ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
СЕВЕРО-КАВКАЗСКИЙ ФИЛИАЛ
ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО
БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО
ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И
ИНФОРМАТИКИ»

КАФЕДРА ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Ткачук Е.О.

ОПЕРАЦИОННЫЕ СИСТЕМЫ

Методическое пособие

для проведения лабораторных работ и практических занятий

Ростов-на-Дону

2019 г.

УДК 004.451
ББК 32.973.3-018.2
Т48
О60

Ткачук Е.О. Операционные системы: Методическое пособие для проведения лабораторных работ и практических занятий. - Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2019. – 160 с.: ил.

В пособии даются организационно-методические указания к лабораторным работам и практическим занятиям а также порядок выполнения и оформления отчётов.

Предназначено для студентов обеих форм обучения, изучающих дисциплину «Операционные системы», а также может быть полезно всем остальным студентам, желающим самостоятельно изучать современные Операционные системы.

Составители:

доцент кафедры ИВТ Ткачук Е.О.

Рецензент: доц. кафедры ИВТ СКФ МТУСИ, к.т.н. доц. А.Н. Чикалов.

Издание рассмотрено и утверждено
на заседании кафедры ИВТ
26.08.2019 года (протокол № 1)
Отв. редактор _____

Оглавление

1. Практическое занятие № 1 Установка ОС Ubuntu на Oracle VM VirtualBox

Цель работы:

- Приобретение навыков установки и создания виртуальных машин в Oracle VM VirtualBox
- Приобретение навыков установки и начальной настройки операционной системы Ubuntu

1.1.Задание:

1. Скачать или получить у преподавателя образ установочного диска ОС Ubuntu
2. Создать виртуальную машину в VirtualBox.
3. Установить ОС Ubuntu на созданную виртуальную машину.
4. В установленной операционной системе:
   -установить дополнения гостевой ОС
   -настроить рабочие столы (эффекты, изображения);
   -изменить раскладку клавиатуры по умолчанию;
   -определить тип сеанса загружаемую по умолчанию.
5. Ознакомиться и описать панель инструментов Ubuntu.

1.2.Краткие теоретические сведения

Занятия по дисциплине «Операционные системы проводятся с использованием технологии виртуализации.

Виртуализация — предоставление набора вычислительных ресурсов или их логического объединения, абстрагированное от аппаратной реализации, и обеспечивающее при этом логическую изоляцию друг от друга вычислительных процессов, выполняемых на одном физическом ресурсе.

Примером использования виртуализации является возможность запуска нескольких операционных систем на одном компьютере: при том каждый из экземпляров таких гостевых операционных систем работает со своим набором логических ресурсов (процессорных, оперативной памяти, устройств хранения), предоставлением которых из общего пула, доступного на уровне оборудования, управляет хостовая операционная система — гипервизор.

Благодаря использованию данной технологии у студента появляется возможность безопасно работать с различными гостевыми операционными системами, не подвергая опасности настройку хостовой операционной системы компьютера учебного класса.

В качества средства виртуализации будем использовать программное средство Oracle VM VirtualBox.

VirtualBox (Oracle VM VirtualBox) — программный продукт виртуализации для операционных систем Microsoft Windows, Linux, FreeBSD, macOS, Solaris/OpenSolaris, ReactOS, DOS и других. Использование виртуальной машины (ВМ) на домашнем ПК, прежде всего, позволит одновременно запускать несколько операционных систем (гостевые ОС).

К примеру, в данный момент на компьютере или ноутбуке установлен один из выпусков операционной системы Microsoft Windows (хостовая ОС). Установка же

виртуальной машины, в данном случае VirtualBox, позволяет использовать в среде хостовой ОС любые другие системы (гостевые), включая macOS, Linux, Android, Windows и так далее, вариантов здесь может быть очень много.

Программа была создана компанией Innotek с использованием исходного кода Qemu. Первая публично доступная версия VirtualBox появилась 15 января 2007 года. В феврале 2008 года Innotek был приобретён компанией Sun Microsystems, модель распространения VirtualBox при этом не изменилась. В январе 2010 года Sun Microsystems была поглощена корпорацией Oracle, модель распространения осталась прежней.

Ключевые возможности

Кроссплатформенность

Модульность

Поддержка USB 2.0, когда устройства хост-машины становятся доступными для гостевых операционных систем (только в проприетарной версии)

Поддержка 64-битных гостевых систем(начиная с версии 2.0), даже на 32-битных хост-системах (начиная с версии 2.1, для этого обязательна поддержка технологии виртуализации процессором)

Поддержка SMP на стороне гостевой системы (начиная с версии 3.0, для этого обязательна поддержка технологии виртуализации процессором)

Встроенный RDP-сервер, а также поддержка клиентских USB-устройств поверх протокола RDP (только в проприетарной версии)

Экспериментальная поддержка аппаратного 3D-ускорения (OpenGL, DirectX 8/9 (с использованием кода wine) (только в 32-битных Windows XP, Vista, 7 и 8, для гостевых DOS / Windows 3.x / 95 / 98 / ME поддержка аппаратного 3D-ускорения не предусмотрена)

Поддержка образов жёстких дисков VMDK (VMware) и VHD (Microsoft Virtual PC), включая snapshots (начиная с версии 2.1)

Поддержка iSCSI (только в проприетарной версии)

Поддержка виртуализации аудиоустройств (эмуляция AC97 или SoundBlaster 16 или Intel HD Audio на выбор)

Поддержка различных видов сетевого взаимодействия (NAT, Host Networking via Bridged, Internal)

Поддержка цепочки сохраненных состояний виртуальной машины (snapshots), к которым может быть произведён откат из любого состояния гостевой системы

Поддержка Shared Folders для простого обмена файлами между хостовой и гостевой системами (для гостевых систем Windows 2000 и новее, Linux и Solaris)[6]

Поддержка интеграции рабочих столов (seamless mode) хостовой и гостевой операционной системой

Поддержка формата OVF/OVA

Есть возможность выбора языка интерфейса (поддерживается и русскоязычный интерфейс)

Базовая версия полностью открыта по лицензии GNU GPL, соответственно нет ограничений в использовании

Пакет дополнений

VirtualBox Guest Additions — комплект программного обеспечения, устанавливаемый в гостевую операционную систему и расширяющий её возможности

по взаимодействию с системой виртуализации и хост-системой. Например, после установки специального драйвера «виртуальной видеокарты» становится возможным изменять разрешение рабочего стола гостевой ОС произвольным образом вслед за размером окна VirtualBox, в котором запущена виртуальная машина.

До версии 4.0.0 существовало две версии, различавшиеся по лицензии и функциональности. Начиная с 4.0.0 закрытые компоненты вынесены в отдельный пакет дополнений (Extension Pack):

Пакет дополнений содержит закрытые компоненты и распространяется под проприетарной лицензией PUEL (бесплатно только в персональных целях или для ознакомления):

RDP сервер — позволяет подключаться к виртуальной системе удалённо с помощью любого RDP-совместимого клиента;

Поддержка USB — позволяет передавать виртуальной машине USB устройства;

Intel PXE boot ROM — загрузка операционной системы по сети. Используется для создания тонких клиентов/бездисковых рабочих станций.

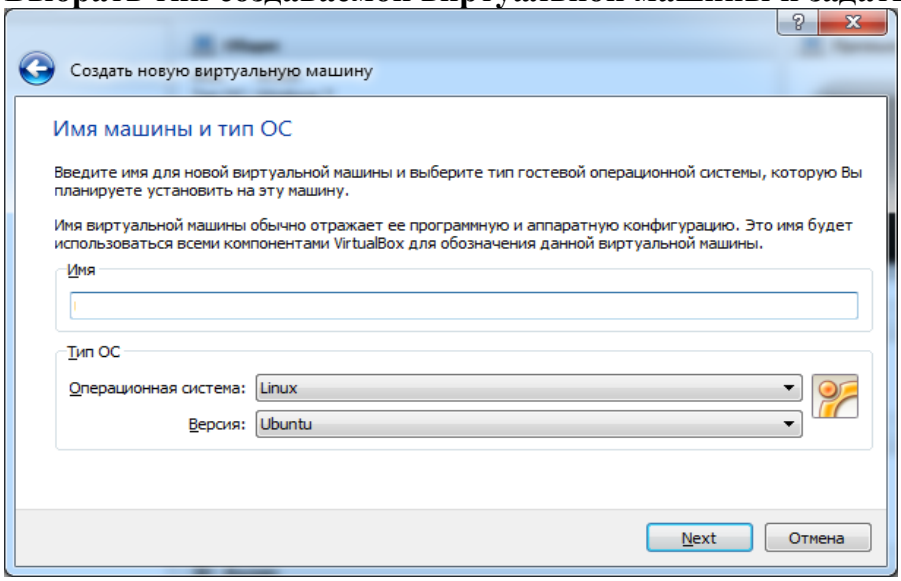
1.3.Ход работы:

Установка операционной системы Ubuntu производиться на виртуальную машину Oracle VM VirtualBox. Сначала необходимо установить Oracle VM VirtualBox и создать виртуальную машину.

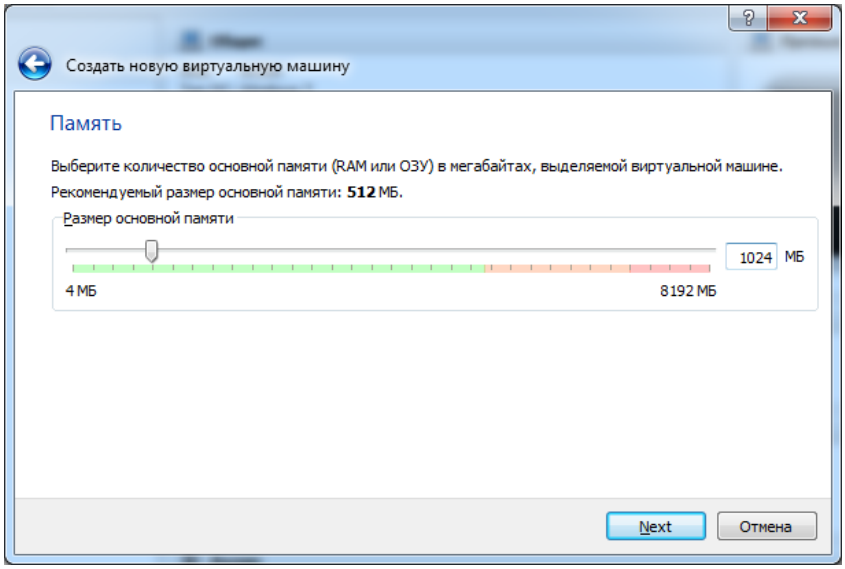
Установка Oracle VM VirtualBox производиться с настройками по умолчанию. После установки виртуальной машины установить пакет расширения (Oracle_VM_VirtualBox_Extension_Pack).

1)Создание виртуальной машины:

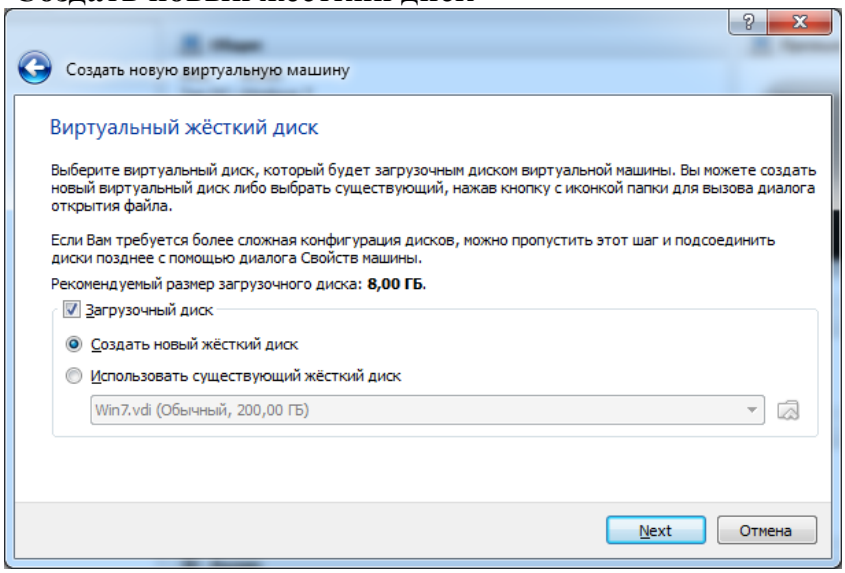
Выбрать тип создаваемой виртуальной машины и задать имя.



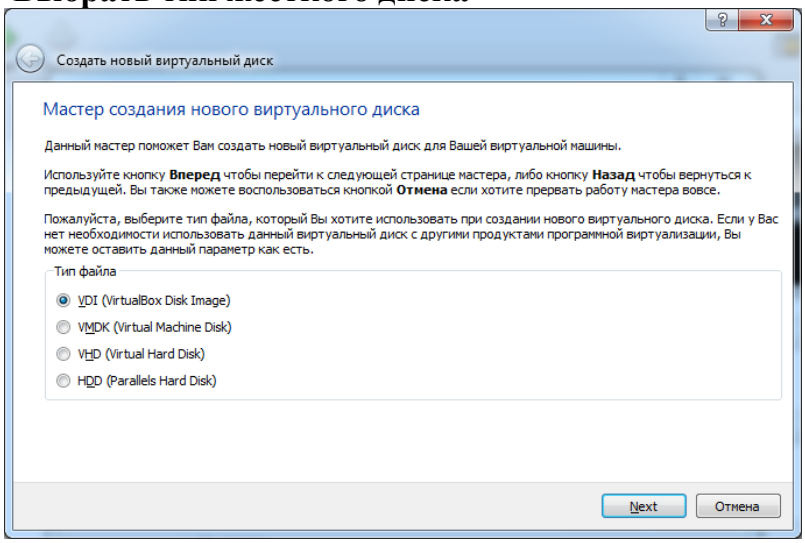
Установить объем памяти 1024 Мб



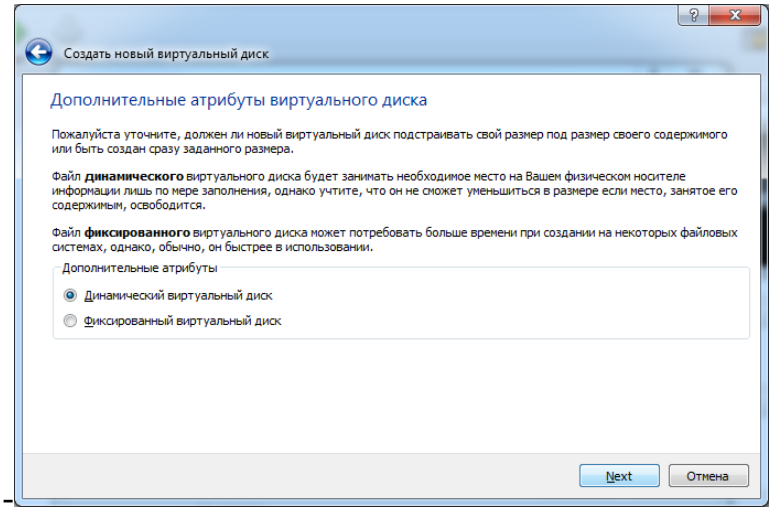
Создать новый жесткий диск



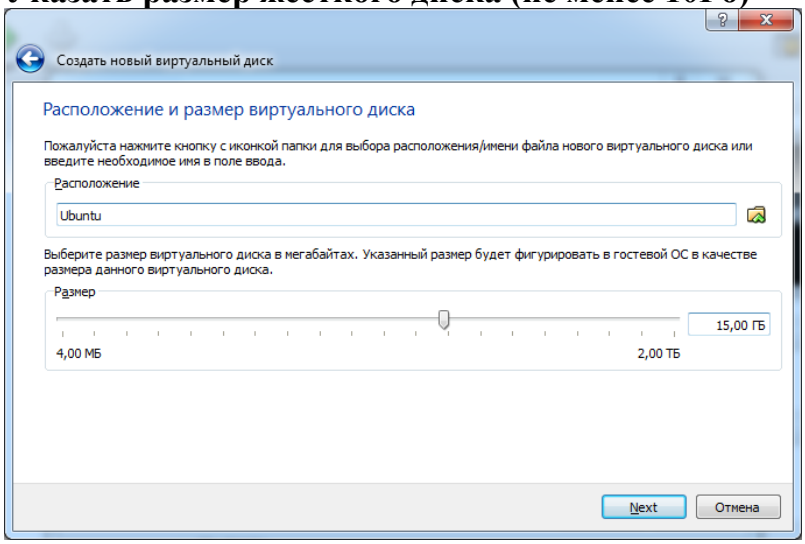
Выбрать тип жесткого диска



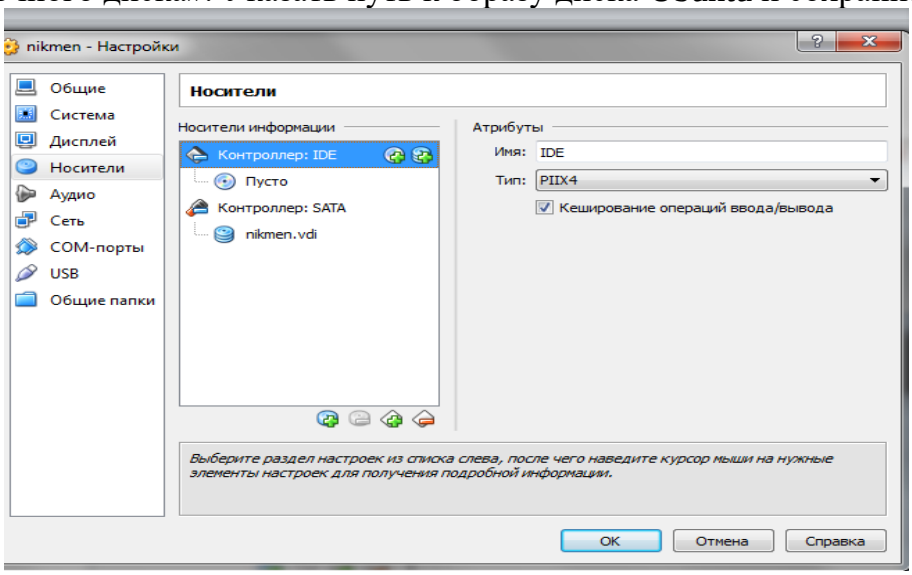
Выбрать атрибуты жесткого диска



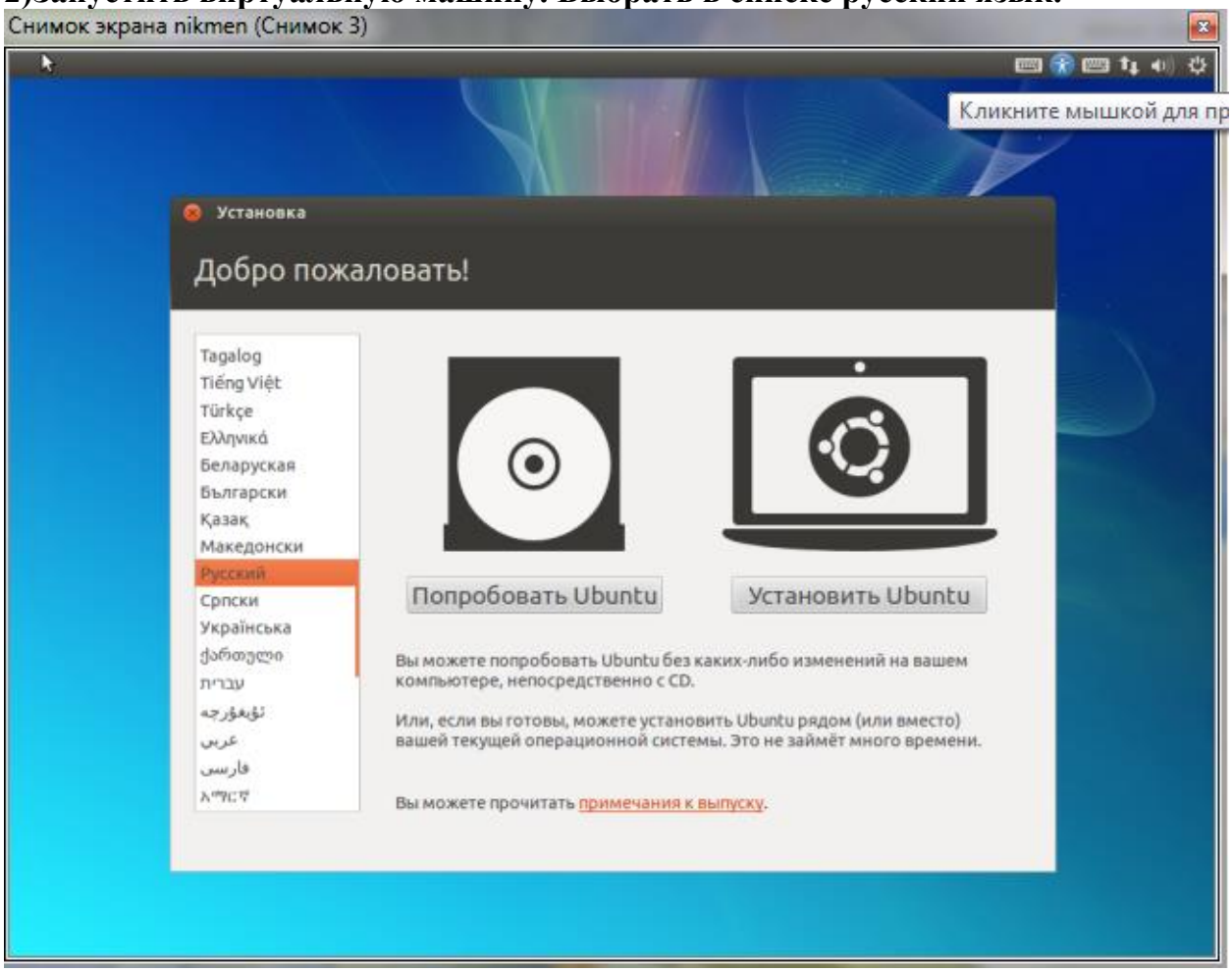
Указать размер жесткого диска (не менее 10Гб)



Дождаться установки виртуальной машины. Открыть окно свойств виртуальный машины и перейти в раздел носители. Выбрать IDE контроллер – пусто. Кликнуть по иконки диска в правой панели и выполнить «Выбрать образ загрузочного диска». Указать путь к образу диска Ubuntu и сохранить настройки.

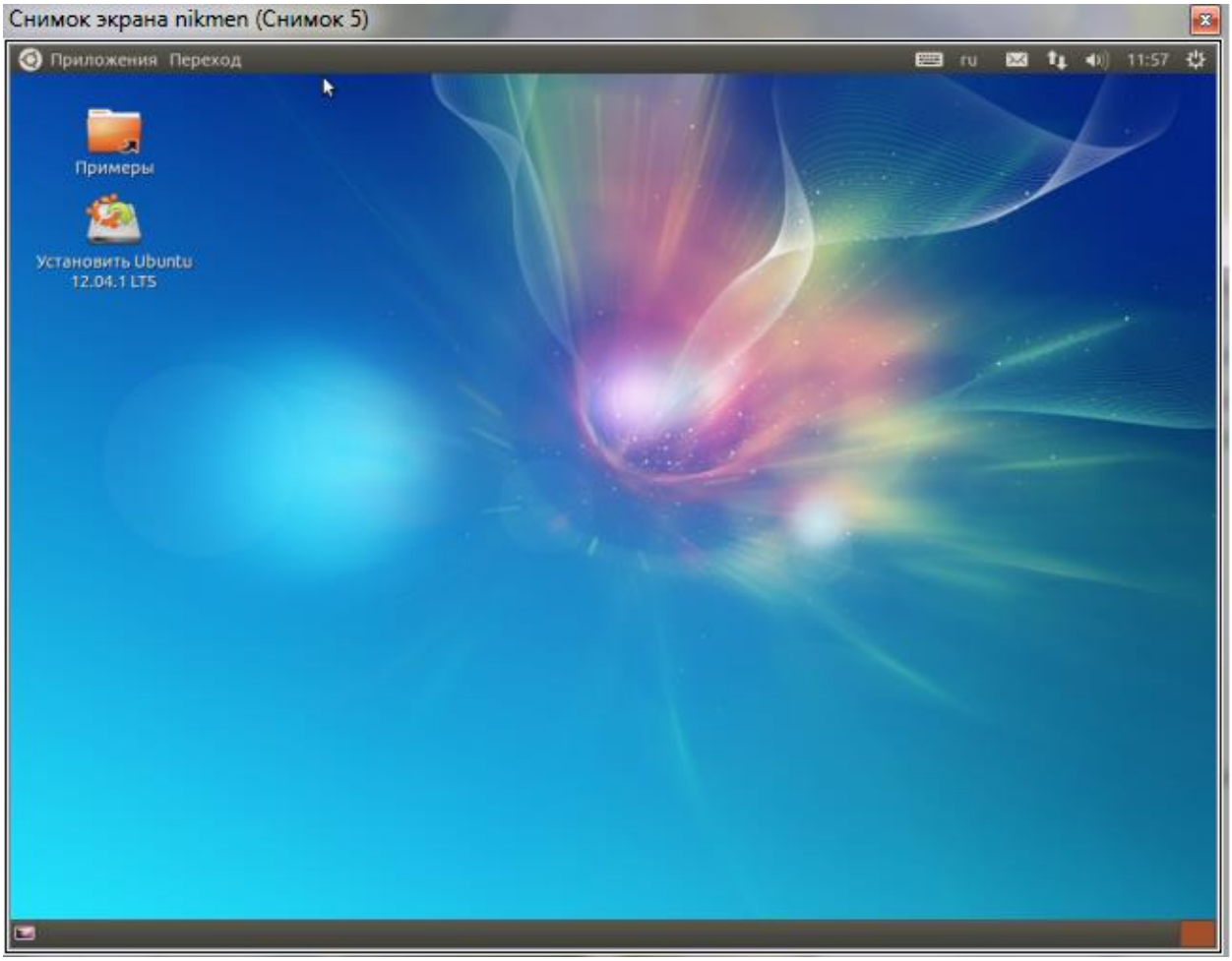


2)Запустить виртуальную машину. Выбрать в списке русский язык.

