое значение переменной окружения в командной строке перед именем команды. Это можно сделать с помощью операции «=**=**» внутри первого слова командной строки

Переменные, которые командный интерпретатор **bash** определяет после запуска, не принадлежат окружению, и не наследуются дочерними процессами. Чтобы переменная попала в окружение, её надо экспортировать командой **export**

Во время сеанса работы пользователя командный интерпретатор к полученному окружению добавляет и собственные настройки.

Большинство заранее определённых переменных используются либо самой командной оболочкой, либо утилитами системы.

Просмотреть окружение в **bash** можно с помощью команды **env**.

К значению любой переменной окружения в **bash** можно обратиться по имени: вместо конструкции **\$имя_переменной** будет подставлено значение этой переменной.

Чтобы просмотреть значение некоторой переменной, можно ввести команду **echo \$имя_переменной**.

\$ echo \$PATH будет выведено значение переменной окружения **PATH**

В переменной окружения **PATH** содержится список путей к каталогам, в которых интерпретатор будет искать заданную внешнюю команду.

После открытия интерпретатора команд, его среда окружения

конфигурируется исходя из того, какой пользователь запустил данный интерпретатор.

Настройки интерпретатора команд **bash** для всех пользователей располагаются в нескольких файлах. Можно сделать так, чтобы собственные версии этих файлов переопределяли системные настройки.

Эти настройки содержатся в файлах двух типов - **файлах запуска** и **файлах инициализации**.

bash задействует **файлы запуска** в случае с любым интерпретатором команд, который является тем, что назначается при входе в систему. Эти файлы определяют настройки, которые будут применяться при каждом входе в систему

bash задействует **файлы инициализации** в случае с интерпретаторами команд, которые работают в интерактивном режиме, то есть без запуска сценария интерпретатора команд.

bash применяет общесистемный файл запуска **/etc/profile**, а также некоторые файлы с точкой из домашнего каталога пользователя для индивидуальных настроек (при наличии таковых): **.bash_profile**, **.bash_login** и **.profile**. Он также использует расположенные в каталоге **/etc/profile.d** сценарии, имена которых оканчиваются на **.sh**.

bash задействует общесистемный файл инициализации **/etc/bash.bashrc**, а также файл **.bashrc** из домашнего каталога (для индивидуальных настроек). Эти файлы задействуются при каждом новом открытии интерпретатора команд **bash**.

При выходе из интерпретатора команд, назначаемого при входе в систему (например, из виртуальной консоли), будут выполнены все команды, указанные в файле **~/. bash_logout**.

Модификация настроек в этих файлах приведет к перманентному изменению настроек интерпретатора команд пользователя, однако не повлияет на интерпретаторы команд, которые уже запущены (другие интерпретаторы команд используют другие конфигурационные файлы).

Есть множество способов, посредством которых можно просматривать и изменять среду интерпретатора команд.

6.5 Изменение уровней выполнения ОС в системе инициализации `systemd`

6.5.1 Уровни выполнения операционной системы

Текущий уровень выполнения системы определяется командой

\$ runlevel

Команда выводит две цифры. Первая указывает на предыдущий уровень выполнения, а вторая - на текущий.

Для работы с уровнями выполнения используется **systemd** - программный пакет, предоставляющий набор системных компонентов для операционных систем **Linux**. Его основная цель - унифицировать конфигурацию и поведение служб в различных дистрибутивах **Linux**.

Основным компонентом **systemd** является "system and service manager" - система инициализации, используемая для загрузки пользовательского пространства и управления пользовательскими процессами.