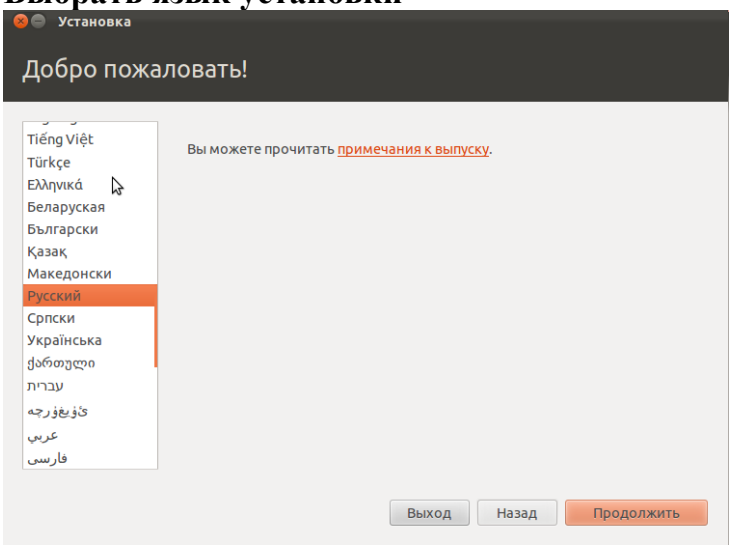


Из предложенных вариантов выбрать «Запустить Ubuntu без установки»

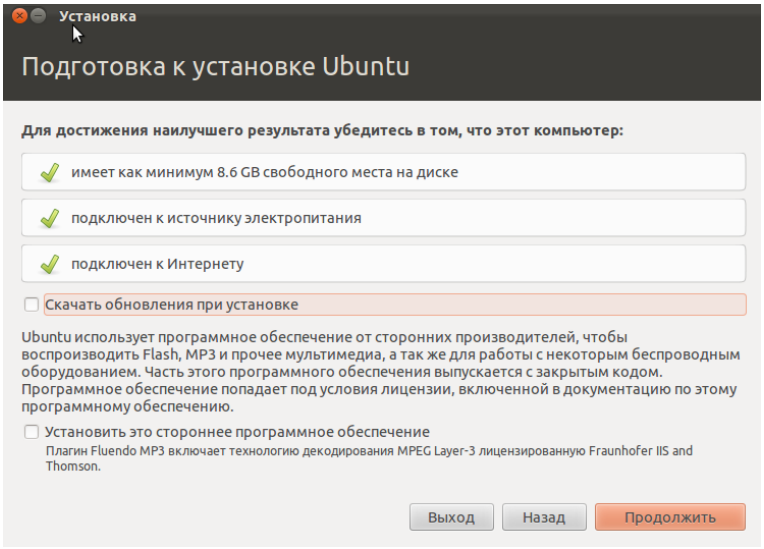
Дождаться загрузки операционной системы. Запустить установка кликнув по ярлыку «Установить Ubuntu»



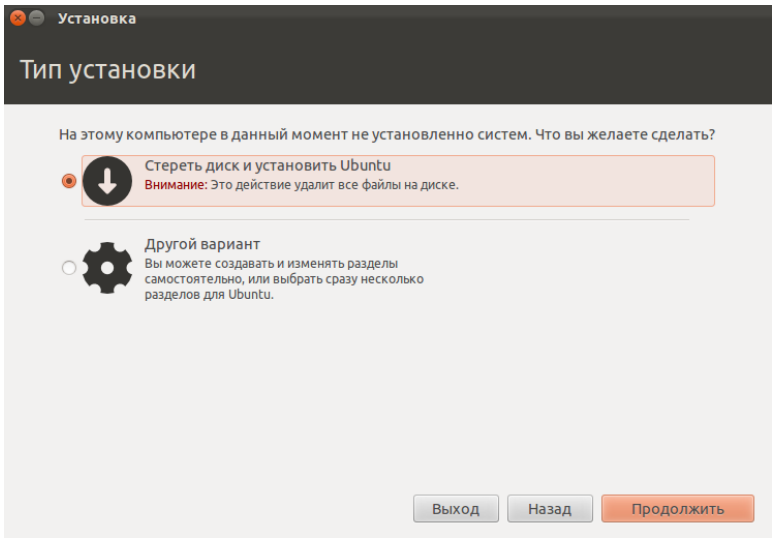
Выбрать язык установки



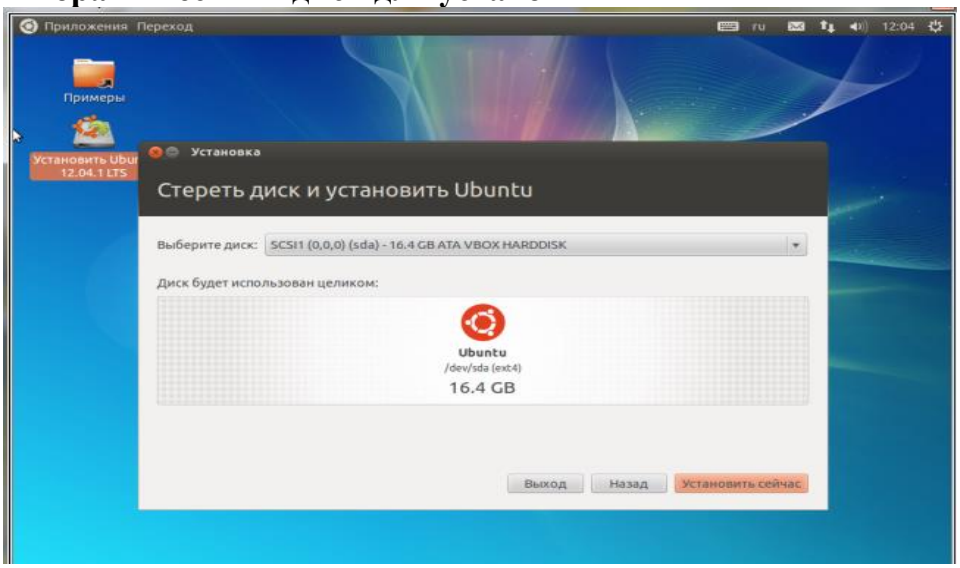
Подтвердить требования к установке



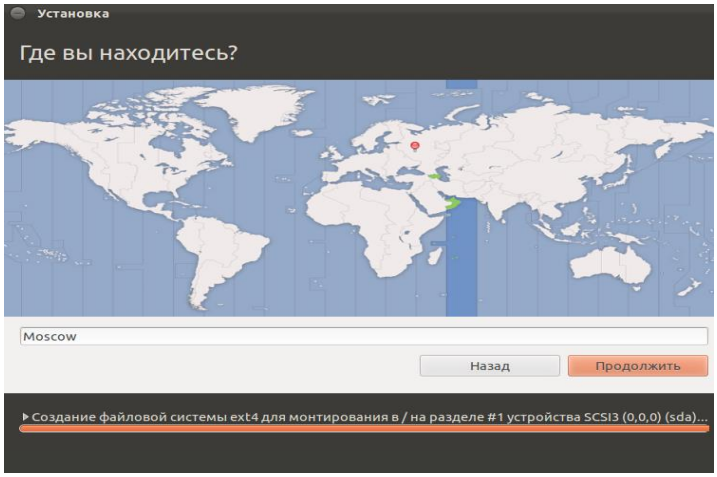
Выбрать тип установки



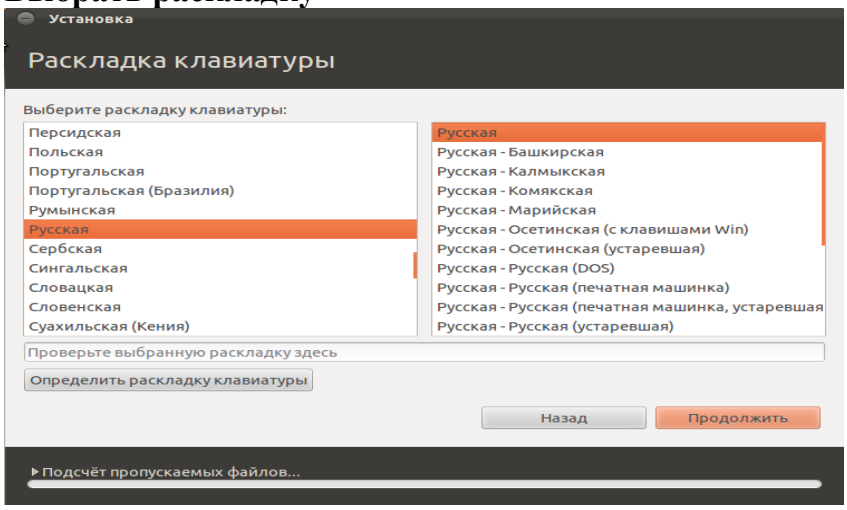
Выбрать жесткий диск для установки



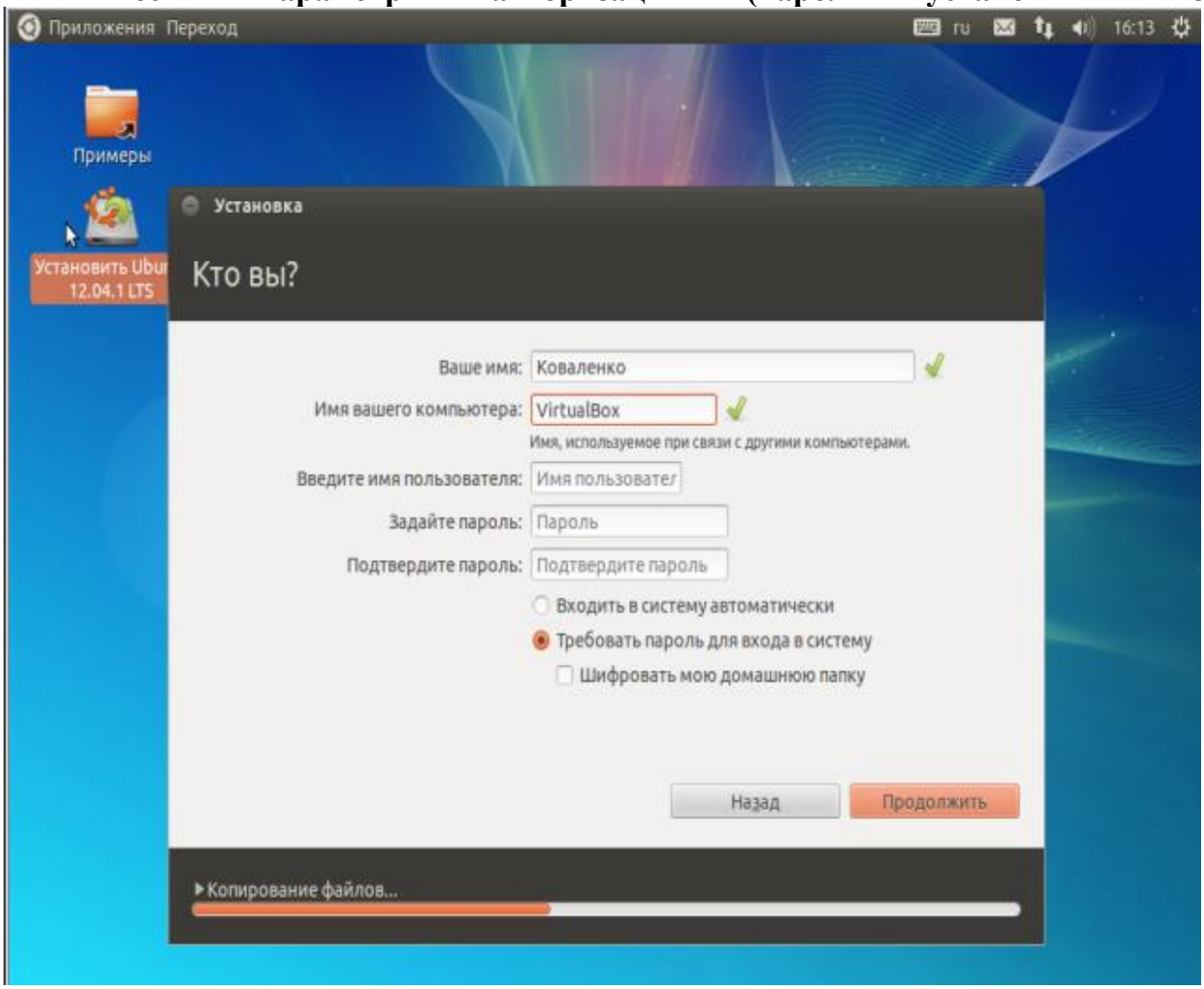
Указать региональные параметры



Выбрать раскладку



Ввести параметры авторизации (пароль установить 1234)

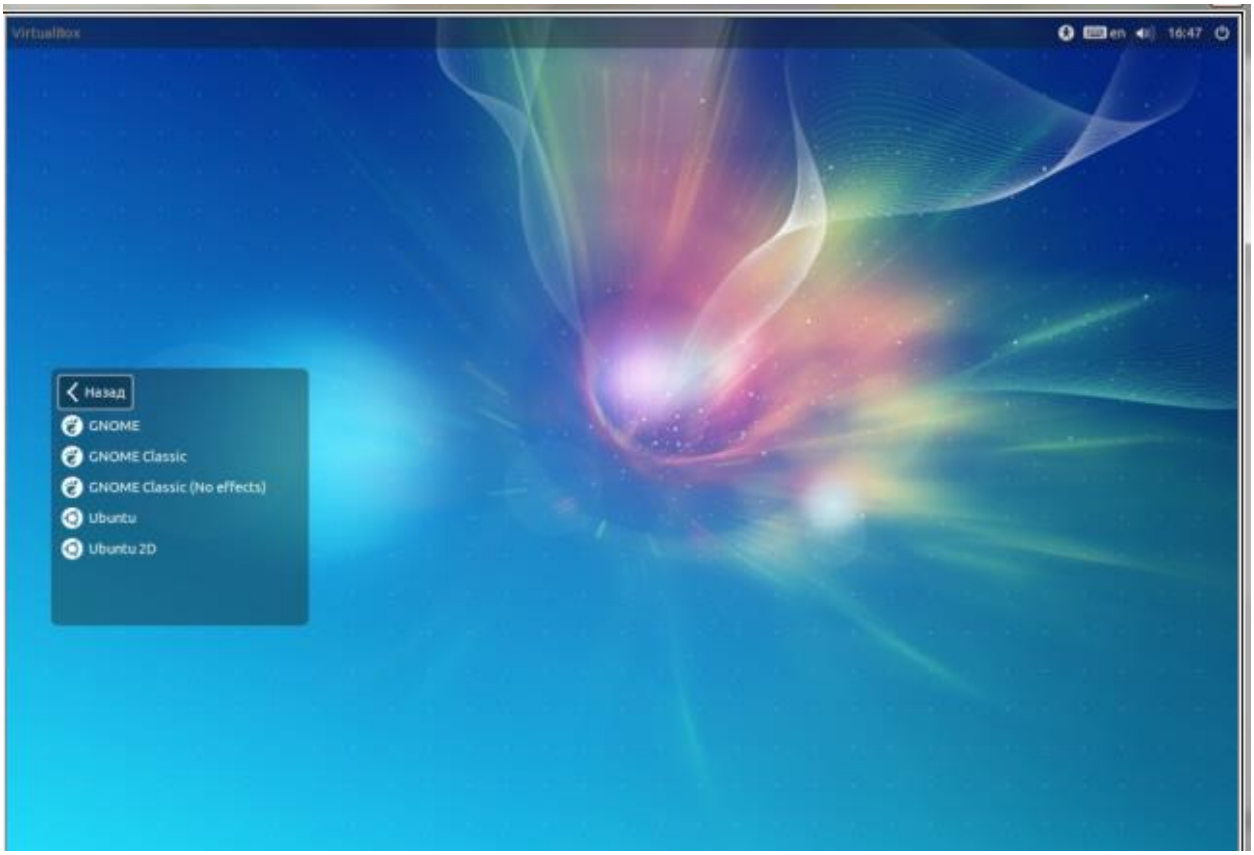


Дождаться установки операционной системы и перезагрузить виртуальную машины для входа в установленную операционную систему.

3)Установить дополнения гостевой операционной системы:

выбрав в меню VirtualBox «Устройства» - «Установить дополнения гостевой ОС». Дождаться установки и перезагрузить операционную систему.

При загрузке определить тип загружаемого сеанса (по умолчанию Ubuntu или Ubuntu 2D)



Выводы:

Получен опыт в создании виртуальных машин в **Oracle VM VirtualBox** знакомство с операционной системой **Ubuntu.**

2. Лабораторная работа № 1 Диспетчер задач Windows

Цель занятия: изучение диспетчера задач, его функций и возможностей.

2.1.Задание:

1. Изучите теорию, представленную в методичке.
2. Выполните практические задания и ответьте на вопросы.
3. Получите у преподавателя вариант задания.
4. Выполните контрольные задания в соответствии с полученным вариантом.

2.2. Краткие теоретические сведения

Утилита Диспетчер задач

Диспетчер задач в операционных системах семейства Microsoft Windows – утилита для вывода на экран списка запущенных процессов и потребляемых ими ресурсов (в частности статус, процессорное время и потребляемая оперативная память). Также есть возможность некоторой манипуляции процессами. Другими словами **Диспетчер задач** – это такая специальная программа, которая показывает нам много разной информации о том, что происходит за компьютером. Что именно показывает диспетчер?

Во-первых он показывает какие программы запущенны на компьютере в данный момент.

Во-вторых он показывает какие процессы сейчас протекают на компьютере.

В-третьих – сколько системных ресурсов занимает каждый запущенный процесс и какие ресурсы компьютера процесс занимает.

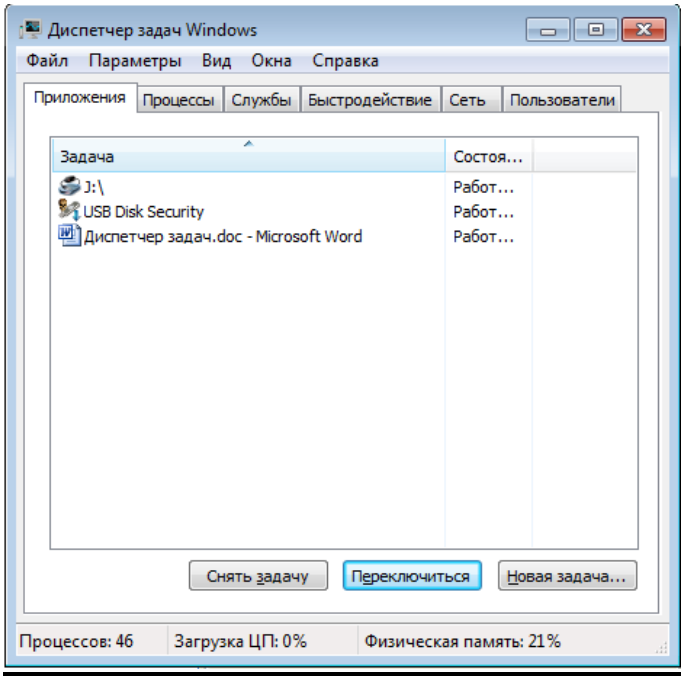


Рисунок 1 – Окно программы Диспетчер задач Windows

Способы запуска диспетчера задач

Для запуска диспетчера задач выполните любое из следующих действий:

1. Щелкните правой кнопкой мыши по пустой области на панели задач и выберите команду Запустить диспетчер задач.

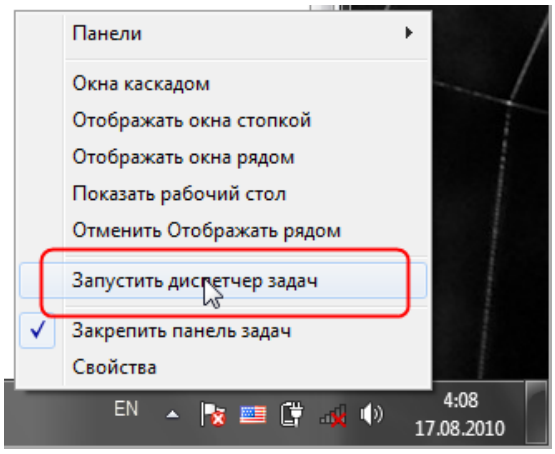


Рисунок 2 – Окно запуска диспетчера задач

2. Нажмите клавиши Ctrl+Shift+Esc.

3. Нажмите клавиши Ctrl+Alt+Delete и выберите опцию запустить Диспетчер задач.

После нажатия этих кнопок система выдаст вам довольно обширное меню команд:

– Блокировка компьютера;

– Смена пользователя;

– Завершение сеанса;

– Смена пароля;

– Ссылка же на Диспетчер Задач находится в самом низу этого списка.

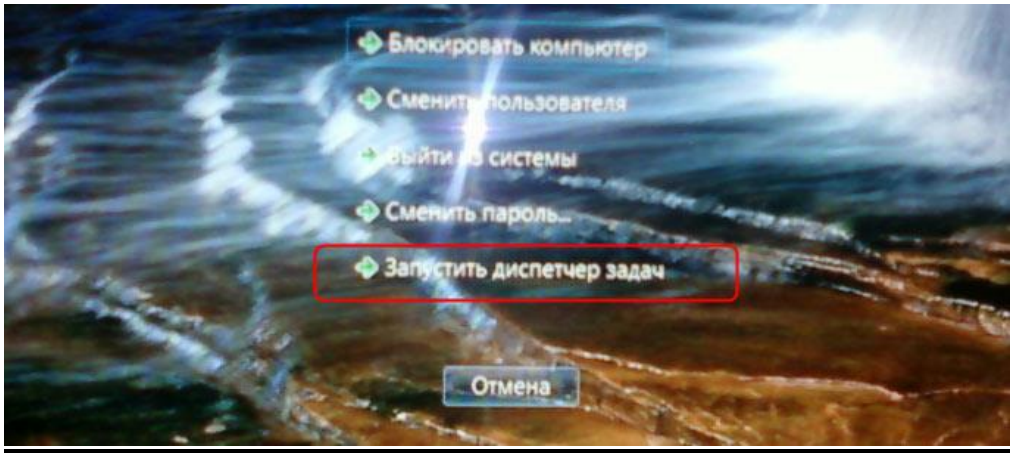


Рисунок 3 – Меню команд

4. Нажмите кнопку Пуск, введите в поле Найти слово taskmgr и нажмите клавишу Enter.

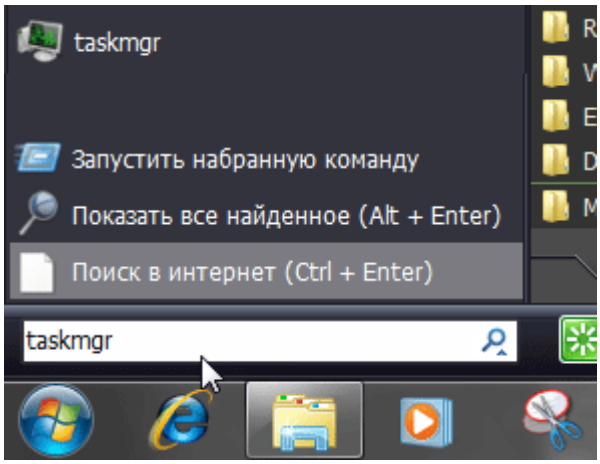


Рисунок 4 – Поле «Найти»

Окно Диспетчер задач Windows под основным меню (**Файл, Параметры, Вид, Окна, Завершение работы, Справка**) содержит 6 вкладок, каждая из которых представляет полезную информацию:

- Вкладка *Приложения*. Здесь можно найти список запущенных приложений и их состояние.
- Вкладка *Процессы*. Перечислены все программы и процессы, запущенные в системе, – это основное окно для прекращения работы «зависших» программ или процессов.
- Вкладка *Службы*. Содержит список программ, которые работают в фоновом режиме.
- Вкладка *Быстродействие*. Основная вкладка для оценки производительности операционной системы.
- Вкладка *Сеть*. Отображает объём передаваемых по локальной сети данных.
- Вкладка *Пользователи*. С помощью этой вкладки можно увидеть список всех пользователей, подключенных к вашему компьютеру по локальной сети. В противном случае будет указана только одна учётная запись пользователя.

Вкладка «Приложения»

Переходим на неё, нажимая левой кнопкой мыши. Перед нами список запущенных приложений и два столбца: столбец «Задача» и столбец «Состояние».

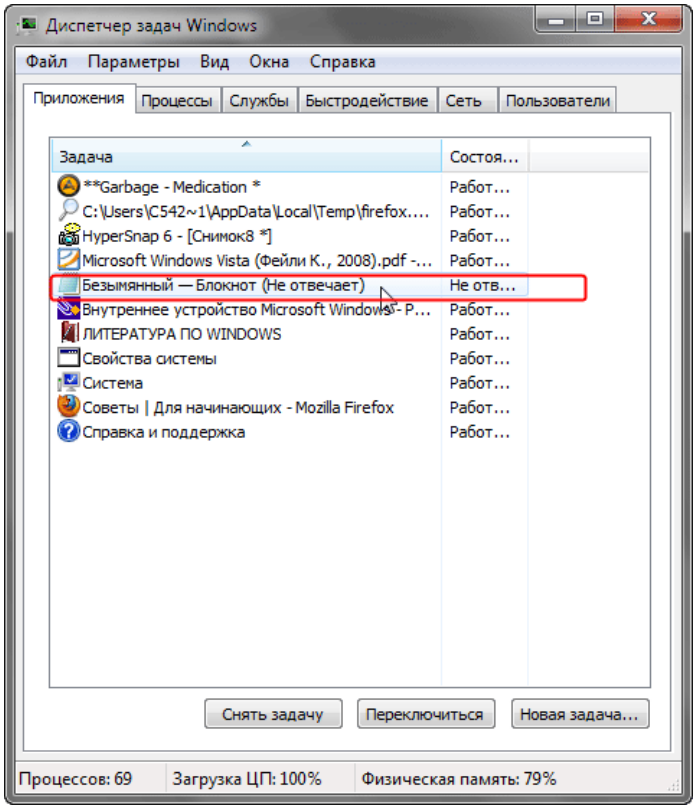


Рисунок 5 – Вкладка «Приложения»

Ищем приложение с состоянием «Не отвечает», либо «Не работает», зависит от Windows. Выделяем приложение с таким состоянием, щёлкая левой кнопкой мыши по названию в столбце «Задача».

Нажимаем на кнопку «Снять задачу». Если из списка программа сразу исчезла, то это означает, что она завершилась.

В основном зависшая программа работает некорректно и её придётся завершить вынужденно. В этом случае появится окошко с вопросом: «Завершить сейчас?», нажимаем на кнопку «Завершить сейчас».

Все несохранённые изменения, выполненные в этом приложении, будут потеряны. Если несохранённые данные важны для пользователя, есть смысл дождаться отклика программы.

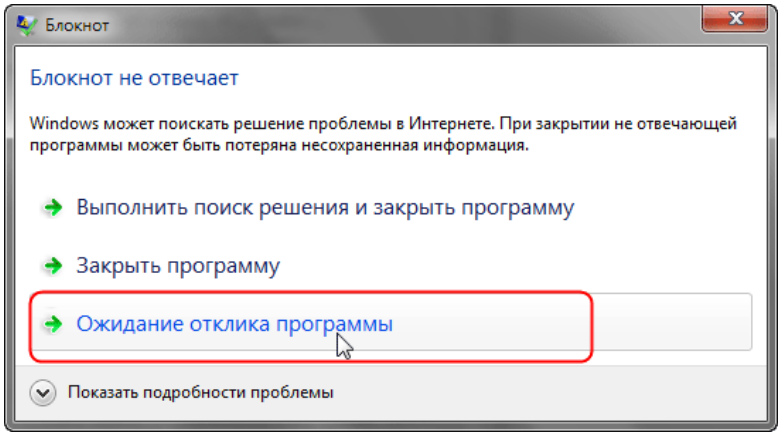


Рисунок 6 – Ожидание отклика программы

Программы, которые зависли/дали сбой, в среде Windows называются «неотвечающими»; пользователь может поместить курсор мыши в окно программы, но она не будет реагировать на щелчки мыши или нажатия клавиш. Если программа не отвечает, это не значит, что пользователь должен перезагружать компьютер. Воспользуйтесь Диспетчером задач и закройте зависшую программу.Перед тем как закрывать программу, убедитесь в том, что она действительно не отвечает. Подождите некоторое время; возможно,операционная система Windows пытается выделить для программы дополнительные ресурсы памяти. Например, если пользователь запускаете макрос Visual Basic в программе Microsoft Excel или Word, ему может показаться, что программа «зависла». Сложное изменение форматирования или выполнение операций по поиску и замене в объёмном документе могут создать впечатление того, что текстовый редактор «не отвечает». Открытое диалоговое окно или окно сообщения могут не позволить пользователю выполнять никакие действия в определенной программе; поищите их под текущим окном.

Если какая либо программа часто «зависает», то можно попробовать найти решение этой проблемы в интернете, создав отчет об ошибке.

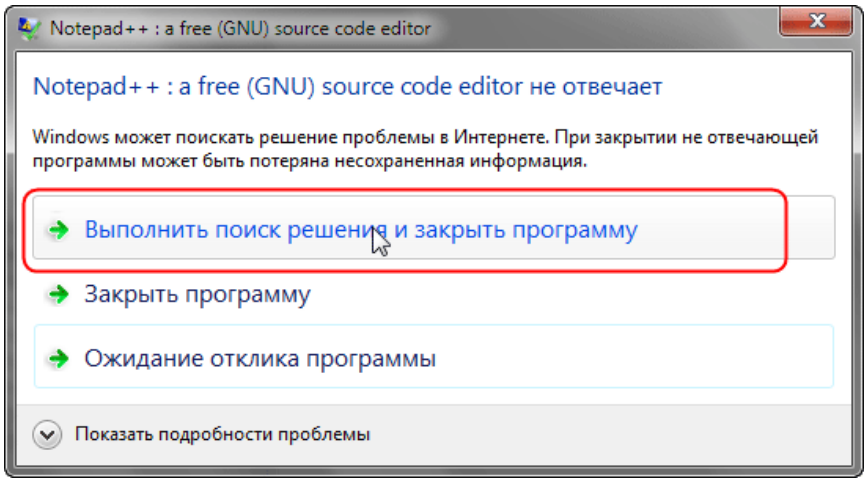


Рисунок 7 – Выполнить поиск решения и закрыть программу

Отчеты об ошибках Windows можно использовать, чтобы сообщать в корпорацию Майкрософт о проблемах на компьютере. Корпорация Майкрософт использует отчеты о проблемах для поиска решений, соответствующих описаниям проблем. В ОС Windows отображается уведомление при наличии возможных решений проблемы, а также о возможности поиска дополнительных сведений в Центре поддержки. Если же решения нет, то сведения, отсылаемые в сообщении о проблеме, могут помочь Microsoft найти или создать новое решение.

Если решение проблемы, о которой сообщил компьютер, существует, оно появится в Центре поддержки.

Кнопка Переключиться – откроет выделенную программу.

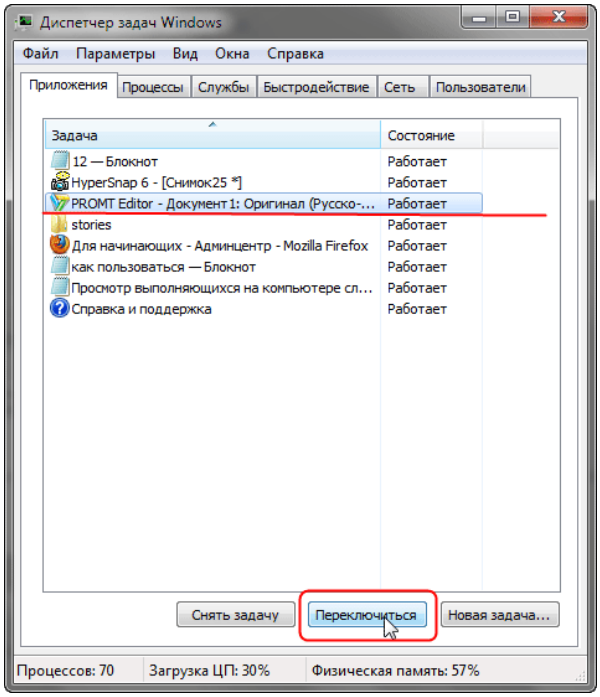


Рисунок 8 – Кнопка переключиться

Кнопка Новая задача – откроет диалоговое окно Выполнить. С помощью команды Выполнить можно быстро запускать программы, открывать файлы и папки, а также переходить на веб-сайты, если компьютер подключен к Интернету.

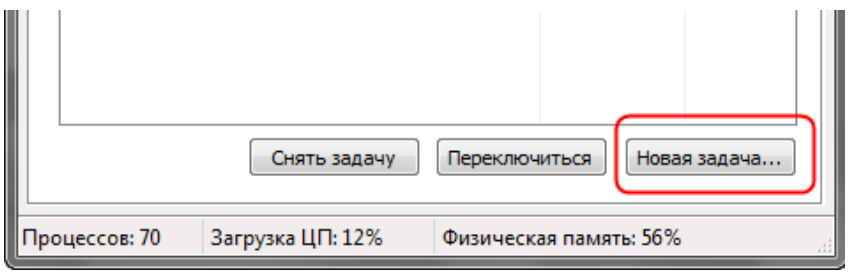


Рисунок 9 – Новая задача

С каждой программой, выполняемой на компьютере, связан определенный процесс, который запускает эту программу. Если программа перестает отвечать (или «зависает»), определение связанного с этой программой процесса может помочь в устранении возникших неполадок. Например, если известно, какой процесс используется для запуска данной программы, то чтобы закрыть зависшую программу, необходимо завершить этот процесс.

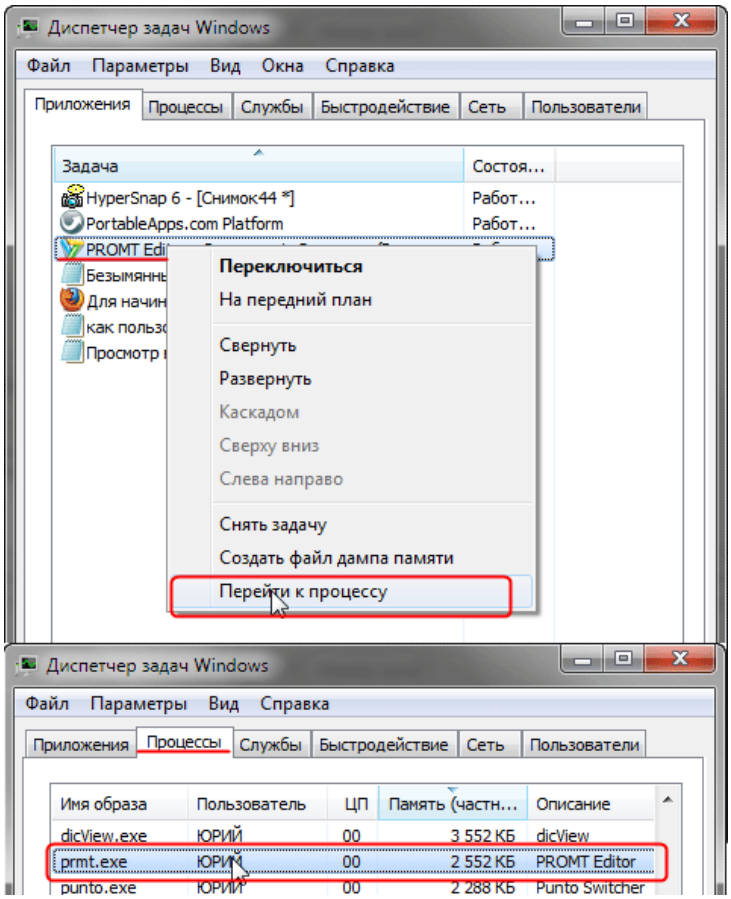


Рисунок 10 – Перейти к процессу

Чтобы определить, какой процесс используется программой, щелкните правой кнопкой мыши нужную программу и выберите команду Перейти к процессу. Связанный с данной программой процесс будет выделен на вкладке Процессы.

Вкладка «Процессы»

Для просмотра сведений о процессах, выполняющихся в данный момент на компьютере, можно использовать **Диспетчер задач.**

Процесс – это файл, например исполняемый файл, имя которого заканчивается расширением EXE. Этот файл используется компьютером для непосредственного запуска программ и других служб в специально выделенной для него области

оперативной памяти. Каждое запущенное приложение имеет соответствующий ему процесс.

Перейдите на вкладку **Процессы**. В диспетчере задач отображаются процессы, выполняющиеся в данный момент под текущей учётной записью пользователя.

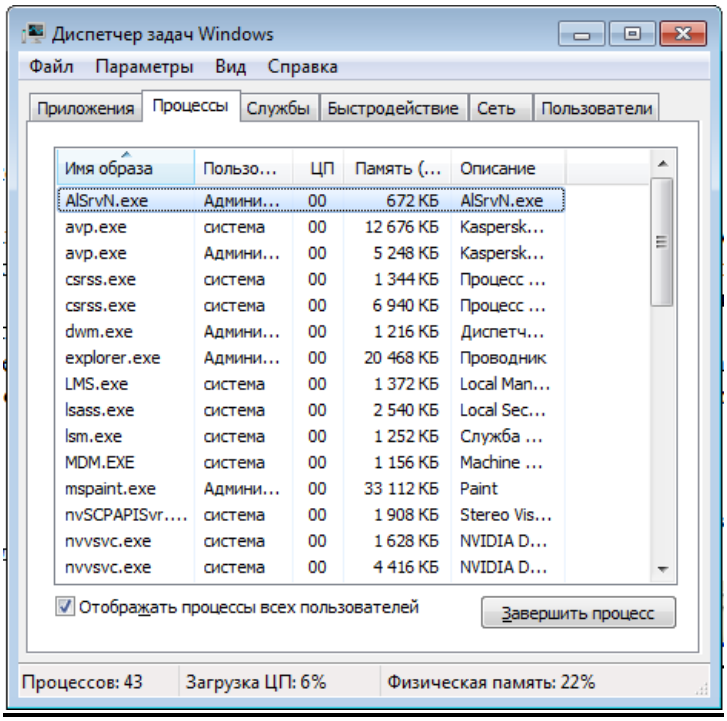


Рисунок 11 – Вкладка «Процессы»

Сколько процессов выполняется на данный момент времени, можно увидеть в нижней части диспетчера.

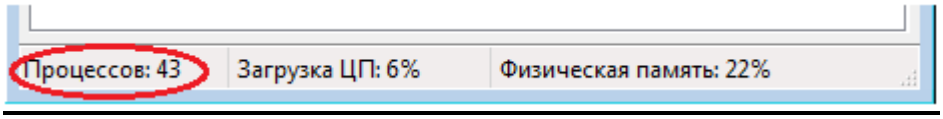


Рисунок 12 – Наличие процессов

В действительности процессов может быть намного больше. Где остальные процессы и почему они скрыты? Всё дело в том, что по умолчанию отображаются лишь процессы, на которые может влиять пользователь. Стоит лишь установить флажок *Отображать процессы всех пользователей*, как в основном окне появятся все процессы, поскольку будут добавлены процессы с атрибутами SYSTEM (системные процессы), NETWORK (сетевые процессы) и LOCAL SERVICE (локальные службы). Останавливать выполнение этих процессов не рекомендуется, поэтому не устанавливайте данный флажок без необходимости.

По умолчанию для каждого процесса отображается следующая информация:

- *Имя образа* – название процесса;

- *Пользователь* – имя пользователя, запустившего процесс;
- *ЦП* – процент мощности процессора, используемый процессом;
- *Память (частный рабочий набор)* – объём памяти, используемый процессом в ходе работы.

Распространённые процессы:

svchost.exe – это главный системный процесс для тех служб, которые запускаются из динамически загружаемых библиотек (DLL-файлов). И действительно несколько экземпляров процесса svchost.exe могут быть запущены одновременно (но с разными PID* (PID ─ это идентификатор пакета.)). Так как каждый из таких экземпляров представляет собой определённую преимущественно системную службу или же группу служб. Эти группы определены в следующем разделе реестра:

HKEY_LOCAL_MACHINESoftwareMicrosoftWindowsNTCurrentVersionSvcHost

Каждое значение в этом разделе представляет отдельную группу Svchost и отображается при просмотре активных процессов как отдельный экземпляр. Каждое из этих значений имеет тип REG_MULTI_SZ и содержит службы, выполняемые в этой группе Svchost. Каждая группа Svchost может содержать одно или несколько имен служб, извлекаемых из следующего раздела реестра, в котором подраздел Parameters содержит значение ServiceDLL:

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Служба

Чтобы просмотреть список служб, работающих в процессе Svchost, выполните описанные ниже действия.

1. Нажмите на панели задач Windows кнопку Пуск и выберите пункт Все программы, затем Стандартные, затем Монитор.

2. Введите команду Tasklist /SVC и нажмите клавишу ENTER.

Команда Tasklist выводит список активных процессов

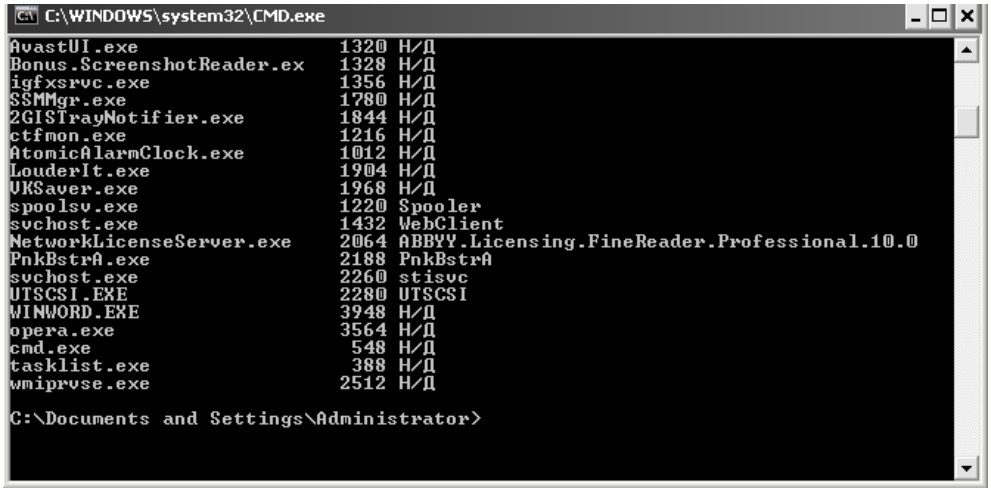


Рисунок 13 – Список активных процессов

Параметр /SVC используется для вывода списка активных служб в каждом процессе. Для получения дополнительных сведений о процессе введите следующую команду и нажмите клавишу ENTER:

Tasklist /FI «PID eq идентификатор_процесса» (кавычки обязательны).

DLL (англ. Dynamic–link library – динамически подключаемая библиотека) – понятие операционных систем Microsoft Windows и IBM OS/2; динамическая библиотека, позволяющая многократное применение различными программными приложениями. К DLL относятся также элементы управления ActiveX и драйверы. В мире UNIX аналогичные функции выполняют т. н. shared objects («разделяемые объекты»).

Цели введения DLL

Первоначально предполагалось, что введение DLL позволит эффективно организовать память и дисковое пространство, используя только один экземпляр библиотечного модуля для различных приложений. Это было особенно важно для ранних версий Microsoft Windows с жёсткими ограничениями по памяти. Далее, предполагалось улучшить эффективность разработок и использования системных средств за счёт модульности. Замена DLL–программ с одной версии на другую должна была позволить независимо наращивать систему, не затрагивая приложений. Кроме того, библиотеки DLL могли использоваться разнотипными приложениями — например, Microsoft Office, Microsoft Visual Studio и т. п.

В дальнейшем идея модульности выросла в концепцию COM (Component Object Model – объектная модель компонентов, компьютерная технология, разработанная компанией Microsoft.)

Фактически, полных преимуществ от внедрения DLL (Dynamic-link library – библиотека динамической компоновки) получить не удалось по причине явления, называемого DLL hell («ад DLL»). DLL Hell возникает, когда несколько приложений

требуют одновременно различные, не полностью совместимые, версии DLL–библиотек, что приводит к сбоям в этих приложениях. Когда система выросла до определённых размеров, количество DLL стало превышать многие тысячи килобайт не все из них обладали полной надёжностью и совместимостью, и конфликты типа DLL Hell стали возникать очень часто, резко понижая общую надёжность системы. Поздние версии Microsoft Windows стали разрешать параллельное использование разных версий DLL, что уничтожало преимущества изначального принципа модульности.

Службы Windows (англ. Windows Service, сервисы) – приложения, автоматически запускаемые системой при запуске Windows и выполняющиеся вне зависимости от статуса пользователя. Имеет общие черты с концепцией. В процессе загрузки на основании записей в реестре Svchost.exe составляет список служб, которые необходимо запустить. Одновременно могут быть запущены несколько экземпляров процесса Svchost.exe. Каждый сеанс Svchost.exe содержит группу служб, следовательно, отдельные службы могут выполняться в зависимости от того, как и когда был запущен Svchost.exe. Таким образом улучшается контроль и упрощается отладка. Файл Svchost.exe расположен в папке %SystemRoot%\System32. В других каталогах под именем Svchost.exe может скрываться вирусы (Троянская программа, вирус или сетевой червь). Наиболее известные злонамеренные программы, скрываются под именем системного процесса Svchost.exe – W32.Welchia.Worm, W32/Jeefo и W32.Assarm@mm. В этом случае злонамеренный процесс должен быть немедленно завершён.

csrss.exe – процесс управляет отображением окон в Windows. Если пользователь откроет новое окно, доступ к нему обеспечит csrss.exe. Одновременно csrss.exe отвечает и за управление другими процессами. Если пользователь обнаружил более двух записей csrss.exe в списке Диспетчера задач и эти файлы сильно загружают процессор, это означает что за этим скрывается вредоносная программа. В этом случае выполните проверку вашего компьютера с помощью программ Антивирус Касперского или Dr.Web.

smss.exe – процесс отвечает за запуск сеансов пользователей. Данный процесс запускается системным потоком и отвечает за различные действия, в частности, за запуск процессов winlogon и win32 (csrss.exe) и за установку системных переменных. После того как указанные процессы запущены, процесс **smss** ожидает завершения работы winlogon или csrss. Если это происходит «нормально», система может завершить свою работу; если же это происходит неожиданным образом, **smss.exe** вызывает зависание системы (система перестаёт реагировать на запросы).

lsass.exe – процесс отвечает за программы и настройки безопасности Windows и поэтому очень важен. lsass.exe очень часто является целью хакерских атак. Поэтому следует внимательно следить за этим процессом в Диспетчере задач. Если lsass.exe

заражён, не стоит самостоятельно пытаться завершить этот процесс или удалить файл. С этой задачей прекрасно справится Антивирус Касперского ,он в состоянии вылечить файлы. Многие вирусы используют для маскировки маленькую хитрость – в списке Диспетчера задач в их имени вместо прописной буквы «i» используется заглавная буква «I».

explorer.exe – процесс отвечает, в частности, за отображение Рабочего стола и содержания дисков. Очень часто можно встретить вирусы или «программы–шпионы», которые используют имя explorer.exe. Подобные вредители и «настоящий» файл, как правило, находятся в разных папках.

ctfmon.exe – процесс управляет технологиями альтернативного ввода данных. Он запускает языковую панель в системе при старте операционной системы, и работает в фоновом режиме даже после закрытия всех программ пакета Microsoft Office, независимо от того, запускались ли программы Office XP.

avp.exe – процесс, который зарегистрирован в качестве Kaspersky Anti–Virus

vmsrvc.exe – Virtual Machine Additions Services Application

vpcmap.exe – Virtual Machine Additions службы общего доступа к папкам. Работа в Microsoft Virtual PC или Virtual Server и установлена виртуальная машина дополнений, которые обеспечивают интеграцию виртуальный ПК / сервер с операционной системой. Эта услуга специально реализует Virtual Machine Additions функция, которая позволяет Virtual PC, / Server, чтобы использовать папки с хост-компьютера (локальные и сетевые папки) для совместного использования и обмена файлами между ПК хозяина и Virtual PC, / Server.

spoolsv.exe – процесс отвечает за обработку процессов печати на локальном компьютере в операционных системах Microsoft Windows. Служба spooler ответственна за управление заданиями на печать и передачу факсимильных сообщений.

taskmgr.exe – собственно менеджер задач.

services.exe – служба Plug & Play – Позволяет компьютеру распознавать изменения в установленном оборудовании и подстраиваться под них, либо не требуя вмешательства пользователя, либо сводя его к минимуму. Остановка или отключение этой службы может привести к нестабильной работе системы. Этот поцесс запускает первый svchost.

system – большинство системных потоков режима ядра исполняются от имени процесса system.

winlogon.exe – процесс управляет входом пользователей в систему и выходом из неё.Winlogon активируется только при нажатии клавиш CTRL+ALT+DEL, после чего появляется окно «Безопасность Windows».

wmiprvse.exe – инструментарий Windows, обеспечивает обмен управляющей информацией с устройствами.

Бездействие системы – процесс имеет по одному потоку на каждом процессоре и его единственная задача – учитывать время, в течение которого система не занята другими потоками. В диспетчере задач можно видеть, что этому процессу, как правило, соответствует большая часть процессорного времени.

Все процессы, кроме указанных выше, можно удалять, не опасаясь за работу системы. Однако некоторые из них возражают против такого действия (вирусы, вредоносные программы и некоторые антивирусы), в этом случае удаляем из автозапуска лишние программа с помощью AnVir Task Manager и перезапускаем систему.

Вкладка «Службы»

Диспетчер задач можно использовать для просмотра сведений о службах, выполняющихся в данный момент на компьютере Перейдите на вкладку Службы, чтобы увидеть службы, выполняющиеся в данный момент под текущей учетной записью пользователя.

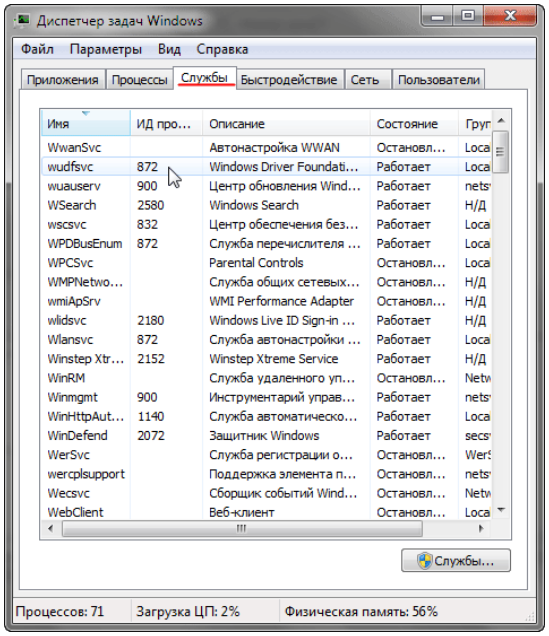


Рисунок 14 – Вкладка «Службы»

Можно также просматривать сведения о процессах, связанных с конкретной службой.

Чтобы просмотреть процесс, связанный со службой, щелкните правой кнопкой мыши требуемую службу и выберите команду Перейти к процессу. Если команда Перейти к процессу отображается затененной, значит выбранная служба в данный момент остановлена. Состояние службы (выполняется или остановлена) отображается в столбце Статус.

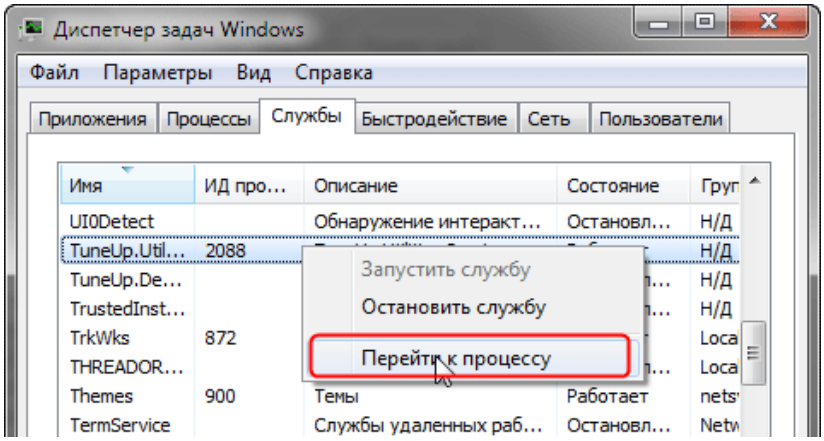


Рисунок 15 – Процесс связанный со службой

Если после выбора команды Перейти к процессу процесс не выделен на вкладке Процессы, значит он не выполняется под текущей учетной записью пользователя. Чтобы просмотреть все процессы, перейдите на вкладку Процессы и установите флажок Отображать процессы всех пользователей. При появлении запроса пароля администратора или подтверждения введите пароль или предоставьте подтверждение. Перейдите на вкладку Службы и повторите попытку просмотреть сведения о процессе.

При нажатии кнопки Службы в нижней части вкладки Службы откроется оснастка консоли управления (MMC), где опытные пользователи могут просматривать более подробные сведения о службах и настраивать дополнительные параметры.

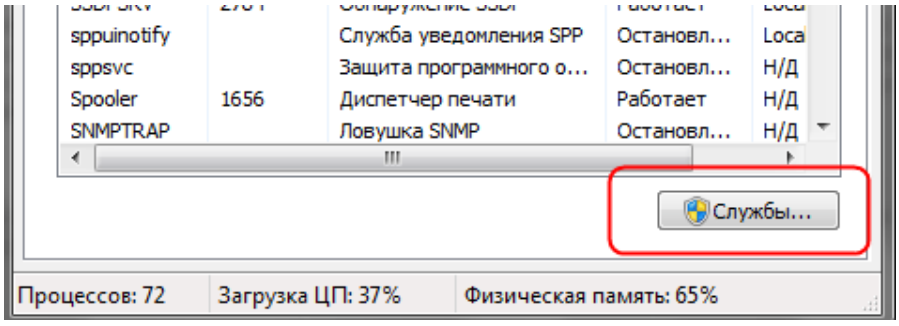


Рисунок 16 – Кнопка Службы

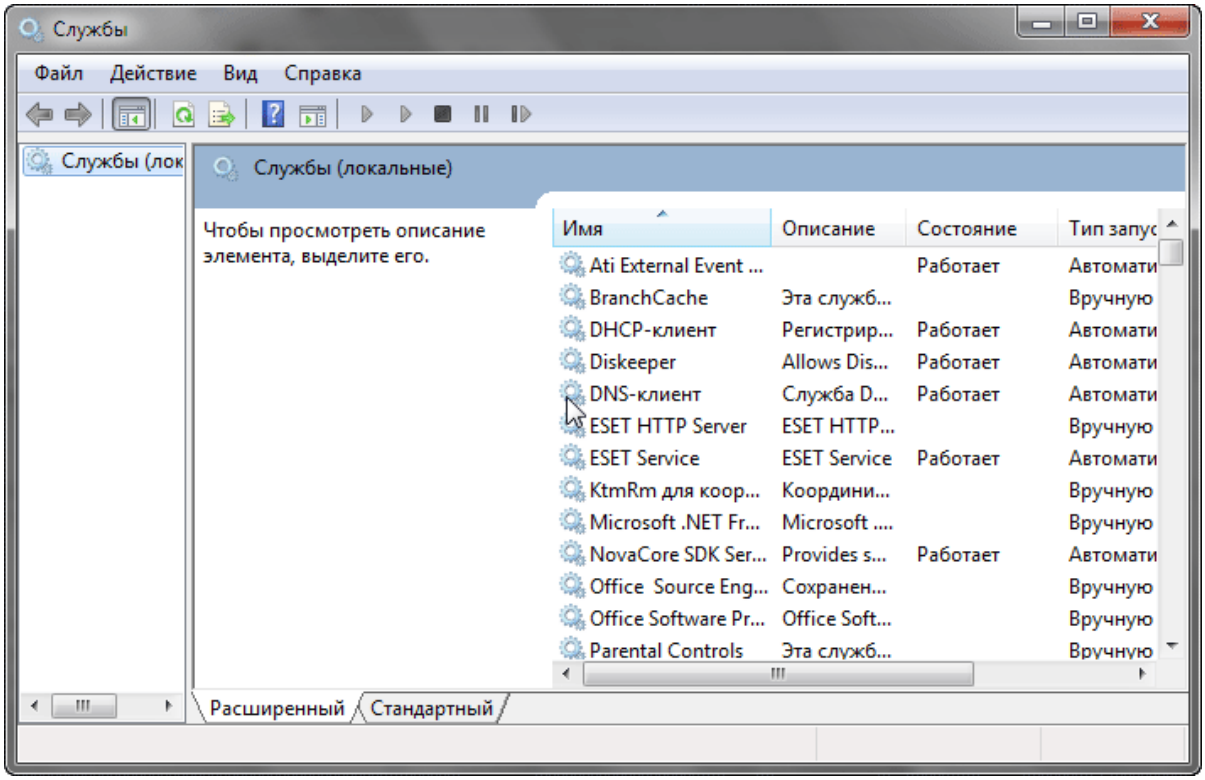


Рисунок 17 – Подробные сведения о службах

Вкладка «Быстродействие»

На вкладке Быстродействие диспетчера задач приведены дополнительные сведения об использовании компьютером системных ресурсов, например памяти (ОЗУ) и ресурсов центрального процессора (ЦП).

На этой вкладке отображается четыре графика. Два графика вверху показывают загруженность ЦП как в текущий момент, так и за несколько последних минут.

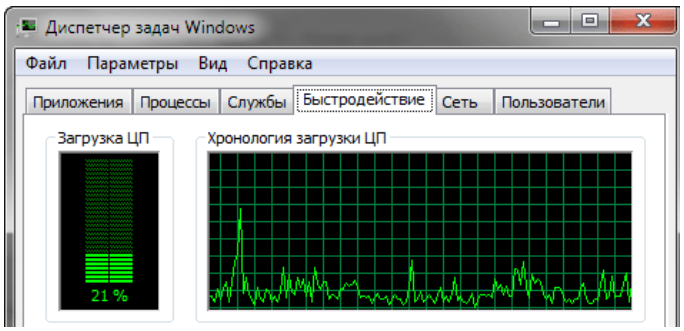


Рисунок 18 – Вкладка «Быстродействие»

Высокое процентное значение показывает, что программы или процессы используют большой объём ресурсов процессора, что может замедлить работу компьютера. Приложение может не отвечать, когда процентный показатель фиксируется или приближается к значению 100%.

Процентное значение загрузки процессора показано так же в нижней части диспетчера задач.

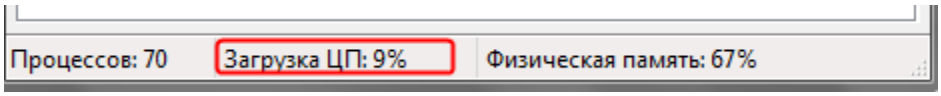


Рисунок 19 – Загрузка ЦП

Два графика внизу показывают объём используемого ОЗУ, или физической памяти, в мегабайтах (МБ) как в текущий момент, так и за несколько последних минут.

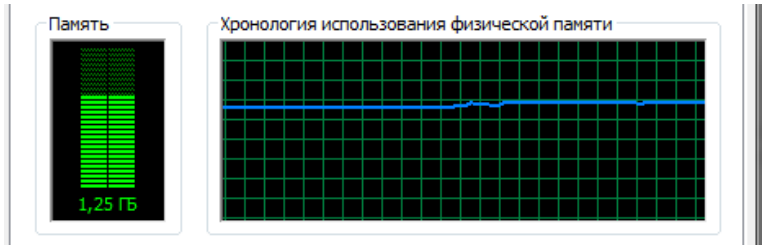


Рисунок 20 – Файл подкачки и хронология использования файла подачки

Самая частая причина замедления работы системы под управлением Windows – заполнение физической памяти. При этом Windows начинает так называемую «подкачку» (paging) – перемещение блоков кода и данных программ (каждый такой блок называется страницей – page) из физической памяти на жесткий диск. Обращение к файлу подкачки время от времени – нормальное явление, не ухудшающее производительность системы, но частые запросы данных из файла на диске могут заметно снизить общую скорость работы системы. Эта проблема становится особенно заметной при переключении между несколькими программами, активно использующими память, на компьютере, который не содержит достаточного количества физической памяти. В результате диск почти постоянно находится в работе, потому что система пытается «перекачать» данные с него в память и обратно.

Самый быстрый способ получить информацию об использовании памяти в данный момент – запустить Диспетчер задач Windows и взглянуть на строку состояния внизу любой вкладки. Статистика использования памяти приведена в правой части вкладки в виде двух чисел, а точнее дроби. Первое число (числитель) представляет собой текущий объём выделенной памяти – количество физической и виртуальной памяти, используемой всеми выполняемыми процессами. (*Виртуальная память, собственно, и есть файл подкачки*.) Знаменатель — общее количество доступной памяти (физической и виртуальной). Само по себе это число способно лишь предупредить вас о том, что память скоро закончится совсем, – другими словами, выделенная память примерно совпадает с доступной.

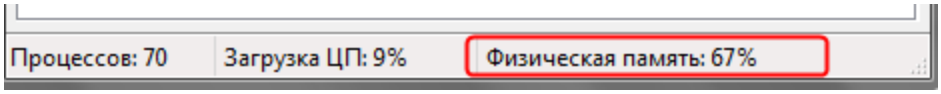


Рисунок 21 – Физическая память

Для того чтобы узнать об использовании памяти подробнее, переключитесь на вкладку Быстродействие и взгляните на таблицы в нижней части диалогового окна. Учтите: числа и подписи к ним могут означать абсолютно не то, что видно изначально.

В трех дополнительных таблицах под графиками содержатся сведения об использовании памяти и ресурсов.

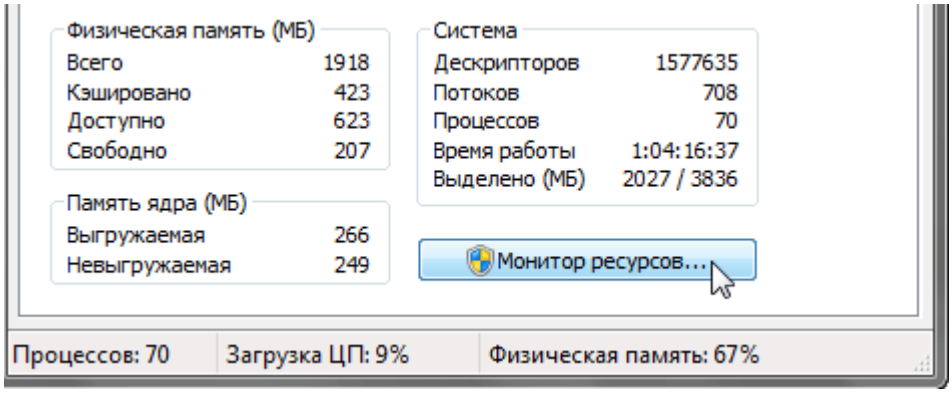


Рисунок 22 – Сведения об использовании памяти и ресурсов

Под заголовком Физическая память (МБ) значение:

- **Всего —** то объём в мегабайтах (Мбайт) оперативной памяти, установленной на компьютере.
- **Кэшировано —** это объём физической памяти, использованной за последнее время для системных ресурсов.
- **Доступно —** это объём памяти, непосредственно доступный для использования процессами, драйверами и операционной системой.
- **Свободно —** это то количество памяти, которое в данный момент не используется или не содержит полезной информации (в отличие от кэшированных файлов, которые содержат полезную информацию).

Под заголовком Память ядра (МБ):

- **Выгружаемая ─** это объём виртуальной памяти, используемый основной частью Windows, называемой ядром.
- **Невыгружаемая ─** это объём оперативной памяти, используемый ядром.