Уровни выполнения системы:

- **runlevel 0 = poweroff.target (runlevel0.target)**
- **runlevel 1 = rescue.target (runlevel1.target)**
- **runlevel 2 = multi-user.target (runlevel2.target)**
- **runlevel 3 = multi-user.target (runlevel3.target)**
- **runlevel 4 = multi-user.target (runlevel4.target)**
- **runlevel 5 = graphical.target (runlevel5.target)**
- **runlevel 6 = reboot.target (runlevel6.target)**

Текущий уровень выполнения по умолчанию определяется командой

\$ systemctl get-default

Все уровни выполнения можно посмотреть с помощью команды


```
$ ls -l /lib/systemd/system/runlevel*
```

Чтобы поменять уровень выполнения по умолчанию, надо создать новую символическую ссылку

```
$ ln -sf /lib/systemd/system/runlevel3.target  
/etc/systemd/system/default.target
```

Данная командой поменяла режим выполнения системы по умолчанию с уровня выполнения 5, на 3 и при следующей загрузке система будет выполняться на этом уровне.

Ключ **f** указывает на то, что перед созданием новой символической ссылки целевой файл должен быть удален.

Чтобы проверить, какие службы запускаются во время загрузки системы в режиме графического интерфейса (уровень выполнения 5), надо выполнить:

```
$ systemctl list-dependencies graphical.target
```

Чтобы просмотреть список доступных служб другого уровня, просто надо заменить "**graphical.target**" на нужную.

Чтобы посмотреть по каким уровням выполнения запущена та или иная служба, можно ввести команду:

```
$ systemctl show -p WantedBy [name of service]
```

Каждому уровню выполнения соответствует каталог **/etc/rc.d/rcN.d**, где **N** - номер уровня работы.

В этих каталогах содержатся символические ссылки, указывающие на те стартовые сценарии, которые должны быть выполнены при переходе на соответствующий уровень. Стартовые сценарии предназначены для запуска, остановки или перезапуска системных служб (сервисов).

Все стартовые сценарии располагаются в каталоге **/etc/init.d**.

Любую службу, вписавшую в **/etc/init.d/** свой стартовый сценарий, можно запускать и останавливать вручную, вызвав этот сценарий с параметром **start** и **stop** соответственно. Для проверки состояния службы (работает - остановлена - сбойт) тот же сценарий можно запускать с параметром **status**.

То же самое делает и команда **service**: она вызывает сценарий (имя которого передаётся ей первым параметром) из **/etc/init.d/** и запускает его.

Чтобы перезапустить службу **httpd** можно использовать любую из команд:

```
/etc/init.d/httpd restart
```

```
service httpd restart.
```

Также происходит и регистрация стартовых сценариев на уровнях выполнения.

Если необходимо, чтобы при переходе на третий (сетевой) уровень запускалась служба **httpd**, надо организовать символическую ссылку на **/etc/rc.d/httpd** в каталоге **/etc/rc3.d/**. Ссылка эта должна иметь имя **S<номер>httpd**, чтобы быть запущенной в подходящее время.

Если не надо, чтобы **httpd** запускался на этом уровне, ссылка должна иметь имя **K<номер>httpd**.

Создание и удаление таких ссылок, показ их текущего состояния, можно выполнить с помощью команды

```
chkconfig.
```

Самые важные её ключи:

--list (просмотреть список зарегистрированных служб),

on (добавить запуск службы),

off (запретить запуск службы).

6.5.2 Изменение уровня выполнения системы во время сеанса

systemctl isolate graphical.target - установка уровня 5 (**graphical** - многопользовательский режим с поддержкой сети и графического интерфейса)

systemctl isolate multi-user.target - установка уровня 3 (**multi-user** - многопользовательский режим с поддержкой сети и без графического интерфейса)

systemctl reboot - перезагрузка

systemctl poweroff выключение

Примечание. В данной редакции ОС поддерживаются команды **init N** системы инициализации **SysV (SystemV)**, поэтому можно использовать команды

init 0 – выключение системы

init 6 – перезагрузка системы

init 3 – закрытие графического режима и переход в консольный режим

init 5 – переход в графический режим

6.5.3 Установка уровня выполнения системы по умолчанию при загрузке системы

Установка уровня 3 (**multi-user** - многопользовательский режим с поддержкой сети без графического интерфейса) при загрузке системы

systemctl set-default multi-user.target

Created symlink /etc/systemd/system/default.target -> /lib/systemd/system/multi-user.target.