Под заголовком Система:

- **Дескрипторы**. Число уникальных идентификаторов объектов, используемых процессами. Это значение представляет интерес главным образом для ИТ-специалистов и программистов.
- **Потоки**. Число объектов или процессов, выполняющихся внутри более крупных процессов или программ. Это значение представляет интерес главным образом для ИТ─специалистов и программистов.
- **Процессы**. Число отдельных процессов, исполняемых на компьютере (эти сведения можно видеть и на вкладке «Процессы»).
- **Время работы**. Время, прошедшее после перезагрузки компьютера.
- **Выделено (МБ)**. Описание использования виртуальной памяти (также называемой файлом подкачки). Страничный файл ─ это место на жестком диске, используемое Windows в дополнение к ОЗУ. Первое число представляет собой объём используемой в данное время оперативной и виртуальной памяти, а второе ─ объём доступной оперативной и виртуальной памяти.

Для того чтобы узнать об использовании памяти подробнее, переключитесь на вкладку Быстродействие и взгляните на таблицы в нижней части диалогового окна. Учтите: числа и подписи к ним могут означать абсолютно не то, что видно изначально.

Ниже представлены некоторые замечания по поводу значений, которые помогут выполнять мониторинг параметров памяти.

- если значение *Физическая память* (КБ): *Доступно* приближается к нулю, это означает, что данной системе не хватает ресурсов памяти. Причина этого может заключаться в следующем: в системе одновременно работает довольно большое количество программ или одна большая программа использует практически все ресурсы памяти.
- если значение *Физическая память (КБ): Системный кэш* более чем в половину меньше значения *Физическая память (КБ): Всего*, это означает, что данная система функционирует не настолько эффективно, насколько могла бы, потому что Windows XP не удаётся сохранять достаточное количество недавно использовавшихся данных в памяти. Поскольку Windows XP не использует

системный кэш, когда ей требуется физическая память, закройте программы, которые не нужны.

- если значение *Выделение памяти (КБ): Всего* превышает значение *Физическая память (КБ): Всего* (и остаётся таковым), это означает, что Windows XP постоянно выгружает данные в страничный файл(page file, или paging file) и загружает их из него, а это существенно снижает степень производительности.

- если значение *Выделение памяти (КБ): Пик* выше значения *Физическая память (КБ): Всего*, это означает, что в какой–то момент во время текущего сеанса Windows XP пришлось прибегнуть к помощи страничного файла. Если на текущий момент времени значение *Выделение памяти (КБ): Всего* меньше значения *Физическая память (КБ): Всего*, критически высокое значение, скорее всего, было лишь временным событием, однако, чтобы быть абсолютно уверенным в этом, за данным значением лучше понаблюдать.

Чуть подробнее о «файле подкачки»:

Как уже говорилось ранее, Windows работает не одними только чипами ОЗУ. Помимо физической памяти для хранения программ и рабочих данных, Windows использует скрытый файл на жестком диске. При необходимости освободить ОЗУ данные перекачиваются в этот файл. Файл подкачки (page file) также называют свопом (swap file, paging file), причем в Windows XP в разных местах используются разные термины. Для единообразия мы будем повсеместно пользоваться только термином файл подкачки.

В процессе установки Windows XP файл подкачки автоматически создаётся в корневой папке на том же диске, где расположены системные файлы Windows. Размер файла подкачки определяется, исходя из объёма физической памяти в вашей системе. По умолчанию минимальный размер файла подкачки в 1,5 раза больше, чем объём физической памяти, а максимальный размер – в 3 раза больше. Файл подкачки можно увидеть в окне Проводника, если включить режим отображения скрытых и системных файлов. Файл pagefile.sys можно найти в корневой папке системного диска.

Открыв окно Панель управления Система, вы сможете наблюдать параметры файла подкачки и изменить их по своему желанию: изменить размер файла подкачки, перенести его на другой диск или разделить на несколько физических дисков для увеличения производительности.

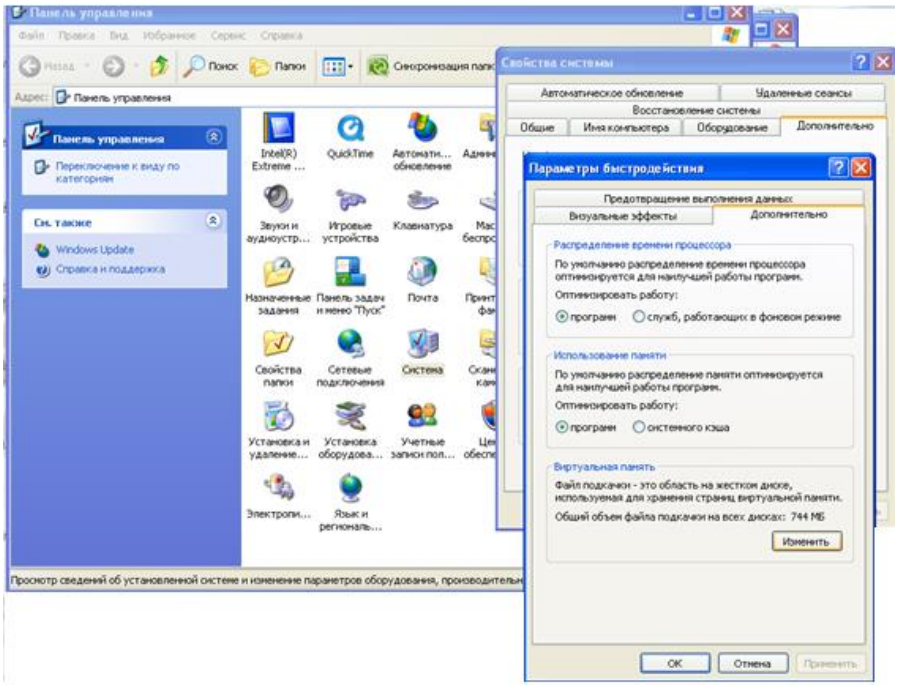


Рисунок 23 – изменение Файла подкачки

Кнопка Монитор ресурсов предназначена для просмотра дополнительных сведений об используемой памяти и ресурсах ЦПУ.Монитор ресурсов предоставляет графические сведения подобно тем, которые отображаются в диспетчере задач, но в более подробном виде. Здесь также приводятся дополнительные сведения о ресурсах, такие как использование дисков и сети.

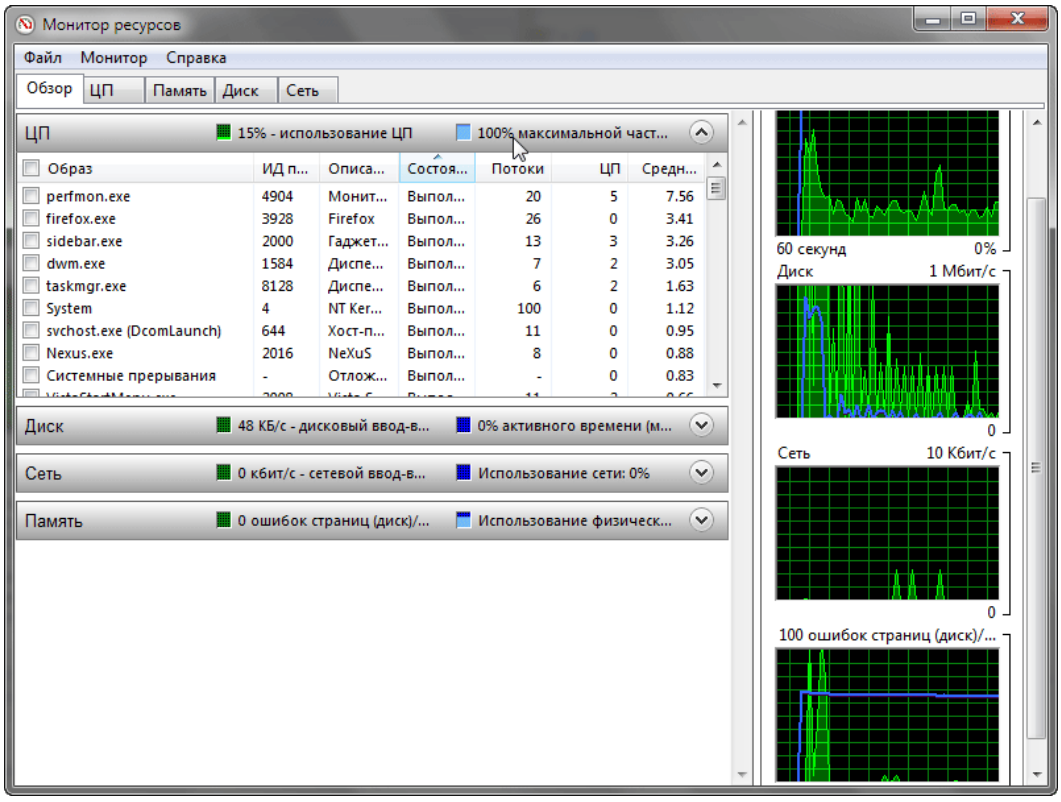


Рисунок 24 – Кнопка Монитор

Вкладка «Сеть»

При наличии сетевого оборудования на этой вкладке вы увидите список имеющихся сетевых подключений и графики их активности. С их помощью можно определить интенсивность использования сети и её пропускную способность.

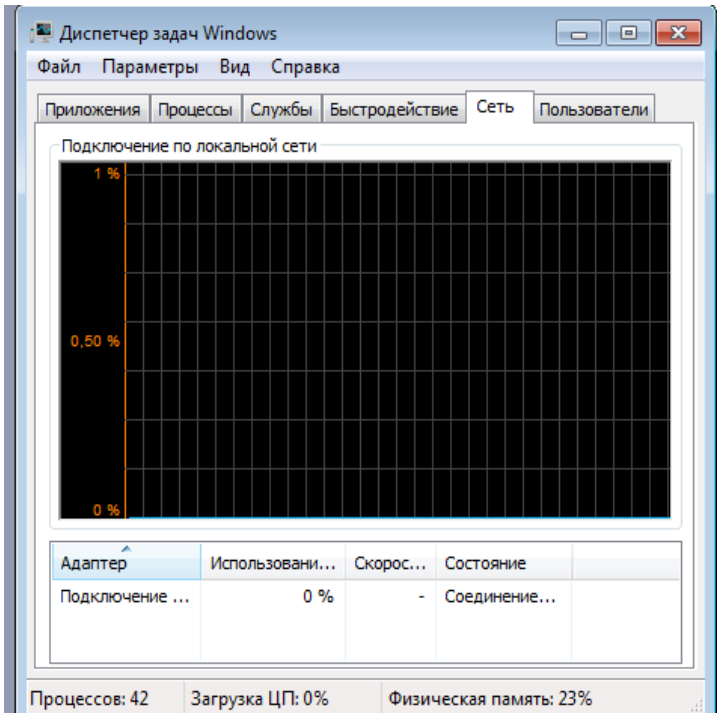


Рисунок 25 – Вкладка «Сеть»

В нижней части вкладки присутствует таблица с текущими параметрами сетевых подключений. Для детального анализа работы сетевого адаптера пользователь может использовать более двух десятков дополнительных сетевых параметров, которые можно отобразить в таблице, выполнив команду **Вид → Выбрать столбцы**.

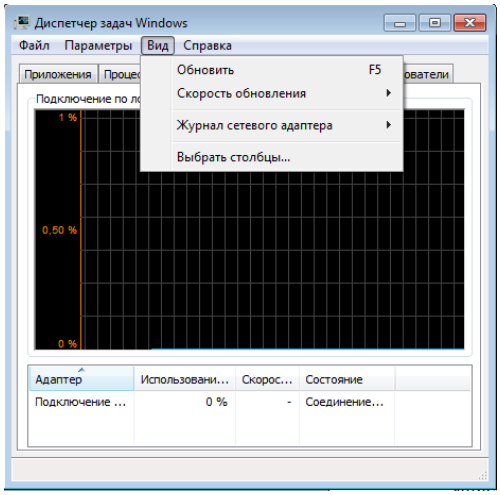


Рисунок 26 – Выбрать столбцы

А с помощью команды **Вид → Журнал сетевого адаптера** можно включать или выключать отображение дополнительных графиков.

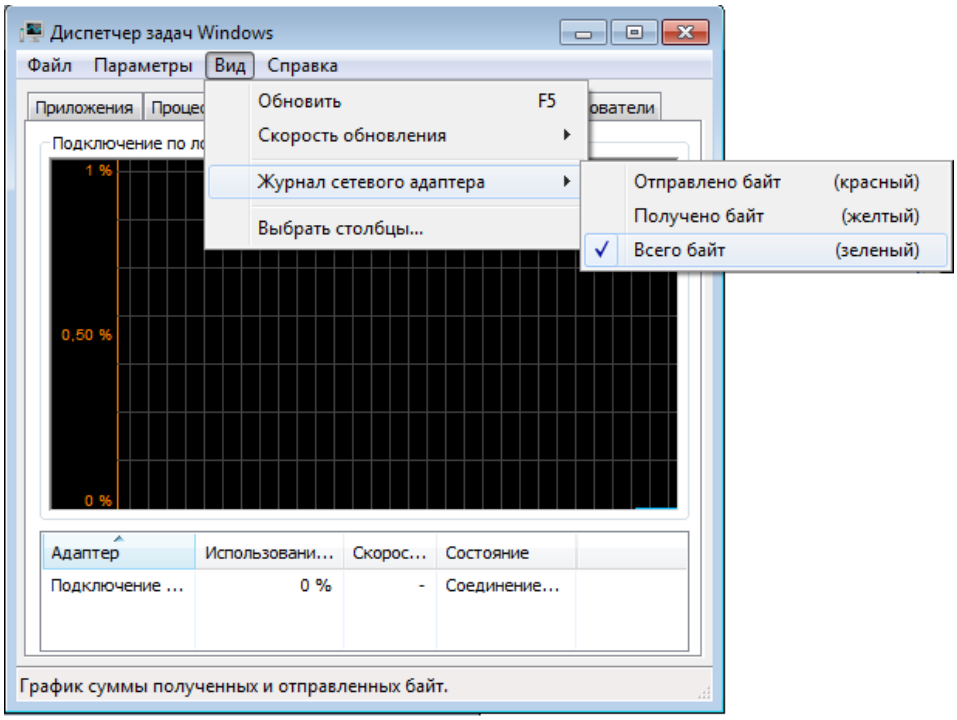


Рисунок 27 – Журнал сетевого адаптера

Существуют способы наложения запрета на автозагрузку программ через записи в реестре, указанные выше. Используются параметры типа DWORD. Все параметры должны храниться в разделе KLMSoftware\Microsoft\Windows\CurrentVersion\Policies\Explorer

Для запрета запуска программ, прописанных в подразделе Run раздела LOCAL MACHINE используется параметр DisableLocalMachineRun со значением 1. В этом случае система игнорирует содержимое списка Run, находящегося в LOCAL MACHINE. Аналогично действует запрет списка Run Once для LOCAL MACHINE. За состояние этой политики отвечает параметр DisableLocalMachineRunOnce. Система игнорирует содержимое RunOnce в LOCAL MACHINE.

Вкладка «Пользователи»

Данная вкладка позволяет просмотреть список активных пользователей и при необходимости выполнить для выбранного пользователя операции отключения или выхода из системы с помощью кнопок в нижней части окна.

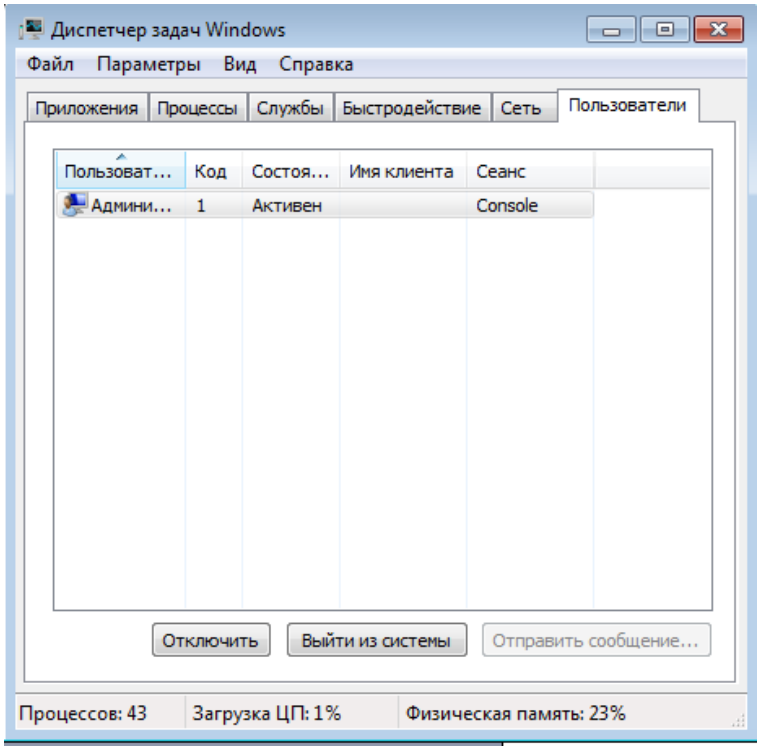


Рисунок 28 – Вкладка «Пользователь»

Закрыть окно Диспетчера задач Windows.

Контрольные вопросы:

1. Способы запуска Диспетчера задач?
2. Сколько экземпляров svchost.exe может быть запущенно на компьютере? Почему?
3. Что такое spoolsv.exe?
4. Чем отличается файл подкачки от виртуальной памяти? Где можно увидеть?
5. Дискиптор это-…
6. Что такое службы? Для чего предназначены?
7. Что необходимо сделать, чтобы запустить настройку системы?
8. Что обеспечивает обычный запуск операционной системы?
9. В каком случае рекомендуется использовать диагностический запуск?
10. Что такое system.ini? Для чего он нужен?
11. Что обеспечивает раздел boot?
12. Какие функции возложены на файл boot.ini?
13. Что такое Application Layer Gateway Service?
14. Что такое timeout, rdisk(0), disk(0), default?
15. Что делает Computer Browser?
16. Расскажите про Cryptographic Services?

3. Практическое занятие № 2 Файловая система NTFS.

Цель работы

Научиться устанавливать разрешения NTFS для файлов и для папок для отдельных пользователей и групп в операционной системы Windows 7, а также устранять проблемы доступа к ресурсам.

Оборудование: персональный компьютер (монитор, системный блок, клавиатура, мышь), ОС Windows, программа для виртуализации Oracle VM VirtualBox, образ виртуальной машины (Win7_01).

3.1. Теоретические сведения

Разрешения NTFS позволяют явно указать, какие пользователи и группы имеют доступ к файлам и папкам и какие операции с содержимым этих файлов или папок им разрешено выполнять. Разрешения NTFS применимы только к томам, отформатированным с использованием файловой системы NTFS. Они не предусмотрены для томов, использующих файловые системы FAT или FAT32. Система безопасности NTFS эффективна независимо от того, обращается ли пользователь к файлу или папке, размещенными на локальном компьютере или в сети.

Разрешения, устанавливаемые для папок, отличаются от разрешений, устанавливаемых для файлов. Администраторы, владельцы файлов или папок и пользователи с разрешением «Полный доступ» имеют право назначать разрешения NTFS пользователям и группам для управления доступом к этим файлам и папкам.

Список управления доступом

В NTFS хранится ***список управления доступом*** (access control list —**ACL)** для каждого файла и папки на томе NTFS. В этом списке перечислены пользователи и группы, для которых установлены разрешения для файла или папки, а также сами назначенные разрешения. Чтобы пользователь получил доступ к ресурсу, в ACL должна быть запись, называемая ***элемент списка управления доступом*** (access control entry — ACE) для этого пользователя или группы, к которой он принадлежит. Эта запись назначит запрашиваемый

тип доступа (например, **Чтение)** пользователю. Если в ACL нет соответствующей ACE, то пользователь не получит доступ к ресурсу.

Множественные разрешения NTFS

Вы можете установить несколько разрешений пользователю и всем группам, членом которых он является. Для этого вы должны иметь представление о правилах и

приоритетах, по которым в NTFS назначаются и объединяются множественные разрешения и о наследовании разрешений NTFS.

Эффективные разрешения. Эффективные разрешения пользователя для ресурса — это совокупность разрешений NTFS, которые вы назначаете отдельному пользователю и всем группам, к которым он принадлежит. Если у пользователя есть разрешение «Чтение» для папки, и он входит в группу, у которой есть разрешение «Запись» для той же папки, значит, у этого пользователя есть оба разрешения.

Установка разрешений NTFS и особых разрешений

Вы должны руководствоваться определенными принципами при установке разрешений NTFS. Устанавливайте разрешения согласно потребностям групп и пользователей, что включает в себя разрешение или предотвращение наследования разрешений родительской папки подпапками и файлами, содержащимися в родительской папке.

Если вы уделите немного времени на планирование ваших разрешений NTFS и будете соблюдать при планировании несколько принципов, то обнаружите, что разрешениями легко управлять.

- Для упрощения процесса администрирования сгруппируйте файлы по папкам следующих типов: папки с приложениями, папки с данными, личные папки. Централизуйте общедоступные и личные папки на отдельном томе, не содержащем файлов операционной системы и других приложений. Действуя таким образом, вы получите следующие преимущества:

- — сможете устанавливать разрешения только папкам, а не отдельным файлам;
- — упростите процесс резервного копирования, так как вам не придется делать резервные копии файлов приложений, а все общедоступные и личные папки находятся в одном месте.

- Устанавливайте для пользователей только необходимый уровень доступа. Если необходимо чтение файла, установите пользователю разрешение Чтение для этогофайла. Это уменьшит вероятность случайного изменения файла или удаления важных документов и файлов приложений пользователем.

- Создавайте группы согласно необходимому членам группы типу доступа, затем установите соответствующие разрешения для группы. Назначайте разрешения отдельным пользователям только в тех случаях, когда это необходимо.

- При установке разрешений для работы с данными или файлами приложений установите разрешение Чтение и выполнение для групп **Пользователи** и

Администраторы. Это предотвратит случайное удаление файлов приложений или их повреждение вирусами или пользователями.

- При установке разрешений для папок с общими данными назначьте разрешения Чтение и выполнение и Запись группе Пользователи и разрешение Полный доступ для группы Создатель-владелец. По умолчанию пользователь, создавший документ, также является его владельцем. Владелец файла может дать другому пользователю разрешение на владение файлом. Пользователь, который принимает такие права, в этом случае стано-вится владельцем файла. Если вы установите разрешение Чтение и выполнение и Запись группе Пользователи и разрешение Полный доступ группе Создатель-владелец, то пользователи получат возможность читать и изменять документы, созданные другими пользователями, а также читать, изменять и удалять файлы и папки, создаваемые ими.
- Запрещайте разрешения, только если необходимо запретить отдельный тип доступа определенному пользователю или группе.
- Поощряйте пользователей в установке разрешений для файлов и папок, которые они создают, и научите их это делать самостоятельно.

Администраторы, пользователи с разрешением Полный доступ и владельцы файлов и папок могут устанавливать разрешения для отдельных пользователей и групп.

Дополнительно Позволяет получить доступ к дополнительным возможностям поиска, включая возможность поиска удаленных учетных записей пользователей, учетных записей с не устаревшими паролями и учетных записей, по которым не подключались определенное количество дней.

Назначение или запрещение особых разрешений

Щелкните кнопку **Дополнительно,** чтобы открыть диалоговое окно **Дополнительные параметры безопасности,** где перечислены группы и пользователи и установленные для них разрешения для этого объекта. В поле **Элементы разрешений** также указано, от какого объекта разрешения унаследованы и к каким объектам применимы. Вы можете воспользоваться диалоговым окном **Дополнительные параметры безопасности** для изменения разрешений, установленных для пользователя или группы. Для изменения разрешений, установленных для пользователя или группы, выделите пользователя ищелкните кнопку **Изменить.** Откроется диалоговое окно **Элемент разрешения для.** Затем выделите или отмените определенные разрешения, которые вы хотите изменить.

3.2.Задание для самостоятельной работы

Задание 1. Открыть Oracle VM VirtualBox

Задание 2. Загрузить виртуальную машину Windows 7 (Win7_01) и создать новую учетную запись uir.

Задание 3. Загрузить виртуальную машину Windows 7 с учетной записью uir.

Задание 4. Определение разрешений NTFS по умолчанию для только что созданной папки.

Запустить **Проводник,** создать папки **C:\FoIderl** и **C:\FoIderl\Folder2.** Просмотреть разрешения, установленные для созданных папок, щелкнув по вкладке **Безопасность** диалогового окна свойств папки. Обратить внимание на наследование разрешений папкой **Folder2** от родительской папки **Folder 1.**

Если на экране не видна вкладка **Безопасность,** вам следует уточнить два вопроса:

1) Раздел вашего диска отформатирован как NTFS или как FAT? Только на разделах NTFS используются разрешения NTFS, и, таким образом, только на разделах NTFS видна вкладка

Безопасность.

2) Используете вы простой общий доступ к файлам или нет? Щелкните кнопку **Отмена,** чтобы закрыть диалоговое окно свойств папки. **В** пункте меню **Сервис** выберите пункт **Свойства папки.** В диалоговом окне **Свойства папки** перейдите на вкладку **Вид.** В списке **Дополнительные параметры** снимите флажок **Использовать простой общий доступ к файлам (рекомендуется)** и щелкните **ОК.**

Определить для какой группы установлены особые разрешения. Щелкнуть кнопку **Дополнительно,** выделить эту группу и просмотреть установленные разрешения.

Закрыть диалоговое окно свойств папки. Закрыть окно **Проводник** и завершить сеанс.

Задание 5. Создать новую учетную запись **uir-1**.

Задание 6. Войти в систему с учетной записью **uir-1**. Запустить **Проводник,** войти в папку **C:\FoIderl**. Создать два текстовых документа, присвоив им имена file **1** и file **2** Попытаться выполнить следующие операции с **файлом filel**: открыть файл; изменить файл; удалить файл. Какие действия вы смогли успешно совершить и почему?

Завершить сеанс работы и войти в систему, используя учетную запись **uir-2.** Запустить **Проводник,** войти в папку **C:\Folderl.** Попытаться выполнить следующие операции с файлом **fiie2:** открыть файл; изменить файл; удалить файл. Какие действия вы смогли успешно совершить и почему? В настоящее время ваша регистрационная запись — **uir-2.** Можете ли вы изменить разрешения, установленные для пользователя, пока вы подключены как **uir-2?** Почему? Завершить сеанс.

Задание 7. Установить разрешения NTFS для папки **C:\Folderl.** При этом необходимо соблюдать следующие правила:

1) все пользователи должны иметь возможность читать документы и файлы в папке **Folderl;**

2) все пользователи должны иметь возможность создавать документы в папке **Folderl;**

3) все пользователи должны иметь возможность изменять содержание, свойства и разрешения для создаваемых ими документов в папке **Folderl;**

4) пользователь **uir-2** несет ответственность за содержимое папки **Folderl** и должен иметь возможность изменять и удалять все файлы в папке **Folderl.**

Основываясь на полученной информации, определить, как следует изменить разрешения для соответствия этим четырем критериям?

Войти в систему, используя учетную запись **uir.** Открыть Проводник. Открыть папку **Folderl.** Щелкнуть правой кнопкой мыши значок папки **Folderl,** затем выбрать пункт меню Свойства. Перейти на вкладку **Безопасность диалогового окна** свойств папки. На вкладке **Безопасность** щелкнуть кнопку **Добавить.** Откроется диалоговое окно **Выбор: Пользователи или Группы.**

В текстовом поле **Введите имена выбираемых объектов** ввести **uir-2,** затем щелкнуть кнопку **Проверить имена.** В текстовом поле **Введите имена выбираемых объектов** должна появиться надпись ***<имя компьютера>\тг-2.*** Это свидетельствует, что Windows XP Professional обнаружила пользователя **uir-2** на компьютере ***<имя компьютерен*** и что это действительная учетная запись пользователя. Щелкнуть **ОК,** чтобы закрыть диалоговое окно **Выбор: Пользователи или Группы**. Теперь пользователь **uir-2** включен в список Группы или пользователи диалогового окна свойств папки **Folderl.** Какие разрешения установлены для пользователя **uir-2?**

Щелкнуть кнопку **Дополнительно.** Откроется диалоговое окно **Дополнительные параметры безопасности** для **Folderl,** и вы увидите, что пользователь **uir-2** включен в список **Элементы разрешений.** Убедиться, что строка **uir-2** выделена, и щелкнуть кнопку **Изменить.** Откроется диалоговое окно **Элемент разрешения для Folderl,** и вы увидите в текстовом поле **Имя** учетную запись пользователя **uir-2.**

В колонке **Разрешить** щелкнуть **Полный доступ**. Теперь в колонке **Разрешить** установлены все флажки. Щелкнуть **ОК,** чтобы закрыть диало говое окно **Элемент разрешения для Folder!** и щелкнуть **ОК,** чтобы закрыть диалоговое окно **Дополнительные параметры безопасности для Folder 1.** Щелкнуть **ОК,** чтобы закрыть диалоговое окно свойств папки **Folder 1.**

Закрыть **Проводник** и завершить сеанс Windows XP Professional.

Войти в систему, используя учетную запись **uir-2.** Запустить **Проводник,** войти в папку **C:\Folderl.** Попытаться выполнить следующие операции с файлом **file2:** изменить файл; удалить файл. Какие действия вы смогли успешно совершить и почему? Завершить сеанс Windows 7.

Задание 8. Проверить, как разрешения NTFS наследуются в иерархии папок.

Войти в систему, используя учетную запись **uir-1.** Запустить **Проводник,** войти в папку **C:\Folderl\Folder2.** Создать текстовый файл с именем **Uir3** в папке. Завершить сеанс Windows

Войти в систему, используя учетную запись **uir-2.** Запустить **Проводник,** войти в папку **C:\Folderl\Folder2.** Попытаться выполнить следующие операции с файлом **Uir3:** открыть файл; изменить файл; удалить файл. Какие действия вы смогли совершить и почему? Завершить сеанс Windows 7.

Задание 9. Изучить результаты смены владельца файла.

Войти в систему, используя учетную запись **uir.** В папке **C:\Folderl** создать текстовый файл **file4.**

Щелкнуть правой кнопкой мыши значок документа **file4,** затем выбрать пункт меню **Свойства.** Откроется диалоговое окно **Свойства: file4**с активной вкладкой **Общие.** Перейти на вкладку **Безопасность** для просмотра разрешений, установленных для файла **file4.** Щелкнуть кнопку Дополни**тельно.** Откроется диалоговое окно **Дополнительные параметры** безо**пасности для file4** с активной вкладкой **Разрешения.** Перейти на вкладку **Владелец.** Кто является текущим владельцем файла **file4?**

Установка разрешения, позволяющего пользователю сменить владельца.

В диалоговом окне **Дополнительные параметры безопасности для file4** перейти на вкладку **Разрешения.** Щелкнуть кнопку **Добавить.** Откроется диалоговое окно **Выбор: Пользователи или Группы.** Убедиться, что в текстовом поле **Размещение,** которое расположено вверху диалогового окна, выбрано имя вашего компьютера. В текстовом поле Введите **имена выбираемых объектов** ввести **uir-3,** затем щелкните кнопку **Проверить имена.** Щелкнуть **ОК.**

Станет активным диалоговое окно **Элемент разрешения для file4.** Обратить внимание на то, что все элементы разрешений для пользователя **uir-3** не отмечены. В колонке **Разрешения** установить флажок **Разрешить** для разрешения **Сменить владельца.** Щелкнуть **ОК.** Щелкнуть **ОК** для того, чтобы вернуться к диалоговому окну свойств файла **file4.** Щелкнуть **ОК** для сохранения изменений и закрыть диалоговое окно свойств файла **file4.** Закрыть **Проводник** и выйти из системы. Смена владельца файла.

Войти в систему, используя учетную запись **uir-3.** Запустить **Проводник,** войти в папку **C:\Folderl.** Щелкнуть правой кнопкой мыши значок файла **file4** и выбрать пункт меню **Свойства.** Перейти на вкладку **Безопасность** для просмотра разрешений для файла. Щелкнуть **Дополнительно** и перейти на вкладку **Владелец.** В колонке **Изменить владельца на** выбрать **uir-3,** затем щелкнуть кнопку **Применить.** Кто теперь является владельцем файла **Uir4?**

Щелкнуть **ОК,** чтобы закрыть диалоговое окно **Дополнительные параметры безопасности для file4.**

Проверка разрешений для файла в качестве владельца.

Щелкнуть кнопку **Дополнительно** и снять флажок **Наследовать от родительского объекта применимые к дочерним объектам разрешения, добавляя их к явно заданным в этом окне.** Установить разрешение **Полный доступ** к текстовому документу **file4** и нажать кнопку **Применить.** Щелкнуть **ОК,** чтобы закрыть диалоговое окно **Дополнительные параметры безопасности для file4.** Щелкнуть **ОК,** чтобы закрыть диалоговое окно свойств файла **file4.**

Задание 10. Изучить изменение разрешений и прав владельца при копировании и перемещении папок. Создание папки при подключении с учетной записью пользователя.

Пока вы зарегистрированы в системе под учетной записью **uir-3** создать папку с именем **Tempi** в корневой папке диска **C:\.** Какие разрешения установлены для этой папки? Кто является владельцем папки?

Создание папок при подключении с учетной записью члена группы **Администраторы.**

Подключитесь с учетной записью **uir** и создайте папки **Temp2** и **Temp 3** в корневой папке диска **C:\.**

Каковы разрешения для папок, которые вы только что создали? Кто является владельцем папок **Temp 2** и **Temp 3?**

Установить разрешения для папок **Temp 2** и **Temp 3.**

Снять флажок **Наследовать от родительского объекта применимые к дочерним объектам разрешения, добавляя их к явно заданным в этом окне.** В открывшемся диалоговом окне щелкните **Удалить»** для удаления всех разрешений, кроме указанных ниже.

Папка **Temp 2: Администраторы — Полный доступ; Пользователи — Чтение и выполнение.**

Папка **Temp 3: Администраторы — Полный доступ; Операторы архива — Чтение и выполнение; Пользователи — Полный доступ.**

Копирование папки в другую папку на одном и том же томе NTFS.

Пока вы находитесь в системе под учетной записью **uir,** скопировать папку **C:\Temp2** в папку **C:\Templ.** Для этого выделить значок папки C:**Temp2** и, удерживая нажатой клавишу **CTRL,** перетащить мышью C:**Temp2**в **C:\Templ.**

Выделив **C:\Templ\Temp2,** просмотреть разрешения и права владельца, затем сравнить разрешения и права владельца с папкой **C:\Temp2.**

Перемещение папки на одном и том же томе.

Войти в систему с учетной записью **uir-3.** В **Проводнике** выделить значок папки **C:\Temp3,** затем переместить ее в папку **C:\Templ.** Что произошло с разрешениями и владельцем для папки **C:\Templ\Temp3?**

Задание 11. Самостоятельно определить? как предотвратить удаление пользователями, имеющими разрешение **Полный доступ** к папке, файла в этой папке, для которого установлен запрет на разрешение **Полный доступ?**

3.3.Контрольные вопросы

1. Что такое эффективные разрешения пользователя для ресурса?
2. Какие объекты по умолчанию наследуют разрешения, установленные для родительской папки?
3. Чем отличается разрешение «Удаление» от разрешения «Удаление подпапок и файлов»?
4. Какое разрешение NTFS для файлов следует установить для файла, если вы позволяете пользователям удалять файл, но не позволяете становиться владельцами файла?
5. Если вы хотите, чтобы пользователь или группа не имела доступ к определенной папке или файлу, следует ли запретить разрешения для этой папки или файла?

4. Лабораторная работа № 2 Работа с системным реестром

Цель работы

Получение основных сведений о структуре и функциях системного реестра операционной системы Windows 7

Оборудование: персональный компьютер (монитор, системный блок, клавиатура, мышь), ОС Windows, программа для виртуализации Oracle VM VirtualBox, образ виртуальной машины (Win7_01).

4.1. Задание:

1. Изучить теоретическую часть;
2. Запустить редактор реестра.
3. Перейти в раздел реестра HKEY_CURRENT_USER;
4. Найти ключ, отвечающий за настройки Рабочего стола;
5. Ознакомиться со списком вложенных ключей;
6. Для произвольно выбранных из списка 5 ключей исследовать, аналогом каких

7. Перейти в раздел реестра HKEY_CLASSES_ROOT;
8. Выбрать из списка 5 ключей и описать, для файлов с какими расширениями они используются, и какие параметры для них установлены;
9. Результаты внести в отчет.

4.2.Теоретические сведения:

На смену ini-файлам, имеющим ряд концептуальных ограничений, еще в Windows 3.1 было введено понятие реестра – регистрационной базы данных, хранящей различные настройки ОС и приложений. Изначально реестр был предназначен только для хранения сведений об объектах OLE (Object Linking and Embedding — связь и внедрение объектов) и сопоставлений приложений расширениям имен файлов, однако позже его структура и границы использования

расширились. Реестры разных версий Windows имеют различия; это нужно помнить при импорте reg-файлов. В Windows 7 в архитектуру реестра были введены важные новшества, улучшающие функциональность данного компонента ОС. Реестр хранится в бинарном (двоичном) виде, поэтому для ручной работы с ним необходима специальная программа — редактор реестра. В 7 это Regedit.exe, в других версиях NT ими являются Regedit.exe и Regedt32.exe, имеющий дополнительные возможности работы с реестром (Regedt32.exe есть и в 7, но на самом деле он всего лишь вызывает Regedit.exe). Есть и другие программы, в том числе и консольные (Reg.exe). Ручным модифицированием параметров реестра мы займемся чуть позже, а сейчас рассмотрим основные группы сведений, хранящихся в этой базе данных.

Программы установки. Любая грамотно написанная программа под Windows должна иметь свой инсталлятор-установщик. Это может быть встроенный в ОС Microsoft Installer либо любой другой. В любом случае инсталлятор использует реестр для хранения своих настроек, позволяя правильно устанавливать и удалять приложения, не трогая совместно используемые файлы.

Распознаватель. При каждом запуске компьютера программа NTDETECT.COM и ядро Windows распознает оборудование и сохраняет эту информацию в реестре.

Ядро ОС. Хранит много сведений в реестре о своей конфигурации, в том числе и данные о порядке загрузки драйверов устройств.

Диспетчер PnP (Plug and Play). Абсолютно необходимая вещь для большинства пользователей, которая избавляет их от мук по установке нового оборудования (не всегда, правда:)). Неудивительно, что он хранит свою информацию в реестре.

Драйверы устройств. Хранят здесь свои параметры.

Административные средства. Например, такие, как Панель управления, MMC (Micro-soft Management Console) и др.

Пользовательские профили. Это целая группа параметров, уникальная для каждого пользователя: настройки графической оболочки, сетевых соединений, программ и многое другое.

Аппаратные профили. Позволяют создавать несколько конфигураций с различным оборудованием.

Общие настройки программ. Почему общие? Потому, что у каждого пользователя есть профиль, где хранятся его настройки для соответствующей программы.

Таким образом, выше приведены данные о предназначении реестра. Теперь обратим внимание на логическую структуру реестра. Для лучшего понимания материала рекомендуется запустить Regedit.exe.

Структура реестра

Реестр Windows имеет древовидную структуру, схожую со структурой файловой системы. Папкам здесь соответствуют ключи (keys) или разделы (ветви), а файлам — параметры (values). Разделы могут содержать как вложенные разделы (sub keys), так и параметры. На верхнем уровне этой иерархии находятся корневые разделы (root keys).

HKEY_LOCAL_MACHINE

Содержит глобальную информацию о компьютерной системе, включая такие данные об аппаратных средствах и операционной системе, в том числе: тип шины, системная память, драйверы устройств и управляющие данные, используемые при запуске системы. Информация, содержащаяся в этом разделе, действует применительно ко всем пользователям, регистрирующимся в системе Windows. На верхнем уровне иерархии реестра для этого раздела имеются три псевдонима:

HKEY CLASSES ROOT, HKEY CURRENT CONFIG и HKEY_DYN_DATA HKEY_CLASSES_ROOT

Содержит ассоциации между приложениями и типами файлов (по расширениям имени файла). Кроме того, этот раздел содержит информацию OLE (Object Linking and Embedding), ассоциированную с объектами COM, а также данные по ассоциациям файлов и классов (эквивалент реестра ранних версий. Windows, служивших настройкой над MS-DOS). Параметры этого раздела совпадают с параметрами, расположенными в разделе HKEY_LOCAL_MACHINE\Software\Classes. Подробную информацию о разделе HKEY_CLASSES_ ROOT можно найти в руководстве OLE Programmer's Reference, входящем в состав продукта Windows NT 4.0 Software Development Kit (SDK)

HKEY_CURRENT_CONFIG

Содержит конфигурационные данные для текущего аппаратного профиля. Аппаратные профили представляют собой наборы изменений, внесенных в стандартную конфигурацию сервисов и устройств, установленную данными разделов Software и System корневого раздела HKEY_LOCAL_MACHINE. В разделе HKEY_CURRENT_ CONFIG отражаются только изменения. Кроме того, параметры этого раздела появляются также в разделе HKEY_LOCAL_MACHINE\System \CurentControlSet\HardwareProfites\CuiTent

HKEY_CURRENT_USER

Содержит, профиль пользователя, на данный момент . зарегистрировавшегося в системе, включая переменные окружения, настройку рабочего стола, параметры настройки сети, принтеров и приложений. Этот раздел представляет собой ссылку на раздел HKEY USERS\username, где username — имя пользователя, зарегистрировавшегося в системе на текущий момент HKEY_USERS Содержит все активно загруженные пользовательские профили, включая HKEY_CURRENT_USER, а также профиль по умолчанию. Пользователи, получающие удаленный доступ к серверу, не имеют профилей, содержащихся в этом разделе; их профили загружаются в реестры на их собственных компьютерах. Windows NT/2000 требует наличия учетных записей для каждого пользователя, регистрирующегося в системе.

Раздел HKEY_USERS содержит вложенный раздел \Default, а также другие разделы, определяемые идентификатором безопасности (Security ID) каждого пользователя

Типы данных

Все параметры реестра имеют фиксированный тип. В таблице 2 приводится полный список используемых типов. Не все из них используются в разных версиях NT

REG_QWORD явно предназначен для 64-битной версии 7. Следует учесть, что ряд типов используется только системой в некоторых разделах, и создать свой параметр такого типа с помощью редактора реестра не получится.

REG_BINARY

Двоичные данные. Большинство сведений об аппаратных компонентах хранится в виде двоичных данных и выводится в редакторе реестра в шестнадцатеричном формате

REG_DWORD

Данные, представленные целым числом (4 байта). Многие параметры служб и драйверов устройств имеют этот тип и отображаются в двоичном, шестнадцатеричном или десятичном форматах

REG_E7AND_SZ

Строка Unicode переменной длины. Этот типданныхвключаетпеременные, обрабатываемые программой или службой

REG_MULTI_SZ

Многострочный текст Unicode. Этот тип, как правило, имеют списки и другие записи в формате, удобном для чтения. Записи разделяются пробелами, запятыми или другими символами

REG_SZ

Текстовая Unicode строка фиксированной длины

REG_DWORD_LITTLE_ENDIAN

32-разрядное число в формате “остроконечников” — младший байт хранится первым в памяти. Эквивалент REG_DWORD

REG_DWORD_BIG_ENDIAN

32-разрядное число в формате “тупоконечников” — старший байт хранится первым в памяти REG_LINK Символическая ссылка Unicode. Только для внутреннего использования (некоторые корневые разделы являются такой ссылкой на другие подразделы)

REG_NONE

Параметр не имеет определенного типа данных

REG_QWORD
64-разрядное число

REG_QWORD_LITTLE_ENDIAN
64-разрядное число в формате “остроконечников”. Эквивалент REG_QWORD

REG_RESOURCE_LIST
Список аппаратных ресурсов. Используется только в разделе HKLM\HARDWARE

REG_FULL_RESOURCE_DESCRIPTOR
Дескриптор (описатель) аппаратного ресурса. Применяется только в HKLM\HARDWARE.

REG_RESOURCE_REQUIREMENTS_LIST
Список необходимых аппаратных ресурсов. Используется только в HKLM\HARDWARE.

Хранение реестра

Элементы реестра хранятся в виде атомарной структуры. Реестр разделяется на составные части, называемые ульями (hives), или кустами. Ульи хранятся на диске в виде файлов. Некоторые ульи, такие, как HKLM\HARDWARE, не сохраняются в файлах, а создаются при каждой загрузке, то есть являются изменяемыми (vola-tile). При запуске системы реестр собирается из ульев в единую древовидную структуру с корневыми разделами. Перечислим ульи реестра и их местоположение на диске (для NT старше версии 4.0)

HKLM\SYSTEM %SystemRoot%\system32\config\system

HKLM\SAM %SystemRoot%\system32\config\SAM

HKLM\SECURITY %SystemRoot%\system32\config\SECURITY

HKLM\SOFTWARE %SystemRoot%\system32\config\software

HKLM\HARDWARE Изменяемый улей

HKLM\SYSTEM\Clone Изменяемый улей
HKU\<SID_пользователя> %USERPROFILE%\ntuser.dat
HKU\<SID_пользователя>_Classes %USERPROFILE%\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
HKU\.DEFAULT %SystemRoot%\system32\config\default

Кроме этих файлов, есть ряд вспомогательных, со следующими расширениями:

ALT – резервная копия улья HKLM\SYSTEM (отсутствует в 7).

LOG – журнал транзакций, в котором регистрируются все изменения реестра. SAV – копии ульев в том виде, в котором они были после завершения текстовой фазы установки.

Дополнительные сведения

Реестр является настоящей базой данных, поэтому в нем используется технология восстановления, похожая на оную в NTFS. Уже упомянутые LOG-файлы содержат журнал транзакций, который хранит все изменения. Благодаря этому реализуется атомарность реестра – то есть в данный момент времени в реестре могут быть либо старые значения, либо новые, даже после сбоя. Как видим, в отличие от NTFS, здесь обеспечивается сохранность не только структуры реестра, но и данных. К тому же, реестр поддерживает такие фишки NTFS, как управление избирательным доступом и аудит событий – система безопасности пронизывает всю NT снизу доверху. Да, эти функции доступны только из Regedt32.exe или Regedit.exe для 7. А

еще весь реестр или его отдельные части можно экспортировать в текстовые reg-файлы (Unicode для Windows 2000 и старше), редактировать их в блокноте, а затем экспортировать обратно. Во многих редакторах реестра можно подключать любые доступные ульи реестра, в том числе и на удаленных машинах (при соответствующих полномочиях). Есть возможность делать резервные копии с помощью программы NTBackup.

4.3. Ход работы:

Для запуска системного реестра Windows 7 необходимо нажать кнопку <Пуск>, <Выполнить>, ввести команду <Regedit> и нажать <ОК>. Запуститься программа Редактор реестра .

Для перехода по разделам реестра необходимо выбрать соответствующий раздел

нажать кнопку раскрывающегося списка, находящуюся слева он названия раздела. Ключ, отвечающий за настройки рабочего стола находится по адресу

<HKEY_CURRENT_USER\Control Panel\Desktop>

Размеры элементов экрана в Windows (иконки, шрифты, рамки, меню, полосы прокрутки) хранятся в разделе HKEY_CURRENT_USER\Control Panel\desktop\WindowMetrics реестра

Ниже приведены некоторые параметры, содержащиеся в этом разделе.

BorderWidth Ширина рамки окна

CaptionFont Шрифт заголовка

CaptionHeight Высота шрифта заголовка

CaptionWidth Ширина заголовка

IconFont Шрифт названия иконки

IconSpacing Горизонтальный интервал между иконками

IconSpacingFactorФактор, используемый для вычисления положения иконок

IconVerticalSpacing Вертикальный интервал между значками

MenuFont Параметры шрифта (гарнитура, имя шрифта, и т.д.), используемого в строках меню

MenuHeight Высота ячейки символа, используемого в строке меню

MenuWidthШирина ячейки символа, используемого в строке меню

MessageFont Шрифт, используемый в сообщениях

ScrollHeight Высота горизонтальной полосы прокрутки

ScrollWidth Ширина вертикальной полосы прокрутки

ShellIconBPP Число цветов (битов на точку), используемых для иконок

ShellIconSize Размер иконок на Рабочем столе (и в проводнике в режиме "Крупные значки")

SmCaptionFont Шрифт в маленьких заголовках

SmCaptionHeight Высота ячейки символа в маленьком заголовке

SmCaptionWidth Ширина ячейки символа в маленьком заголовке

Каждый ключ, содержащий данные для шрифта, состоит из последовательности байтов, соответствующих имени шрифта и нескольким флагам, определяющим тип шрифта, типы начертания (полужирный, курсив) и т.д. Эти параметры можно изменять на вкладке «Оформление» диалога «Свойства: Экран».

Некоторые параметры настройки элементов экрана:

HKEY_CURRENT_USER\Control Panel\Desktop\WindowMetrics\ShellIconSize – управляет размером отображения значков рабочего стола. Значение 48 указывает, что значки рабочего стола будут отображаться размером 48x48 точек. Аналог <Свойства: Экран> / <Оформление> / <Эффекты> / <Применять крупные значки>.

HKEY_CURRENT_USER\Control Panel\Desktop\FontSmoothing – управляет сглаживанием неровностей экранных шрифтов. Аналог <Свойства: Экран> / <Оформление> / <Эффекты> / <Применять следующий метод сглаживания экранных шрифтов>.

HKEY_CURRENT_USER\Control Panel\Desktop\DragFullWindows – управляет отображением содержимого окна при его перетаскивании. Аналог <Свойства: Экран> / <Оформление>/<Эффекты>/<Отображать содержимое окна при его перетаскивании>.

HKEY_CURRENT_USER\Control Panel\Desktop\Wallpaper – содержит путь к файлу рисунка обоев, , аналог <Свойства: Экран> / <Рабочий стол>.

HKEY_CURRENT_USER\Control Panel\Desktop\SCRNSAVE.EXE – содержит путь к файлу с заставкой, аналог <Свойства: Экран> / <Заставка>.

Далее рассмотрим ключ реестра HKEY_CLASSES_ROOT

Корневой ключ реестра HKEY_CLASSES_ROOT содержит информацию обо всех ассоциациях (связях) расширений имен файлов, с приложениями, поддерживающими эти типы файлов, и о данных, ассоциированных с объектами COM. Эти данные совпадают с информацией, которая содержится в ключе classes, расположенной в иерархии ниже ключа HKEY_LOCAL_MACHINE\SOFTWARE.

Некоторые ключи раздела HKEY_CLASSES_ROOT:

HKEY_CLASSES_ROOT\.ico – определяет параметры файлов с расширением ico (значков, иконок);

HKEY_CLASSES_ROOT\.xls\Excel.Sheet.8\ShellNew – определяет параметры открытия файлов с расширением XLS (параметр Filename=excel9.xls);

HKEY_CLASSES_ROOT\.zip\ShellNew – определяет параметры открытия файлов

расширением ZIP(параметр Filename= C:\Program Files\WinRAR\zipnew.dat);

HKEY_CLASSES_ROOT\Excel.Template\shell\Print\command – определяет команды печати для шаблонов электронных таблиц Excel

HKEY_CLASSES_ROOT\jpg – определяет программу с которой ассоциированы файлы с расширением JPG

5. Практическое занятие № 3 Консоль администрирования Windows 7

Цель работы:
Практическое освоение инструмента управления операционной системы Windows 7 – графической консоли администрирования MMC.

Оборудование: персональный компьютер (монитор, системный блок, клавиатура, мышь), ОС Windows, программа для виртуализации Oracle VM VirtualBox, образ виртуальной машины (Win7_01).

5.1.Теоретическая часть

Консоль управления Microsoft Management Console (MMC) группирует средства администрирования, которые используются для администрирования сетей, компьютеров, служб и других системных компонентов.

Консоль MMC непосредственно не выполняет административные функции, однако предоставляет возможности интеграции в нее компонентов или системных приложений, выполняющие эти функции. Основной тип интегрируемых на консоль компонентов называется оснасткой, которые не могут выполняться отдельно без консоли. Среди других добавляемых элементов могут быть элементы управления ActiveX, ссылки на Web-страницы, папки, видов панели задач и собственно задачи для выполнения. Дополнительные теоретические сведения об оснастках и других используемых для интеграции на консоль элементах будут добавлены в дальнейшем, в соответствующих разделах настоящей лабораторной работы.

Базовое окно консоли MMC представляет собой графическую форму с контекстными меню, реализующие дружественный пользовательский интерфейс. Имеется панель инструментов с командами создания, открытия и сохранения консолей и, кроме того, область описания и строка состояния в нижней части окна. Чтобы увидеть базовое окно, а также непосредственно саму консоль MMC, необходимо выполнить следующие действия:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне MMC.exe (или просто mmc),
- нажмите Enter для ввода.

Новая консоль MMC представляет собой отдельное окно, разделенное на две вертикальные области, в левой из которых отображается дерево консоли с его корнем. Дерево консоли оказывает доступные элементы и компоненты консоли. Правая область является областью сведений, которая содержит описания элементов и выполняемых ими функций. Содержание области сведений соответствует

выбранному элементу в дереве консоли и может включать Web-страницы, графики, диаграммы, таблицы и столбцы.

Создавая надежные средства управления компьютерами сети, можно собрать и настроить собственную консоль MMC, выполняющую заданные функции администрирования. После того как добавлены все необходимые элементы и компоненты консоли, панель главного меню, панель инструментов, а также область описания и строка состояния могут быть скрыты для предотвращения в дальнейшем нежелательных изменений. Созданные таким образом управляющие системы сохраняются в файлах с расширением .msc (Management Saved Console, сохраненная консоль управления) и могут быть, в частности, распространены в пределах всей системы посредством задания к ним доступа с помощью ярлыков или элементов меню Пуск.

Чтобы увидеть консоль управления локальным компьютером в качестве примера готовой и отлаженной консоли MMC, необходимо выполнить:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне compmgmt.msc (или compmgmt),
- нажмите Enter для ввода.

Существует два основных режима доступа консоли администрирования, задающиеся непосредственно при ее создании:

- пользовательский, в котором можно администрировать систему, работая с уже существующими консолями,
- авторский, в котором можно создавать новые консоли или изменять существующие.

В свою очередь, имеется три уровня режима пользователя, что обусловливает всего четыре варианта предустановленного режима доступа:

- авторский режим;
- режим пользователя — полный доступ;
- режим пользователя — ограниченный доступ, многооконный;
- режим пользователя — ограниченный доступ, однооконный.

Консоль MMC, инициализированная в авторском режиме, предоставляет полный доступ ко всем ее возможностям, включая добавление и удаление оснасток, создание новых окон и панелей задач, а также просмотр любых частей дерева консоли и другие. Однако при выборе одного из трех режимов пользователя авторские

возможности исключаются. В частности, если для консоли установлен параметр «пользовательский режим — полный доступ», то предоставляются все команды управления окном консоли и полный доступ к ее дереву, но запрещается добавление, удаление оснасток и изменение свойств консоли администрирования.

Изменения консоли ММС в авторском и пользовательском режимах сохраняются по-разному. При закрытии консоли в авторском режиме выводится диалоговое окно с предложением сохранить изменения. Однако в пользовательском режиме и снятом флажке «Не сохранять изменения для этой консоли» изменения будут сохранены автоматически при закрытии.

Если консоль открыта при соблюдении одного из следующих условий:

- в базовом окне при загрузке,
- с помощью команды контекстного меню Автор,
- в командной строке с параметром /a,

то предустановленный режим игнорируется, а открытие консоли осуществляется в авторском режиме.

Очевидно, что загрузка консоли ММС в авторском режиме не требуется рядовым пользователям. Системный администратор может настроить профили пользователей так, чтобы запретить им переход в авторский режим, как из командной строки, так и через контекстное меню. Кроме того, запрет перехода в авторский режим может быть организован при использовании возможностей групповой политики, при которой, в частности, осуществляется ограничение доступа к определенным оснасткам. Рассмотрению базовых возможностей оснастки групповой политики будет посвящена вторая часть настоящей лабораторной работы.

Прежде чем создавать новую консоль ММС, необходимо определить действия, для которых предназначена эта консоль, список администрируемых компонентов, оснасток и других элементов, которые потребуются для выполнения поставленных задач. Следует также рассмотреть необходимость создания видов панели задач. После принятия этих решений можно открыть новую консоль и начать добавлять элементы к дереву консоли. Полное руководство по созданию и настройке консолей ММС находится на Web-узле корпорации Майкрософт (http://www.microsoft.com).

В лабораторной работе предполагается ознакомление с основными принципами организации и построения консоли администрирования ММС, а также с базовыми возможностями основных инструментов системного администратора — оснасток «Локальные пользователи и группы» и «Редактор объекта групповой политики» («Групповая политика»).

Перед началом выполнения заданий в среде ОС Windows 7 необходимо выполнить следующее:

1. запустить виртуальную машину с ОС Windows 7 и активировать справочное меню (Пуск | Справка и поддержка);

2. ознакомиться с описанием и возможностями запуска и применения консоли администрирования MMC;

3. ознакомиться возможностью получения сведений пункта 2 из альтернативного источника информации, доступного непосредственно в справке консоли администрирования MMC (Справка | Вызов справки);

4. ознакомиться с описанием и возможностями оснасток «Локальные пользователи и группы» и «Редактор объекта групповой политики» («Групповая политика»).

5.2. Практическая часть

Задание 1. Изменение параметров и способов настройки консоли администрирования MMC

Порядок выполнения:

I. Создание консоли администрирования MMC в авторском режиме требует выполнения следующих действий:

- нажмите Пуск | Выполнить,
- наберите в появившемся окне MMC.exe (или просто mmc),
- нажмите Enter для ввода.

Возможны следующие альтернативные варианты авторского запуска консоли администрирования:

A. запуск из командной строки, используя синтаксис:

Mmc путь\имя_файла.msc /a,

где параметр:

путь\имя_файла.msc — запускает консоль MMC с одновременным открытием файла сохраненной консоли с именем имя_файла.msc (Приложение 2). Если файл консоли не указан, будет открыта новая консоль MMC.

/a — открывает консоль MMC в авторском режиме.

Дополнительными параметрами команды могут быть:

/64 — открывает 64-разрядную версию консоли MMC (MMC64). Этот параметр используется только при работе в ОС Windows 7 64-Bit Edition.

/32 — открывает 32-разрядную версию консоли MMC (MMC32). При работе в ОС Windows 7 64-Bit Edition в окне консоли MMC, запущенной с этим параметром, открываются 32-разрядные оснастки.

В. запуск из файлового менеджера Проводник ОС Windows 7:

• наведите манипулятор мышь на файл с расширением .msc, находящийся в системной папке ОС (%systemroot%\system32\),

• кликните правой кнопкой мыши на файле и из контекстного меню выберите Автор.

II. Настройка параметров консоли администрирования MMC предназначена для ее конфигурирования с целью придания ей уникального вида.

Содержание задания

Для придания уникального вида сохраненной (новой) консоли администрирования MMC в авторском режиме выполните следующие действия:

1. В меню Консоль выберите команду Параметры.

2. На вкладке Консоль в поле названия введите новый заголовок.

3. На вкладке Консоль выполните следующие действия:

• нажмите кнопку Сменить значок,

• в поле Имя файла введите путь к файлу, содержащему значки (например,

%systemroot%\system32\shell32.dll),

• в поле Текущий значок выберите необходимый значок,

• кликните ОК для ввода и Применить для подтверждения.

4. На вкладке Консоль из списка Режим консоли выберите пользовательский режим с полным доступом, в котором будет открываться консоль MMC при ее непосредственном запуске,

5. Для установленного в предыдущем пункте режима выполните указанные ниже действия:

• запретите изменение консоли MMC при ее непосредственном запуске, установив флажок «Не сохранять изменения для этой консоли», сделайте активным

диалоговое окно Вид | Настройка вида консоли ММС при запуске, установив флажок «Позволить пользователю настраивать вид консоли»,

6. Если необходимо удалить файлы, содержащие параметры отображения файлов консоли, на вкладке Очистка диска нажмите кнопку Удалить файлы.

7. Сохраните окончательно сконфигурированную консоль администрирования ММС, выбрав самостоятельно ее имя и путь к месту расположения в меню Консоль | Сохранить как… При сохранении обратите внимание на то, что файлы консоли по умолчанию

размещаются в папке «Администрирование», имеющей полный путь %Pathname%\Главное меню\Программы\Admini-стрирование\.

8. Закройте сконфигурированную и сохраненную консоль администрирования ММС, выполнив соответствующие необходимые действия.

В файловом менеджере Проводник ОС Windows 7 выполните следующие инструкции:

- наведите манипулятором мышь на сохраненный файл консоли администрирования ММС и, дважды кликнув на нем, запустите консоль,
- откройте диалоговое окно Вид | Настройка вида и, изменяя положение флажков, обратите внимание на получаемый результат,
- изменив вид консоли ММС приемлемым образом, кликните ОК для подтверждения полученного результата,
- в контекстном меню Консоль кликните Выход,
- снова запустите консоль администрирования ММС, кликнув манипулятором мышь на сохраненном файле консоли,
- изучите полученный результат,
- сделайте вывод о проделанной работе и запишите его в отчет.

Задание 2. Добавление различных элементов и компонентов к дереву консоли администрирования ММС

Основным, интегрируемым на консоль компонентом является оснастка. Оснастки существуют двух видов:

- изолированные,
- расширения.

Изолированная оснастка (или просто оснастка) добавляется к дереву консоли ММС без предварительного добавления других элементов, то есть непосредственно в корень дерева консоли.

Оснастка расширения (или просто расширение) всегда добавляется к другой изолированной оснастке или расширению, которые уже имеются в дереве консоли ММС. Если для определенной оснастки разрешены расширения, то, как правило, они работают с объектами, управляемыми непосредственно этой оснасткой, например, с компьютером, принтером, модемом или другим внешним устройством.

В дереве консоли оснастки и расширения располагаются для удобства иерархически или по группам. При добавлении новой оснастки или расширения, они появляются в виде нового элемента в дереве консоли ММС или в виде нового пункта контекстного меню, дополнительной панели инструментов, страницы свойств, а также возможно мастера, организующего определенную последовательность действий, к уже установленной оснастке.

Другими элементами, по необходимости применимыми для интеграции на консоль администрирования ММС, являются виды панели задач и собственно задачи, которые могут включать в себя команды меню для элементов консоли и команды, запускаемые из командной строки. Кроме того, могут быть созданы команды, действующие как часть дерева консоли или открывающие другой компонент.

Прежде всего, перед добавлением указанных элементов к консоли ММС, необходимо определить их число. Если, в частности, требуется добавить несколько видов панели задач, то наряду с этим необходимо определить тип каждой панели (для отображения списка и задач или только задач), а также разделить задачи по интегрированным видам панели. Добавление видов панели задач и собственно задач осуществляется посредством работы мастера создания этих элементов. При этом важно помнить, что консоль ММС должна содержать, по крайней мере, одну оснастку, чтобы возможность интеграции появилась в принципе.

Отдельной возможностью, иногда необходимой при администрировании сетей, является добавление элементов и компонентов дерева консоли администрирования ММС в виде списка ярлыков в меню «Избранное».

Дополнительные сведения о добавлении различных элементов в дерево консоли администрирования ММС можно получить, воспользовавшись справкой ОС Windows 7 (Пуск | Справка и поддержка) в разделе Общее представление о ММС \ Консоль ММС в авторском режиме \ Оснастки \ Создание консолей.

Содержание задания

Первым необходимым компонентом, добавляемым к дереву консоли администрирования ММС при ее организации и построении, является оснастка. Для добавления оснастки в авторском режиме выполните следующие действия:

1. Создайте новую Консоль управления ММС одним их описанных в пункте I текущего учебного задания способов.

2. В меню Консоль выберите команду Добавить или удалить оснастку.

3. В диалоговом окне Добавить/удалить оснастку нажмите кнопку Добавить вкладки Изолированная оснастка. Список Оснастки в диалоговом окне Добавить/удалить оснастку определяет элемент дерева консоли, к которому выполняется добавление элементов. В этом списке можно найти любой элемент дерева консоли. Обратите внимание на то, что по умолчанию это Корень консоли.

4. В диалоговом окне Добавить изолированную оснастку, выберите оснастки Службы из списка доступных в системе, кликнув на ней манипулятором мышь и нажав кнопку Добавить. Для добавления другой оснастки из списка, повторите указанные действия настоящего пункта повторно.

5. Для некоторых оснасток в процессе их инсталляции выводится диалоговое окно Выбор целевого компьютера, определяющее чем, устанавливаемая оснастка, будет управлять в дальнейшем — локальным или сетевым компьютером. Выберите Локальный компьютер, установив переключатель в соответствующее положение.

6. Нажмите Готово, Закрыть и затем кликните OK для подтверждения ввода.

7. Скройте меню и панель инструментов оснастки Службы, выполнив действия указанные ниже:

- В меню Вид выберите команду Настроить,
- В группе Оснастка снимите флажок Меню,
- В группе Оснастка снимите флажок Панели инструментов.

При устанавливании или снятии флажков, соответствующие им меню и панели инструментов отображаются или скрываются, причем, для всех оснасток консоли, включая текущую. Если переключение флажков не приводит к изменению вида консоли, тогда текущая оснастка не имеет специальных меню или панелей инструментов.

8. Не закрывая консоль администрирования MMC, сохраните ее, выбрав команду Сохранить в меню Консоль.

Для добавления расширений к уже установленной в предыдущем задании оснастке Службы выполните следующее:

9. В меню Консоль выберите команду Добавить или удалить оснастку.

11. В диалоговом окне Добавить/удалить оснастку выберите вкладку Расширение. На этой вкладке можно выбрать любой элемент дерева консоли из списка Оснастки, которые могут быть расширены, и просмотреть Доступные расширения, которые могут быть включены или отключены. После подключения расширение автоматически размещается в дереве консоли под оснасткой, к которой оно относится. Если дерево консоли содержит больше одного экземпляра оснастки, к которой подключено расширение, все остальные экземпляры автоматически получают это расширение. Среди Доступных расширений оснастки Службы удалите флажок с расширения Расширенный вид и отметьте к чему привело это действие. Повторите аналогичные действия с другими расширениями данной оснастки и изучите получаемый результат.

12. Не закрывая консоль администрирования MMC, сохраните ее. В окне консоли администрирования выполните следующие инструкции:

- последовательно перебирая доступные в системе оснастки, найдите те из них, которые обладают дополнительным меню, панелью инструментов или расширениями,
- изучите полученный результат и сделайте вывод о проделанной работе,

6. Лабораторная работа № 3 Знакомство с операционной системой UBUNTU server 16.04

6.1. Теоретический материал

Ubuntu ([ʊˈbʊntuː]; от зулу *ubuntu* — *человечность*[6]*; «Убу́нту»*) — операционная система, основанная на Debian GNU/Linux. Основным разработчиком и спонсором является компания Canonical. В настоящее время проект активно развивается и поддерживается свободным сообществом[7].

Ubuntu используется примерно 20 миллионами пользователей[8]. Он является 1-м в списке самых популярных дистрибутивов Linux для веб-серверов. По количеству пользователей, посетивших сайт DistroWatch.com (на апрель 2016 года) занимает 3-е место[9].

Обычно новые версии дистрибутива выходят каждые полгода и поддерживаются обновлениями безопасности в течение 9 месяцев (начиная с версии 13.04, до этого поддержка осуществлялась в течение полутора лет).

Версии LTS,[10] выпускаемые раз в 2 года, поддерживаются в течение 5 лет — как серверные, так и десктопные варианты[11][12][13]. (До версии 12.04 LTS срок поддержки для десктопных LTS-версий составлял 3 года.) На другие дистрибутивы LTS семейства Ubuntu действует полная поддержка в 3 года,[14] а для основы системы (ядро, Xorg и прочие компоненты) — 5 лет.

Ubuntu поставляется с подборкой программного обеспечения для серверов и рабочих станций. Она устанавливается на настольные персональные компьютеры с помощью LiveCD (версия *Desktop*), LiveUSB или текстового установщика (версия *Alternate*, предоставлялась до версии Ubuntu 12.04.2)[15][16]. В версии LiveDVDприсутствуют несколько бóльшие возможности — начиная от установки не только в графическом,[17] но и в текстовом[18] режимах, загрузки в режиме восстановления системы и заканчивая полной локализацией и бóльшим количеством пакетов на диске. Есть версии для официально поддерживаемых архитектур, таких как i386, AMD64, ARM. Кроме того, с 2013 года начата разработка специальной версии Ubuntu для смартфонов на архитектуре ARM и x86[1

Текущим LTS-релизом является Ubuntu 16.04.2 LTS. Начиная с версии 13.04 Raring Ringtail, поддержка для рядовых (не LTS) релизов составит 9 месяцев вместо 18[112].

Логотип версии Ubuntu 16.04 представляет собой схематично нарисованное животное:

Языки программирования

Go 1.6

Ubuntu 16.04 включает в себя релиз Go 1.6, который состоялся в феврале этого года.

PHP 7

Репозиторий Ubuntu 16.04 по умолчанию содержит PHP версии v7.0. Одной из существенных особенностей PHP 7 является значительно возросшая по сравнению со старыми версиями производительность. Также в PHP 7 есть изменения, касающиеся синтаксиса, поэтому если вы разрабатываете приложение на базе PHP 5, вам нужно выполнить миграцию согласно инструкции на официальном сайте.

Python 3.5

Репозиторий Ubuntu 16.04 по умолчанию укомплектован Python версии 3.5.1. Если вам необходима предыдущая версия Python, то ее можно установить следующей командой:

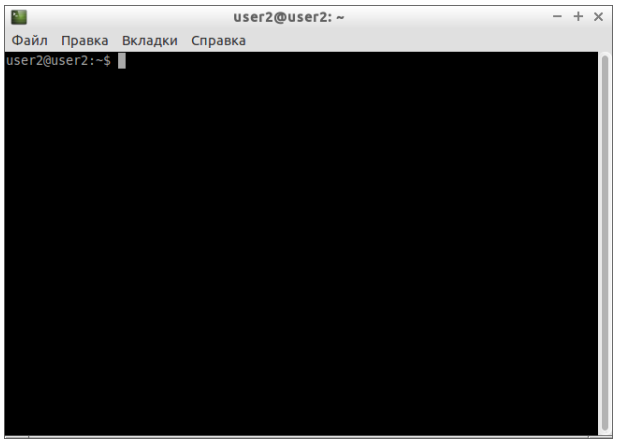
sudo apt-get install python

Тем, кто использует Vim как основной элемент разработки на Python, необходимо иметь в виду, что теперь этот редактор использует Python 3 в качестве стандартной версии Python, а значит, плагины, написанные на Python 2, могут перестать работать.

Управление Ubuntu server 16.04

Управление сервером осуществляется как правило с терминала, в консольном режиме. Графическая оболочка здесь не нужна, так как она требует дополнительных ресурсов сервера, а пользователем здесь является опытный системный администратор. Кроме того, лоступ к серверу требуктся достаточно редко.

Для входа в терминал необходимо с консоли ввести логин и пароль пользователя, обычно это данные системного администратора, они вводятся при установке системы. Следует обратить внимание, что прри вводе пароля вводимые символы не отображаются ни в каком виде, просто в терминале ничего не происходит, пока вы не нажмёте клавишу enter. После этого выводятся сведения о системе и приглашение для ввода команд с указанием текущего директория (папки пользователя).



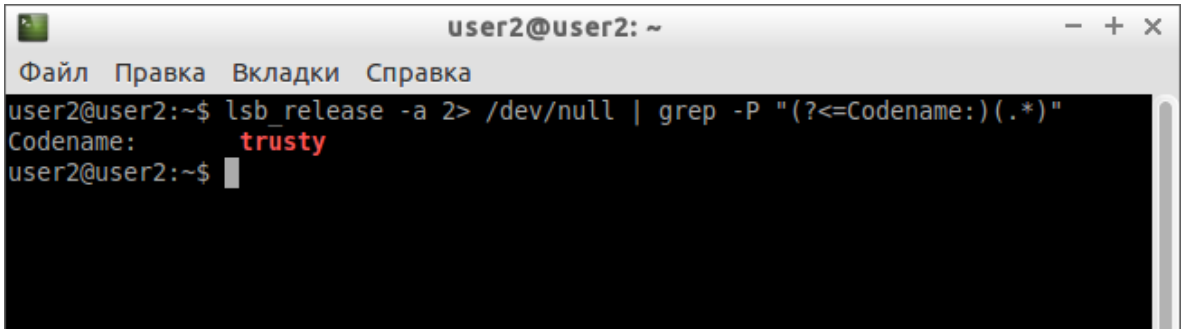
Знакомьтесь, это и есть терминал. Он создан для того, чтобы выполнять текстовые команды, поэтому отложите свою мышку в сторону и пододвиньте поближе клавиатуру[1)].

Управление терминалом

Давайте выполним какую-нибудь команду, например:

lsb_release -a 2> /dev/null | grep -P "(?<=Codename:)(.*)"

На выходе получим кодовое имя нашего дистрибутива:

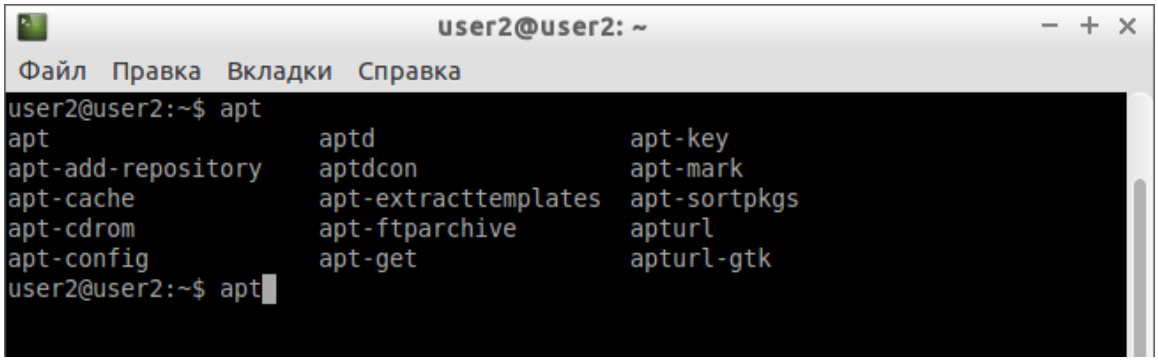


Набирать такие команды с клавиатуры посимвольно немного неудобно, поэтому давайте сразу разберёмся с основами управления терминалом. Начнём с копирования/вставки. Стандартные сочетания клавиш Ctrl+C и Ctrl+V в терминале не работают, вместо них используется старая добрая пара Ctrl+Insert с Shift+Insert или же сочетания с Shift: Ctrl+Shift+Cдля копирования и Ctrl+Shift+V для вставки. Что ж, теперь вы умеете целиком копировать команды из руководств.

Кстати, в большинстве руководств и инструкций вы встретите именно терминальные команды. Это связано с тем, что, графических оболочек очень много, и объяснить, как выполнить какое-то действие для каждой из них бывает очень непросто. А терминал - один для всех, и одна и та же команда работает во всех оболочках (естественно, кроме команд по настройке самой оболочки). К тому же, намного проще дать одну команду, чем объяснить где и как 10 раз нажать мышкой.

Однако, часто всё-таки команды приходится набирать вручную, а не вставлять откуда-то. И вот тут на помощь приходит великолепное свойство терминала, называемое автодополнением. Наберите в терминале символы apt-g, а потом нажмите клавишу Tab. Терминал автоматически дополнит за вас команду. Кстати, apt-get - это основная консольная утилита управления пакетами, но об этом после.

А теперь попробуйте набрать только apt и нажать Tab. Ничего не происходит? А теперь нажмите Tab два раза подряд. Видите, терминал выдал вам список всех команд, начинающихся с apt.



Удобно, не правда ли? Особенно, если привыкнуть.

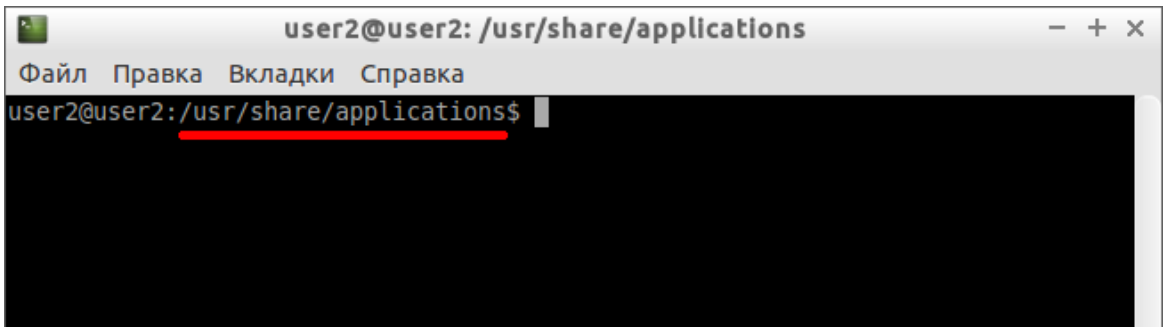
Автодополнение в терминале работает практически везде, и не только для команд, но так же для их аргументов и имён файлов. Поэкспериментируйте с ним, оно значительно сокращает время набора, да и вообще, терминал без автодополнения - это не терминал.

Еще одна хитрость. Откройте терминал нажмите сочетание клавиш Ctrl+R и начните набирать нужную команду. Терминал автоматически подставит подходящие варианты из набранных ранее команд.

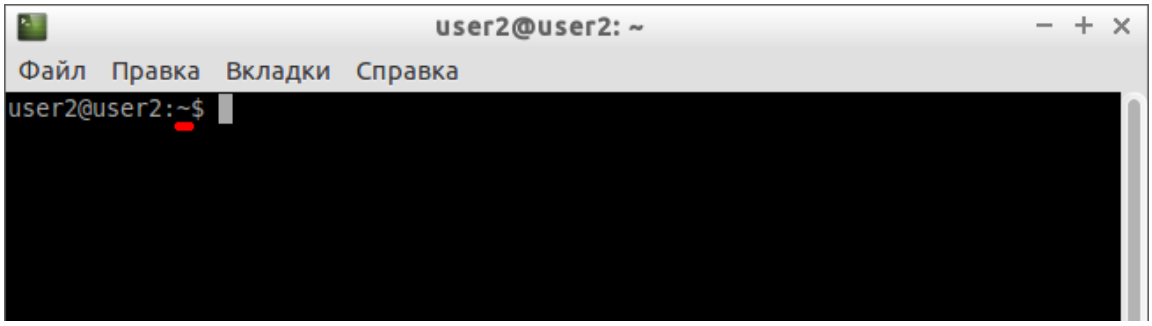
Работа с файлами

Начну рассказ наверно с того, что в любой момент времени работы в терминале вы находитесь в некотором каталоге. При запуске терминала текущей директорией является домашний каталог пользователя, но потом вы конечно можете её поменять.

Узнать, в какой же папке вы сейчас находитесь, очень просто, достаточно посмотреть на приглашение терминала, то есть на те символы, которые печатаются автоматически в начале каждой строки:



Текущий каталог - это то, что между символами : и $. Кстати, обратите внимание, перед : стоит имя пользователя и имя компьютера, разделённые символом @. Но когда вы запускаете терминал, то между : и $ стоит символ ~:



При запуске терминала текущим каталогом становится домашняя папка пользователя, так вот, символ ~ - это синоним адреса домашней папки текущего пользователя[2)]. Кстати, если вы ещё не знаете, полный адрес домашнего каталога выглядит как

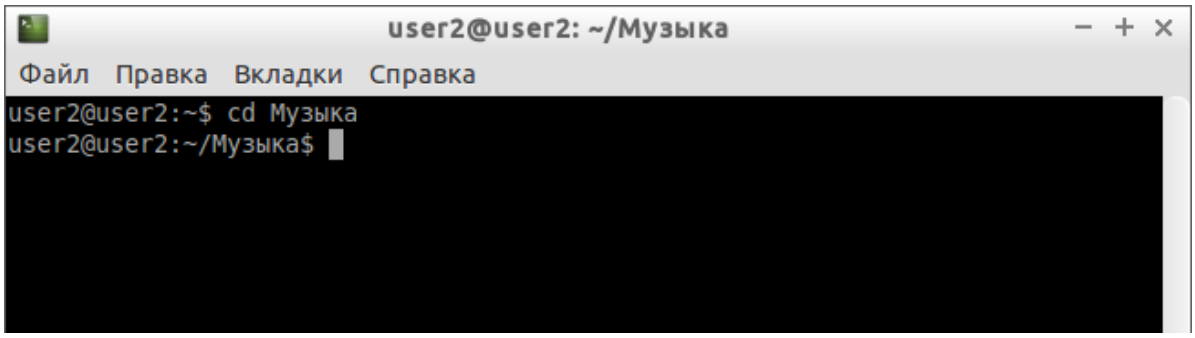
/home/логин_пользователя

В данном случае это /home/user2.

Хорошо, как выяснить текущее местоположение, вроде разобрались, а как же его сменить? Для этого предназначена команда cd, выполните, например, команду

cd Музыка

Видите, текущее местоположение изменилось:



На всякий случай напоминаю, что имена файлов и каталогов в Linux чувствительны к регистру символов, то есть Музыка и музыка - эта два совершенно разных имени.

После команды cd можно указывать как полные пути относительно корня, так и относительные, отсчитывающиеся от текущего каталога. В примере выше я использовал относительный путь. А вот полный:

cd /home/user2/Музыка

Хочу сразу обратить внимание на несколько важных особенностей. Во-первых, при наборе путей так же работает автодополнение по Tab, это очень удобно. Во-вторых, использовать различные небуквенные символы и пробелы напрямую при наборе путей нельзя. Например, для того, чтобы перейти в каталог, содержащий в имени символ пробела, надо при наборе пути к такому каталогу перед пробелом поставить символ обратного слеша \. Вот так:

cd Каталог\ с\ плохими\ символами\ в\ имени\<\>

Установка обратного слеша перед некоторыми символами называется *экранированием*. Кстати, при использовании автодополнения все слеши расставляются автоматически. Кроме того, можно просто заключить путь в двойные кавычки:

cd "Каталог с плохими символами в имени<>"

Но в этом случае автодополнение работать не будет.

Заменитель адреса домашнего каталога ~ можно использовать и при наборе путей, например:

cd ~/Музыка

А для перемещения непосредственно в домашний каталог достаточно просто набрать cd без аргументов.

Для перемещения на каталог выше можно использовать команду

cd ..

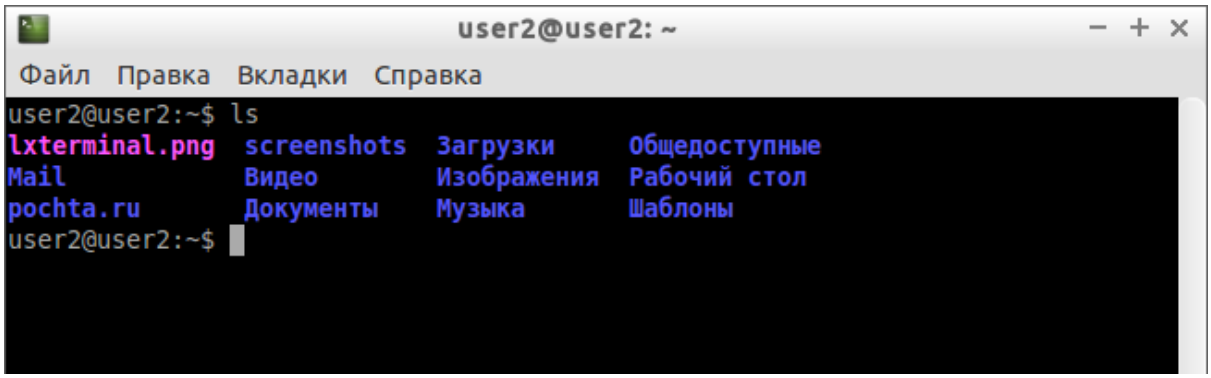
Вообще, две точки обозначают всегда родительский каталог, поэтому можно делать так:

cd ../..

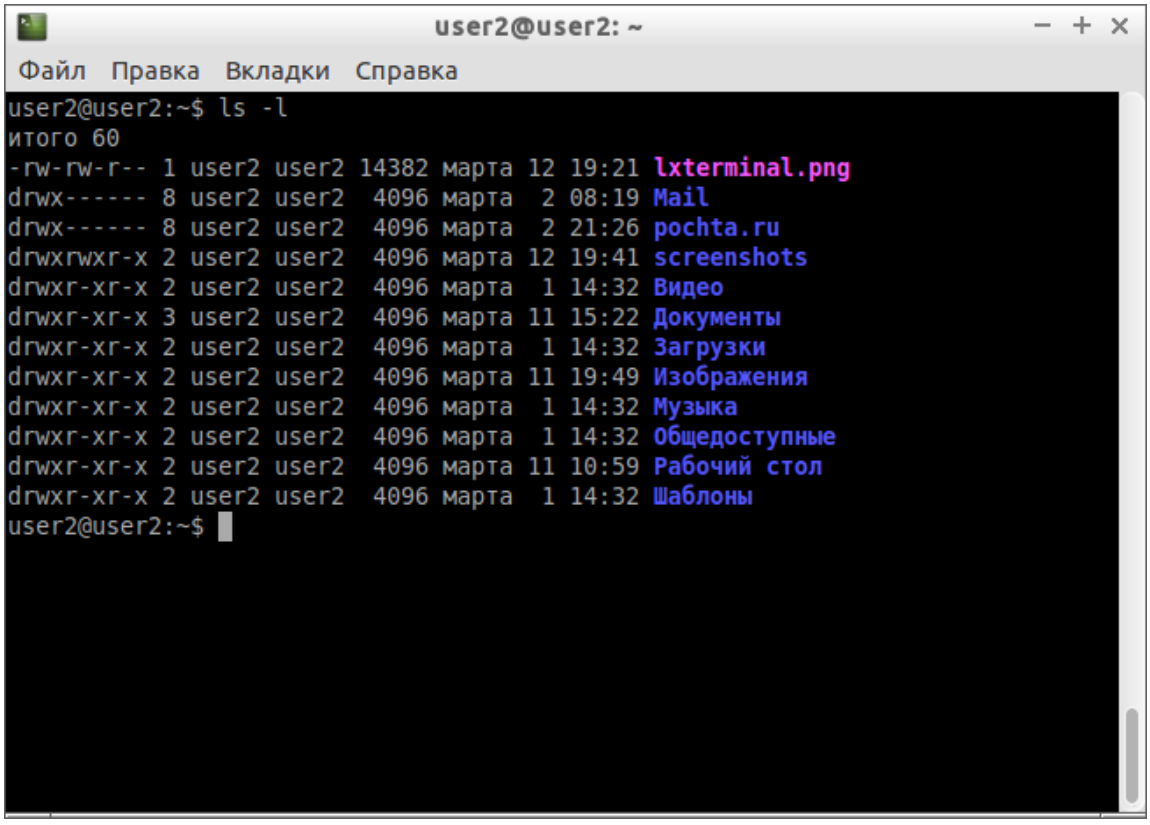
В принципе, всё можно как угодно комбинировать, в разумных пределах, конечно. Ну и напоследок про cd. Переместиться в предыдущий посещённый каталог можно командой

cd -

Как перемещаться по каталогам в первом приближении вроде разобрались, теперь же я расскажу про некоторые другие полезные операции. Посмотреть содержимое текущего каталога можно командой ls:



Обычно командам можно передавать различные модификаторы, например:



Кстати, эта команда показывает различную дополнительную информацию о содержимом каталога.

Очень часто параметрами команд являются имена файлов или папок, именно для этого я так подробно рассказывал о способе перемещения по каталогам и использования различных имен файлов в качестве аргументов. Например, команда cat показывает содержимое текстового файла, и если вы хотите посмотреть содержимое файла test.txt, лежащего в вашем домашнем каталоге, то вы могли бы выполнить команду ***cat ~/text.txt***

Получение справки

Начинающих пользователей Linux очень часто любят пугать так называемыми манами. Дело в том, что man - это система справки о командах для терминала. Пользоваться ей очень легко, просто наберите в терминале

man команда

Например:

man ls

Появится собственно текст справки[3)], разбитый на разделы. Перемещаться по нему можно с помощью стрелок и клавиш PgUpи PgDown, а для выхода просто нажмите Q.

Кроме man-страниц у многих утилит[4)] есть встроенная справка, которую обычно можно посмотреть, запустив программу с ключом --help:

утилита --help

Например:

ls --help

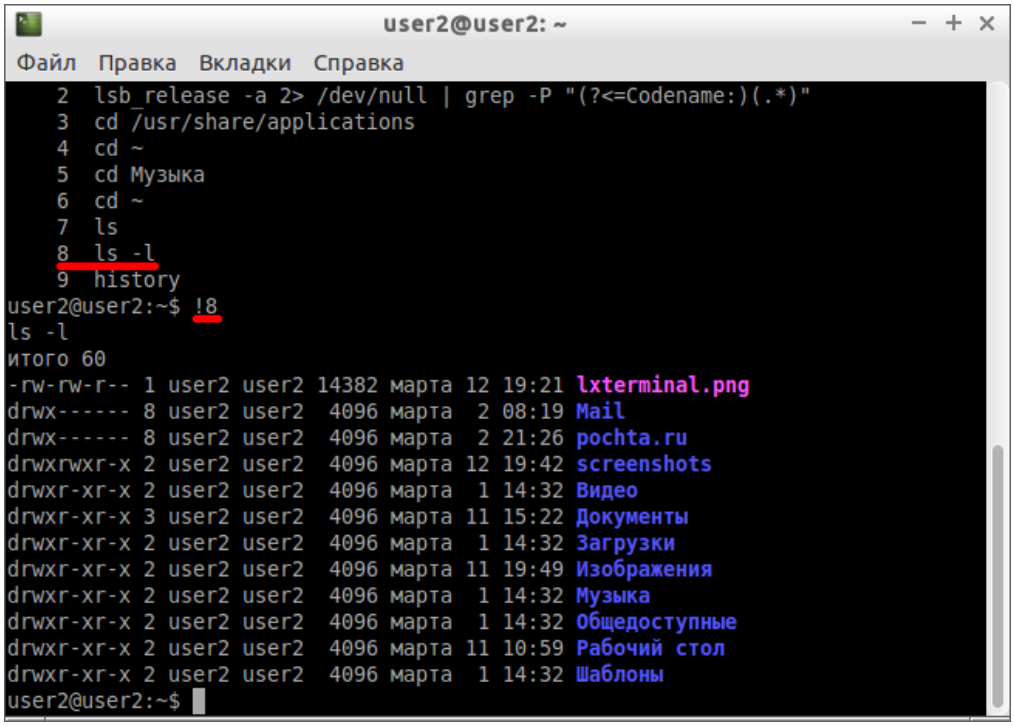
Есть и другие способы получения помощи, например похожая на man утилита info. Но чаще всего наиболее полную информацию о программе можно получить именно из man-страниц, а краткую справку - указав ключ --help при вызове.

История введённых команд

Терминал хранит историю введённых пользователем команд, которую вы можете листать в реальном режиме стрелками вверх и вниз на клавиатуре. Это очень

удобно для повторного исполнения введённых ранее команд. А посмотреть всю историю можно командой ***history***

У каждой команды в истории есть номер, выполнить снова команду с определённым номером можно набрав в терминале восклицательный знак и номер нужной команды:



А повторить предыдущую набранную команду можно просто написав два восклицательных знака !! Двигаться по истории набранных команд можно стрелочками вверх/вниз клавиатуры. Нажав один раз на стрелку вверх - терминал покажет последнюю набранную команду, два раза - предпоследнюю, и так далее. Если нужно вернуться назад - нажмите стрелочку вниз.

Мне кажется, что с непривычки уже всё описанное может показаться дремучим лесом, а на самом деле это только самая верхушка айсберга, существуют ещё тысячи полезных команд и интересных приёмов работы в терминале. С помощью терминала можно редактировать файлы, слушать музыку, смотреть видео и выполнять ещё массу повседневных операций.

Дополнительные возможности терминала

Бывает так, что вы что-то запустили в терминале и хотите прервать работу этого чего-то. Обычно это сделать очень просто, достаточно нажать на клавиатуре сочетание клавиш Ctrl+C[6)].

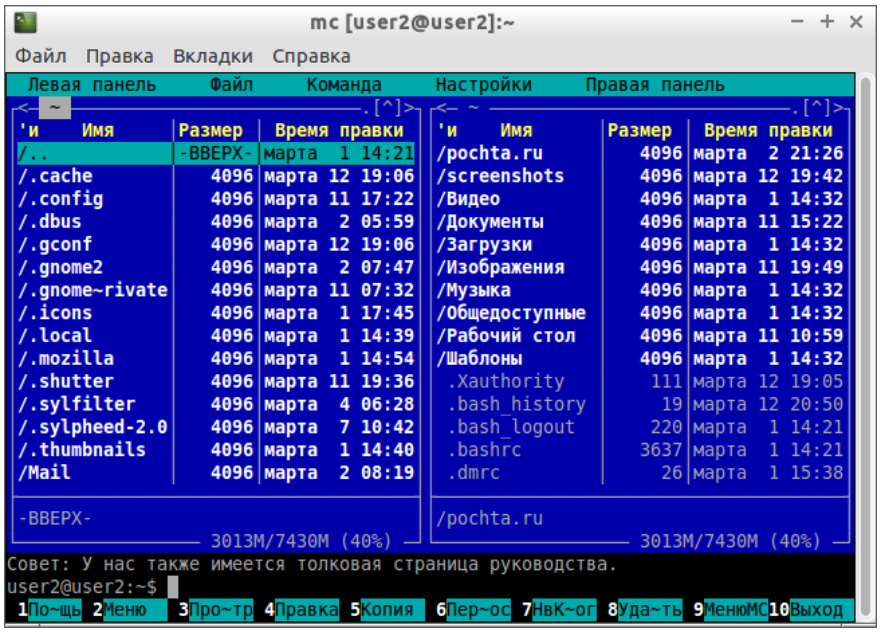
Есть и другие управляющие сочетания, например Ctrl+D посылает сигнал конца файла запущенному приложению, а без запущенных утилит делает тоже, что и терминальная команда exit. Ну а если вы хотите более подробно управлять работающими программами, то посмотрите на системный монитор htop, который, правда, нужно доустанавливать отдельно.

Если у вас сложилось впечатление, что терминал - это какая-то примитивная программа, способная выполнять очень простые команды, то это ложное впечатление. На самом деле есть очень много консольных утилит с богатейшими возможностями. Например, как уже упоминалось выше, серверные версии Ubuntu поставляются без графической оболочки. С помощью только консольных утилит можно настроить и управлять сложнейшими многофункциональными серверами.

Рассмотрим простой консольный файловый менеджер, он доустанавливается отдельно. Введите в терминале команду

mc

и увидите, что получится:



У пользователей постарше, успевших поработать в DOS, это окно может вызвать острый приступ ностальгии.

6.2.Задание:

1. Выполнить импорт конфигурации виртуальной машины – сервера UbuntuServer01 в Oracle VM VirtualBox

2. Пока идёт импорт (около 15 минут) изучить теоретический материал. Сомостоятельно выполнить поиск статей в Интернет, посвящённых Ubuntu Server.
3. Запустить виртуальную машину UbuntuServer01. Войти в терминал с логином admin1 и паролем 1.
4. Добавьте себя в список пользователей системы командой sudo adduser <*ваша фамилия*>, (латиницей) ,система потребует пароль – введите текущий – 1, затем введите свой новый пароль (при вводе символы не отображаются!) и информацию о себе.

```
admin1@ubuntuL01:~$ sudo adduser tkachuk
Добавляется пользователь «tkachuk» ...
Добавляется новая группа «tkachuk» (1001) ...
Добавляется новый пользователь «tkachuk» (1001) в группу «tkachuk» ...
Создаётся домашний каталог «/home/tkachuk» ...
Копирование файлов из «/etc/skel» ...
Введите новый пароль UNIX:
Повторите ввод нового пароля UNIX:
passwd: password updated successfully
Changing the user information for tkachuk
Enter the new value, or press ENTER for the default
        Full Name []: Евгений Ткачук
        Room Number []: 205
        Work Phone []:
        Home Phone []:
        Other []:
chfn: name with non-ASCII characters: 'Евгений Ткачук'
Данная информация корректна? [Y/n] Y
admin1@ubuntuL01:~$ _
```

5. Для того, что вы могли выполнять команды администратора, введите себя в группу sudo командой usermod -aG sudo <*ваша фамилия*>

```
admin1@ubuntuL01:~$ sudo usermod -aG sudo tkachuk
admin1@ubuntuL01:~$
```

6. Сохраните скриншот выполненных команд и включите его в отчет. Для того, чтобы сделать скриншот, надо выйти из виртуальной машины нажатием правой кнопки Shift.
7. Выполните перезагрузку системы командой sudo shutdown -r now
8. Зайдите систему по своим именем, выполните все команды, описанные в теоретическом материале, скриншоты включите в отчёт.

9. В отчете в свободной форме опишите свои действия, сопроводив их скриншотами.

7. Практическое занятие № 4 Мониторинг процессов ОС UBUNTU (Linux)

7.1.Теоретические сведения

Для просмотра запущенных процессов в **Ubuntu** Linux при помощи терминала, необходимо набрать в нем следующие команды:

top – команда выдачи данных об активности процессов в Ubuntu

Программа top динамически выдает в режиме реального времени информации о работающей системе, показывает запущенные процессы и потребление ими ресурсов системы. По умолчанию выдает задачи, наиболее загружающие процессор сервера, и обновляет список каждые пять секунд.
При выполнении top в верхней части окна отображается астрономическое время, время, прошедшее с момента запуска системы, число пользователей в системе, число запущенных процессов и число процессов, находящихся в разных состояниях, данные об использовании ЦПУ, памяти и свопа.
Далее идет таблица, характеризующая отдельные процессы. Число строк, отображаемых в этой таблице, определяется размером окна: сколько строк помещается, столько и выводится. Список процессов может быть отсортирован по используемому времени ЦПУ (по умолчанию), по использованию памяти, по PID, по времени исполнения. Переключать режимы отображения можно с помощью команд, которые программа **top** воспринимает. Это следующие команды (просто нажимайте соответствующие клавиши, только с учетом регистра, то есть вместе с клавишей Shift):

Shift+N — сортировка по PID;

Shift+A — сортировать процессы по возрасту;

Shift+P — сортировать процессы по использованию ЦПУ;

Shift+M — сортировать процессы по использованию памяти;

Shift+T — сортировка по времени выполнения.

Кроме команд, определяющих режим сортировки, команда **top** воспринимает еще ряд команд, которые позволяют управлять процессами в интерактивном режиме. С

помощью команды можно завершить некоторый процесс (его PID будет запрошен), а с помощью команды можно переопределить значение nice для некоторого процесса. Таким образом, эти две команды аналогичны командам **kill** и **renice**.

Команду том можно использовать со следующими параметрами:

t – Включение и выключение выдачи на экран суммарных данных.

m – Включение и выключение выдачи на экран информации об использовании памяти.

A – Сортировка строк по максимальному потреблению различных системных ресурсов. Полезна для быстрой идентификации задач, для которых в системе не хватает ресурсов.

f – Вход в меню интерактивного конфигурирования данных, выдаваемых на экран командой top. Полезна для настройки команды top для выполнения специфической задачи.

o – Позволяет вам интерактивно задавать порядок строк, выдаваемой командой top.

r – Изменение приоритета процессов с помощью команды renice.

k – Удаление процесса с помощью команды kill.

z – Переключение между цветным / монохромным вариантом выдачи изображения.

q – Завершение работы, выход из программы.

Использование Top Command - Task Manager для Ubuntu

1- Top Command

2- Просмотр списка команд – аналог help

3- Добавить убрать Field

4- Поменять порядок Field

5- Распорядок в Field

6- Сортировать по CPU

7- Отобразить процессы определенные пользователем

8- Выделить запущенные процессы

9- Смотреть абсолютный путь процессов

10- Убить процесс

1- Top Command

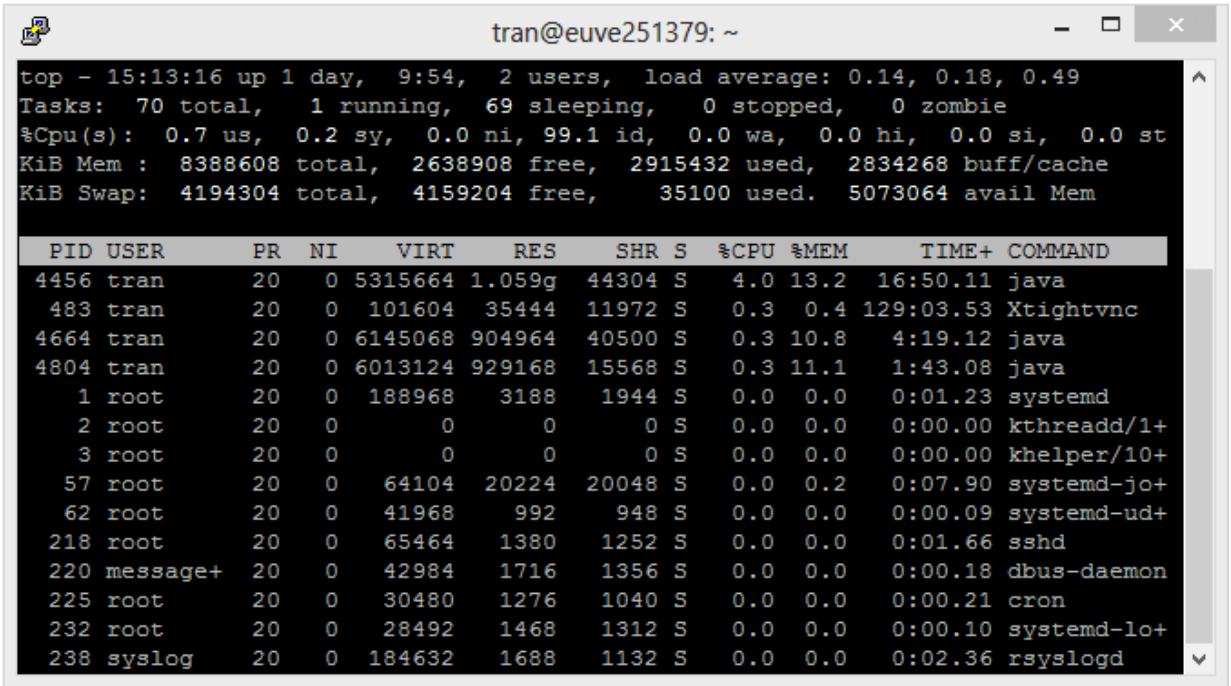
Вы уже знакомы с **Task Manager** (управление задачей) в операционной системе **Windows**, вы зададите вопрос, есть ли что-то похожее с **Task Manager** в **Ubuntu Server** или нет

В **Windows,** Task Manager это полезная программа, которая позволяет узнать, какие приложения работают в системе и количество ресурсов использующееся этими приложениями

Ubuntu обеспечивает вас подобной программой, но интерфейс этого приложения очень рудиментарный. Это приложение называется " **Top Command**"

Для запуска " **Top Command**" запустить команду ***top:***

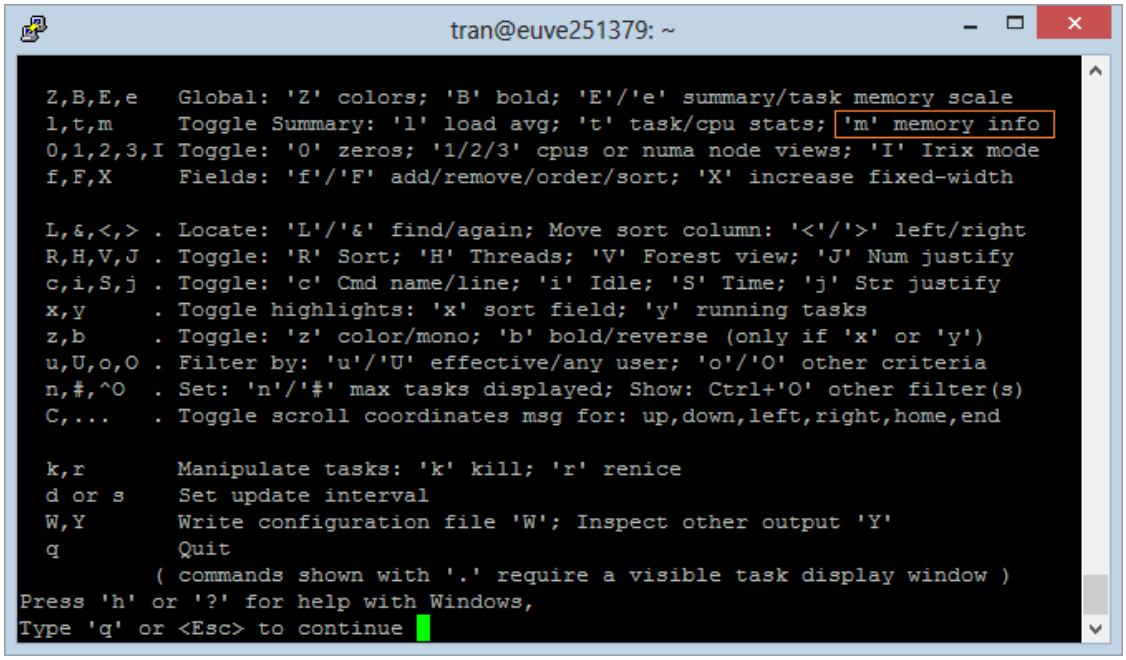
Top Command:



*Примечание: Для того, чтобы выйти из " **Top Command**», нажимате кнопку '**q**'.*

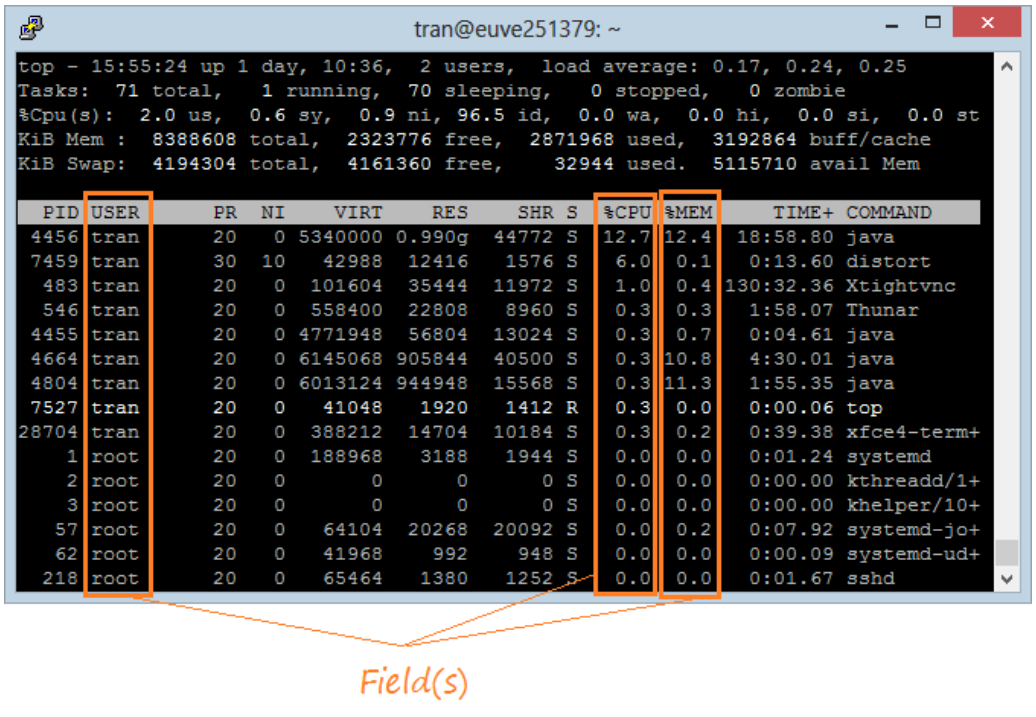
2- Просмотр списка команд – аналог help

Нажмите на ' **h**', вы можете просмотреть список команд и инструкцию, и нажмите "ESC", чтобы вернуться к главному экрану

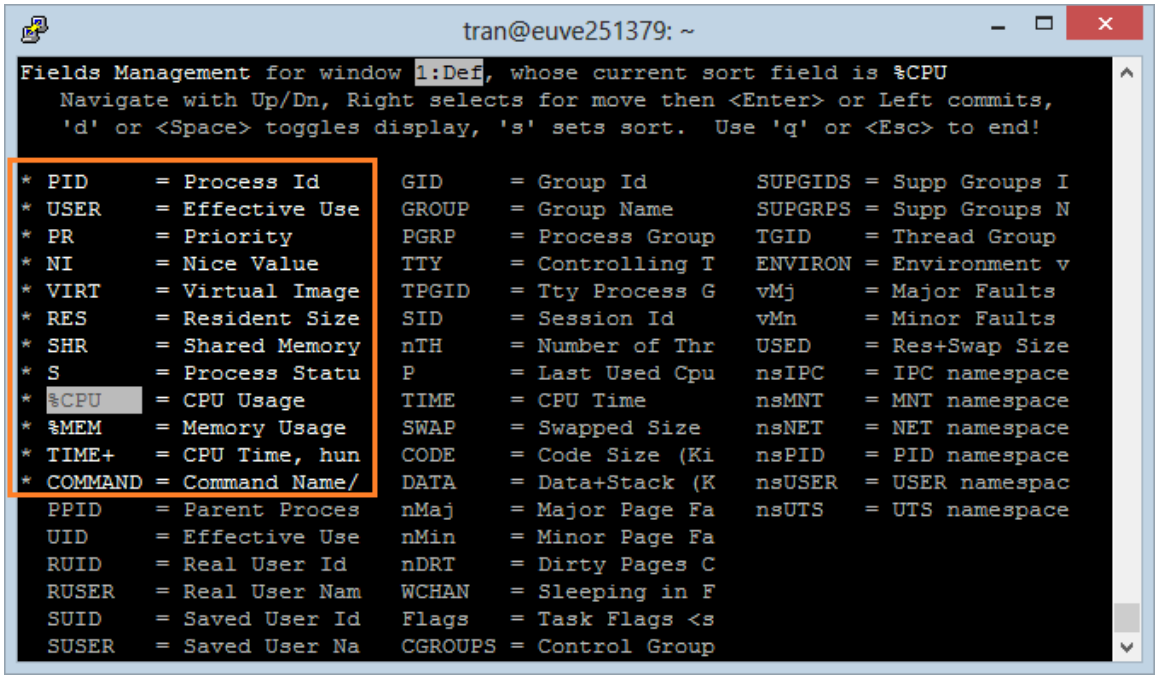


3- Добавить убрать Field

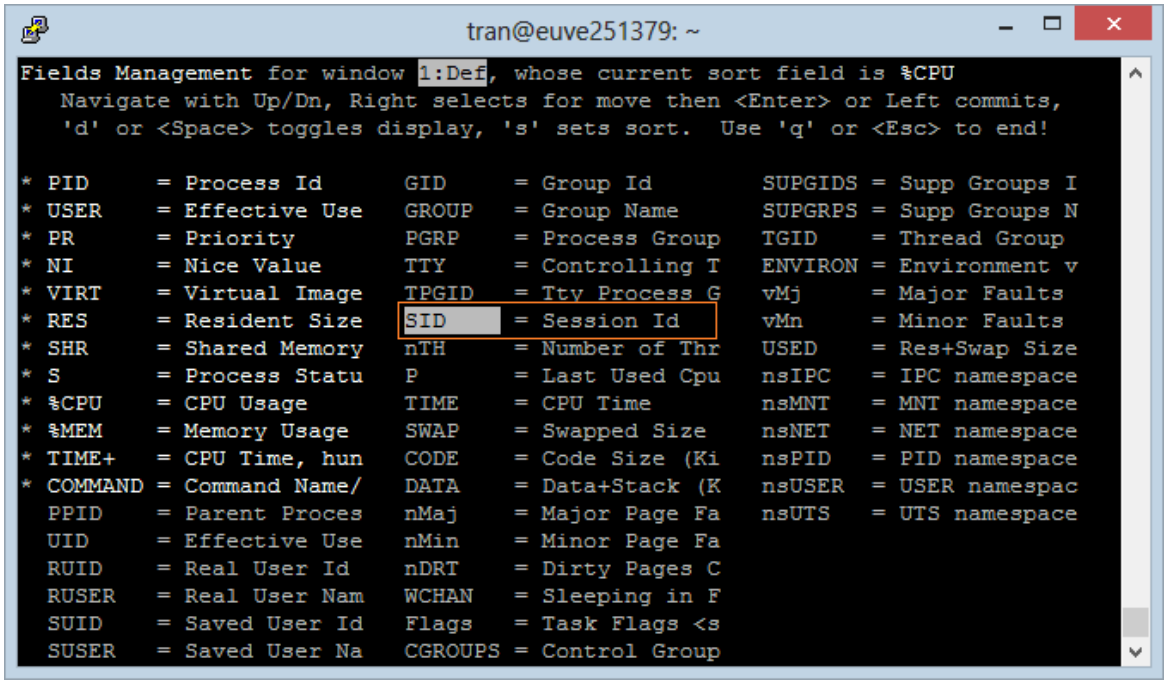
Есть много Field (Поле информации) которые могут быть вам интересны. По умолчанию " **Top Command**" только отобразит некоторые Field . Нажмите '**f**', чтобы добавить или удалить Field



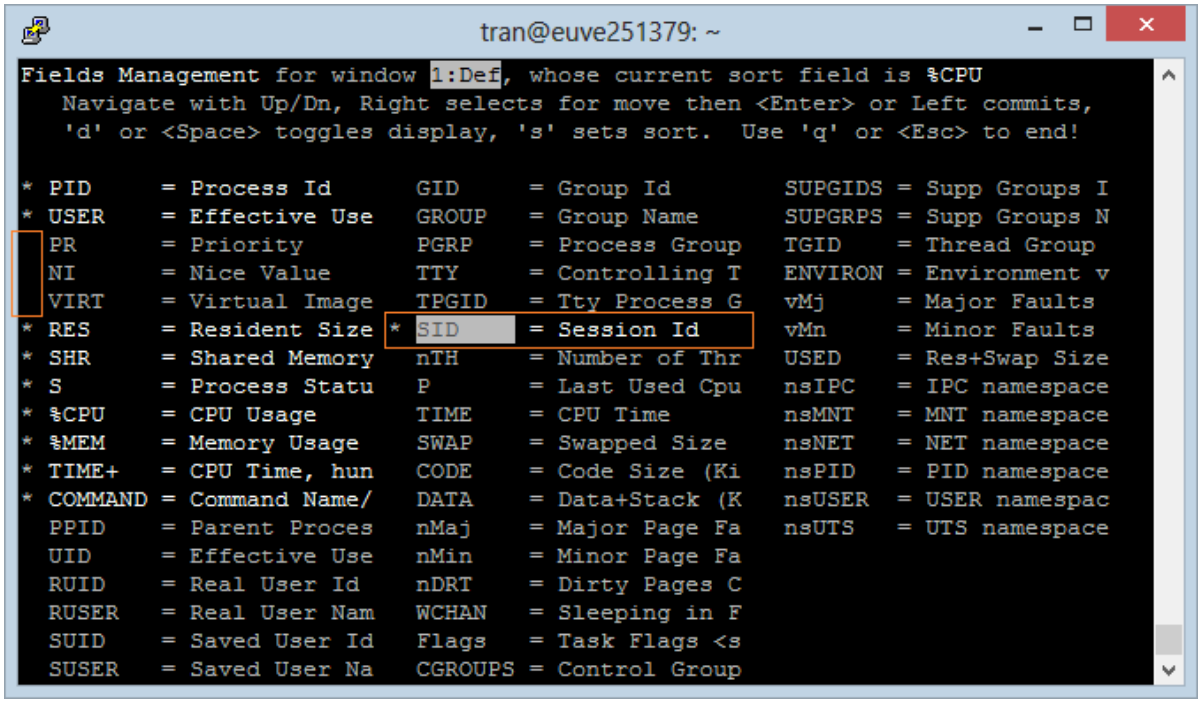
Field со звездочкой (*) это Field, которые отображены, другие Field не отображаются на **"Top Command"**



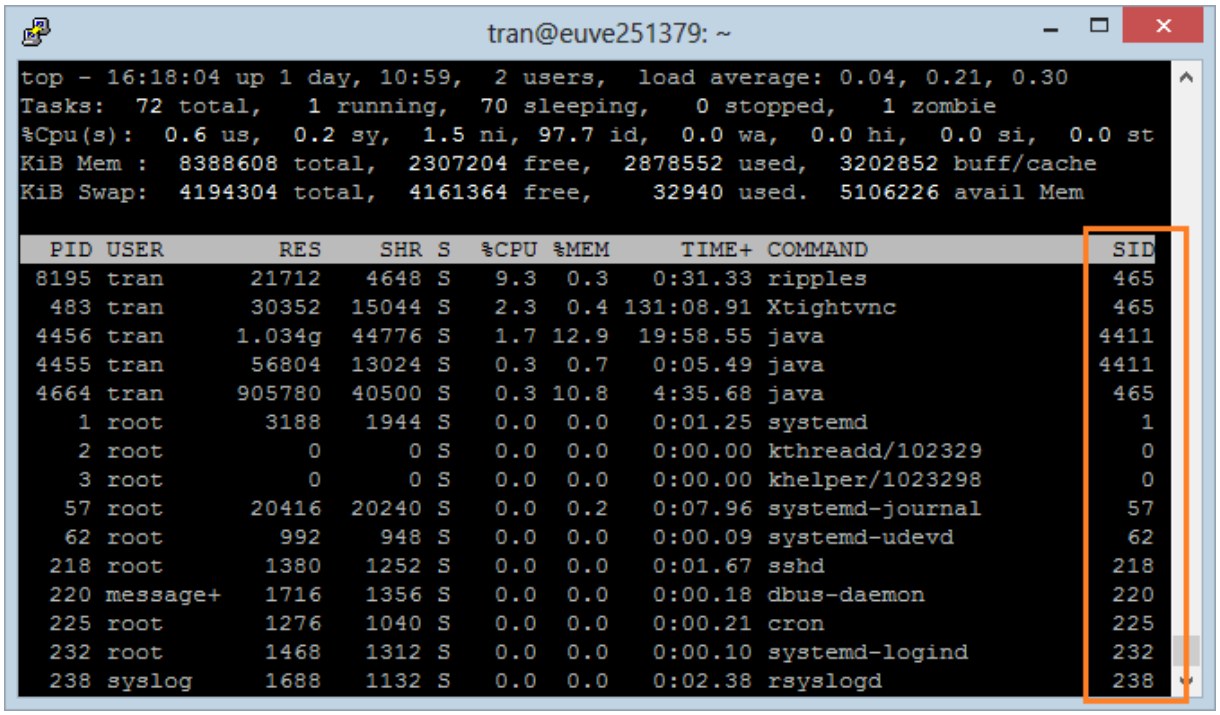
Используйте клавишу со стрелками вверх или вниз, чтобы перейти к нужному **Field**



1. Нажмите на кнопку ENTER.
2. Нажмите пробел (Whitespace), чтобы выбрать или отменить выбор Field.

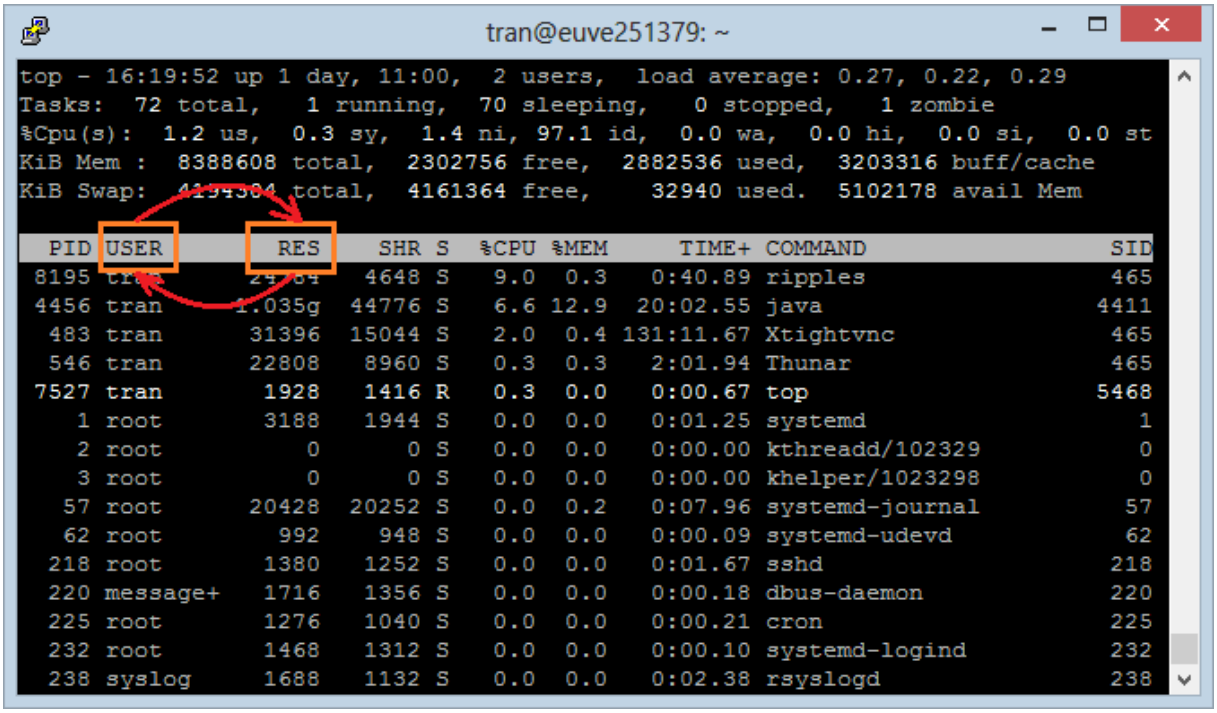


Нажмите на " **ESC**" чтобы вернуться к главному экрану

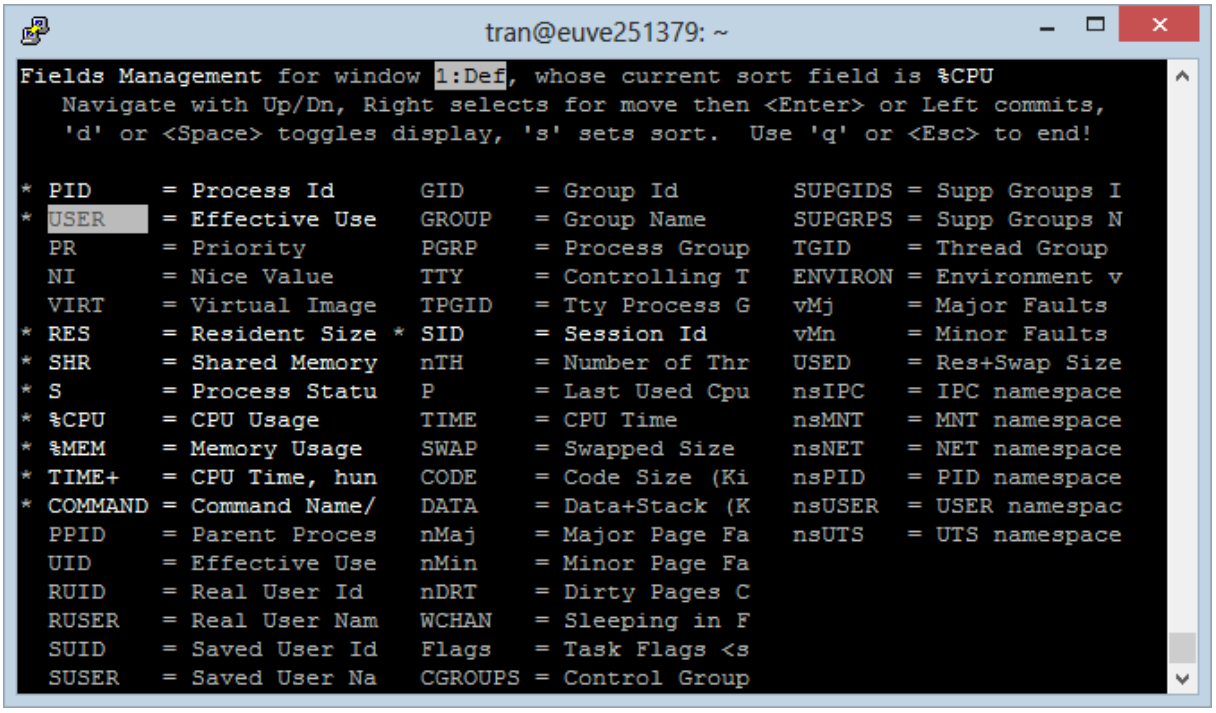


4- Поменять порядок Field

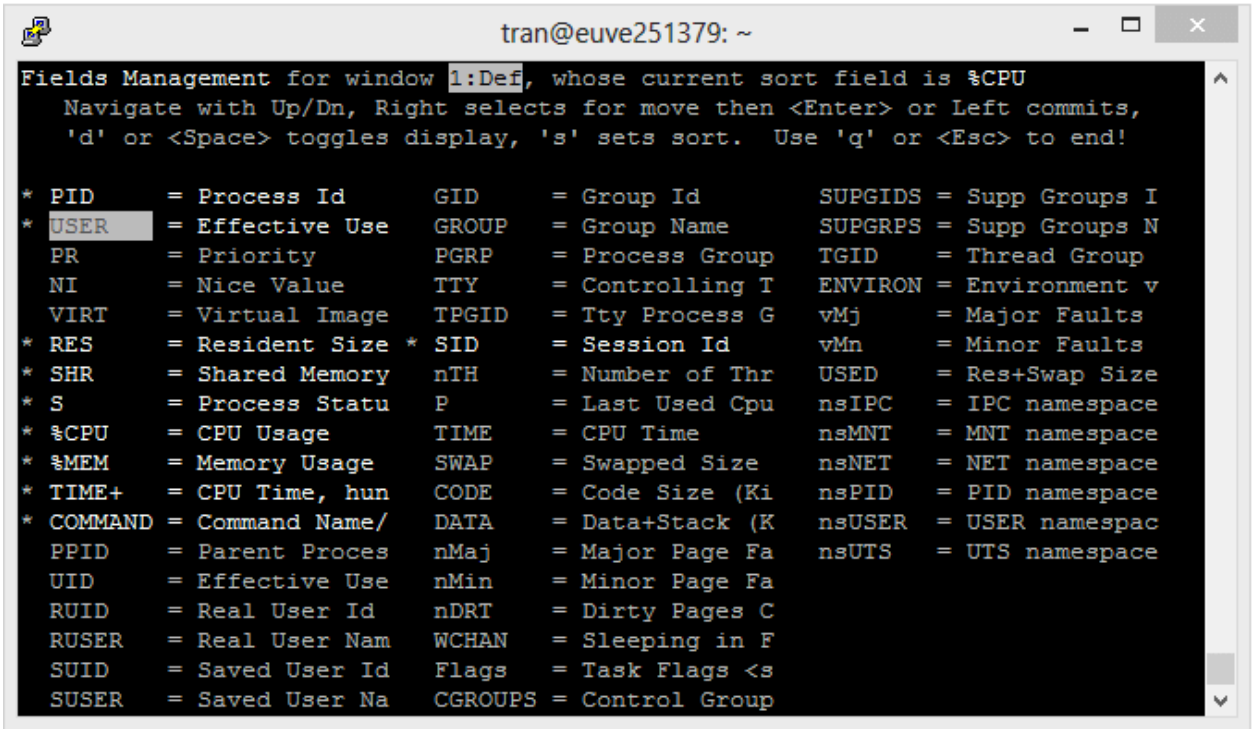
Для того, чтобы улучшить вид, вы можете изменить порядок Field на экране " **Top Command**"



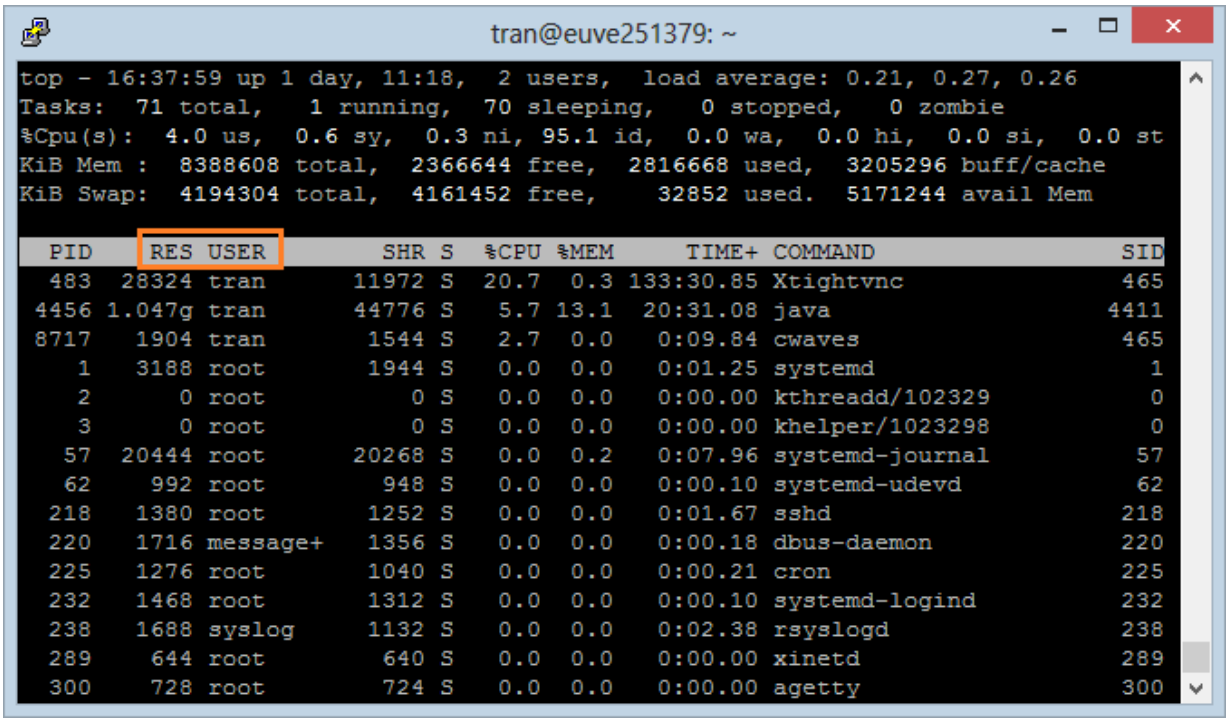
Нажмите ' **F**', чтобы просмотреть список Field.



1. Нажмите клавишу со стрелкой направо, чтобы выделить Field для перемещения.
2. Нажмите стрелку вверх / вниз для перемещения Field (обозначение выше) в другое место.
3. Нажмите кнопку ENTER для завершения.

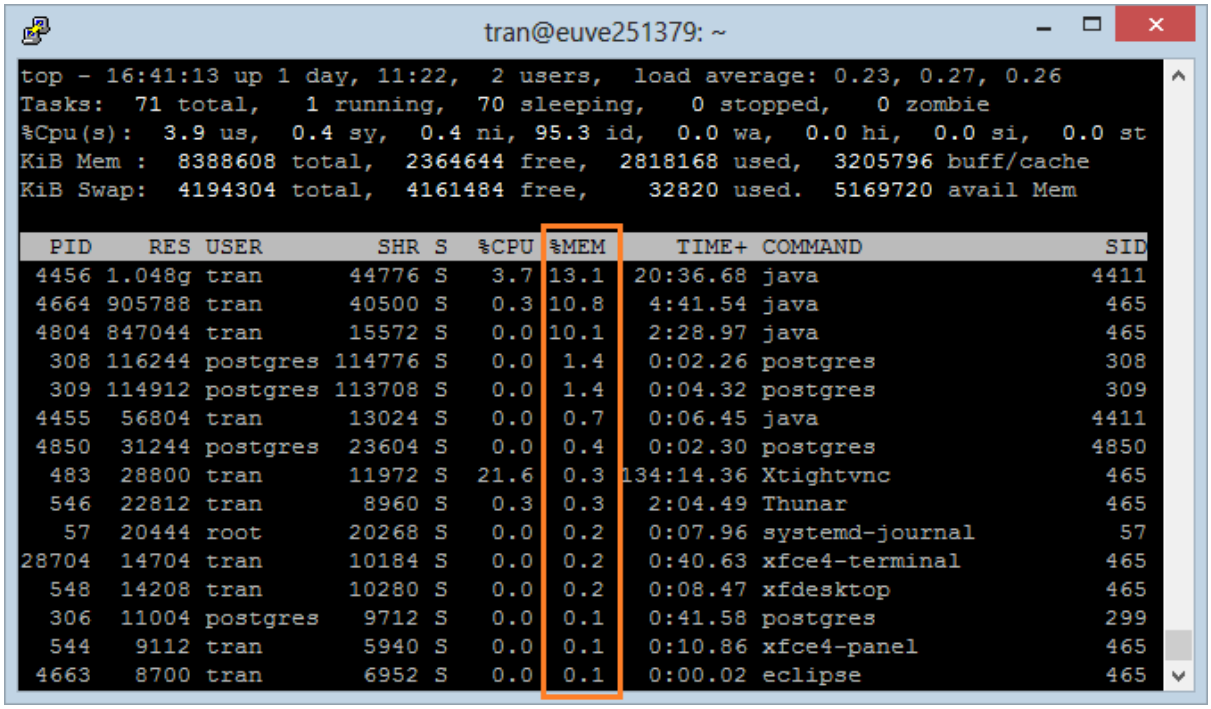


Нажмите " **ESC**" чтобы вернуться к главному экрану



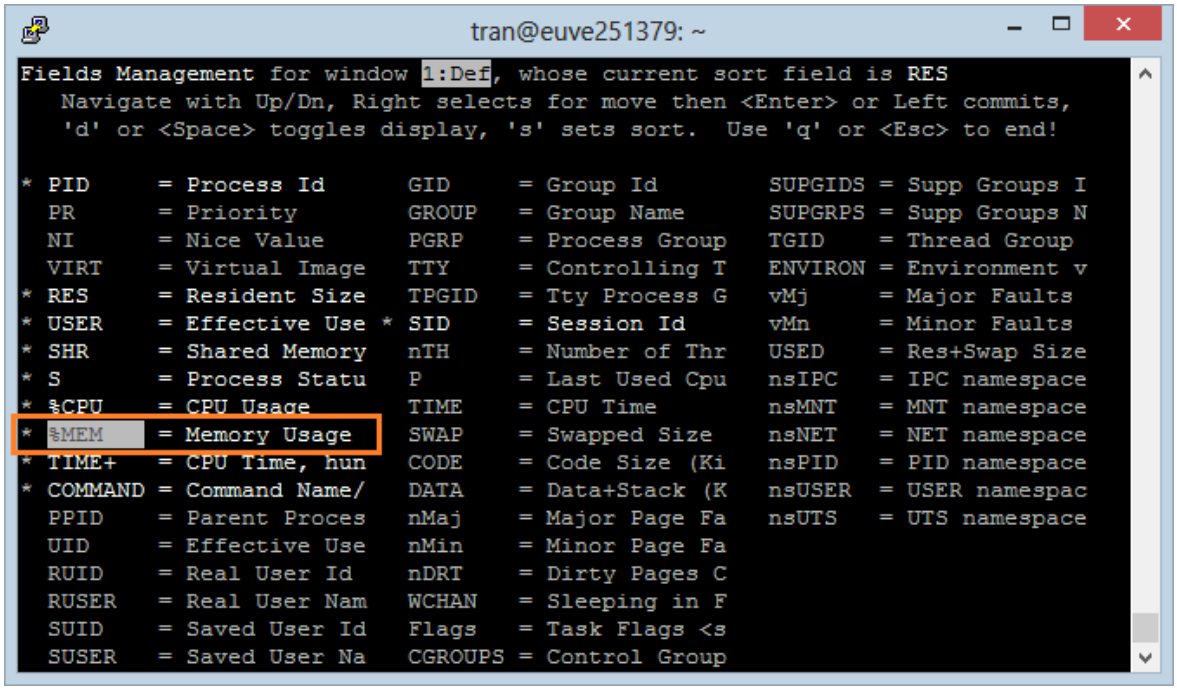
5- Сортировка в Field

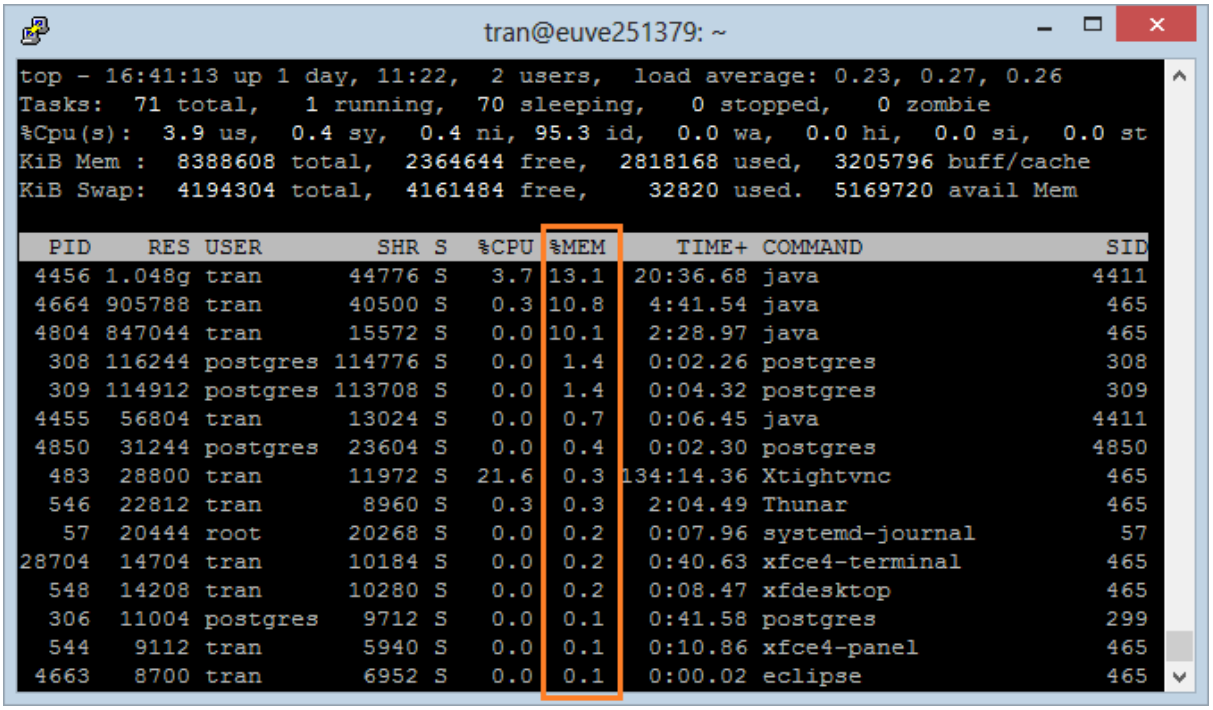
Предположим, вы хотите знать, какие программы используют наибольший объем памяти, вам нужно сортировать по **% MEM (% Memory)**.



Нажмите ' **F**', чтобы просмотреть список Field.

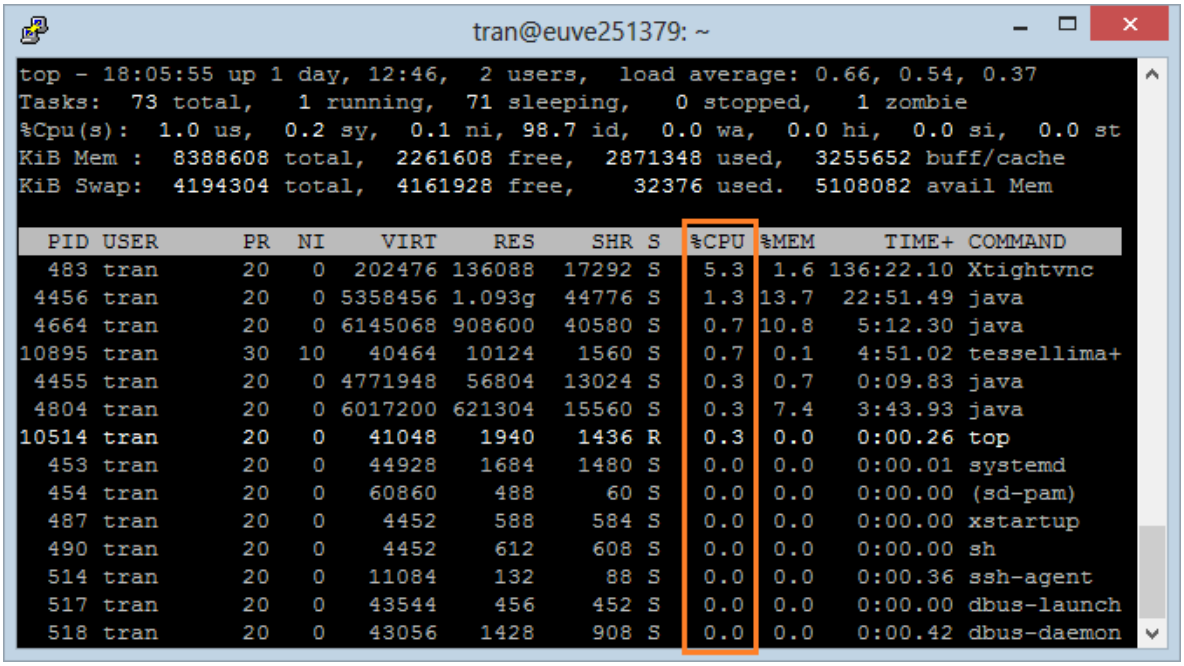
1. Выберите Field для сортировки.
2. Нажмите 'S'.
3. Нажмите "ESC", чтобы вернуться к главному экрану.





6- Сортировать по CPU

CPU (field) имеет важное значение, он говорит вам какой процесс (Process) заставляет компьютер обрабатывать больше всего на данный момент. Нажмите " **Shift + P** 'для сортировки по CPU.



7- Отобразить процессы определенные пользователем

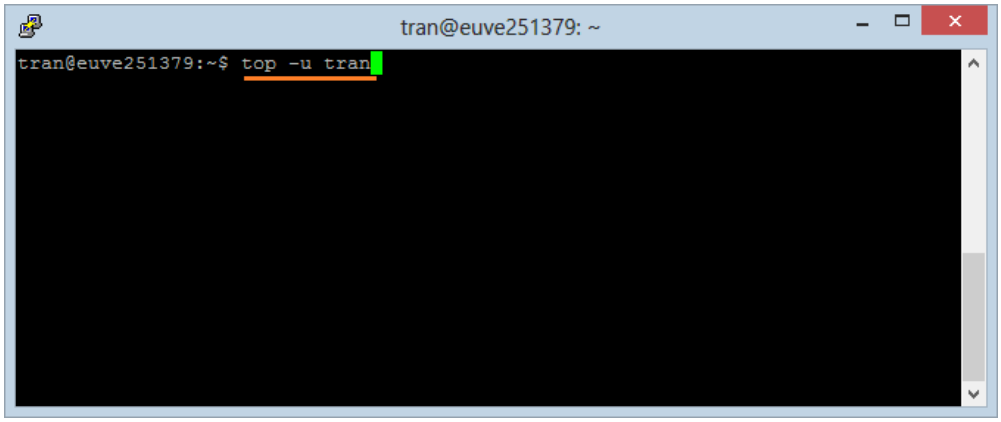
Используйте опцию **'-u',** который позволяет указать пользователя, и увидеть какие процессы выполняются этим пользователем.

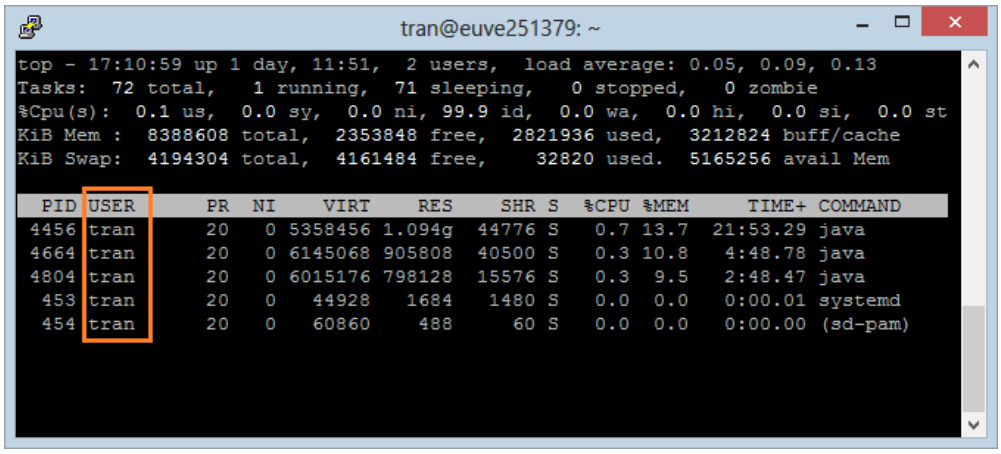
?

top -u

user_name

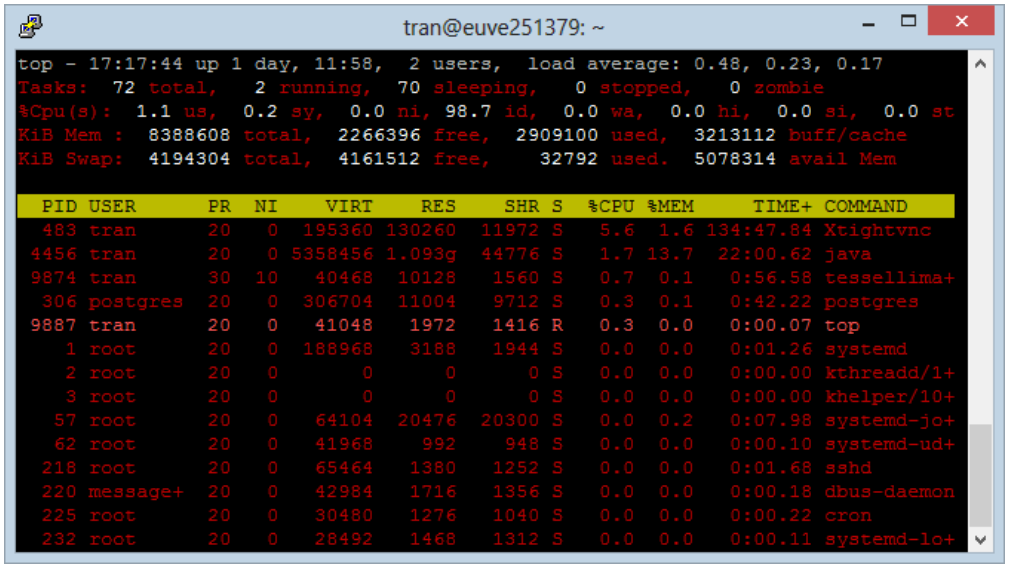
Например, смотреть процессы (process) выполняемые пользователем 'Тран'





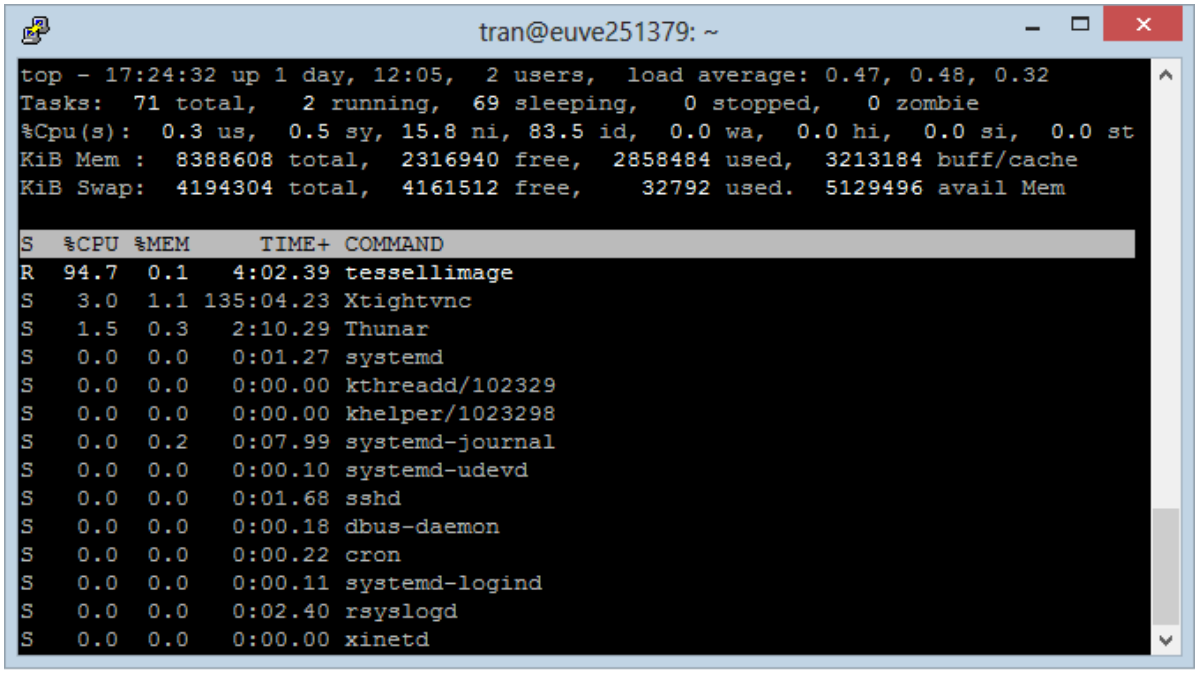
8- Выделить запущенные процессы

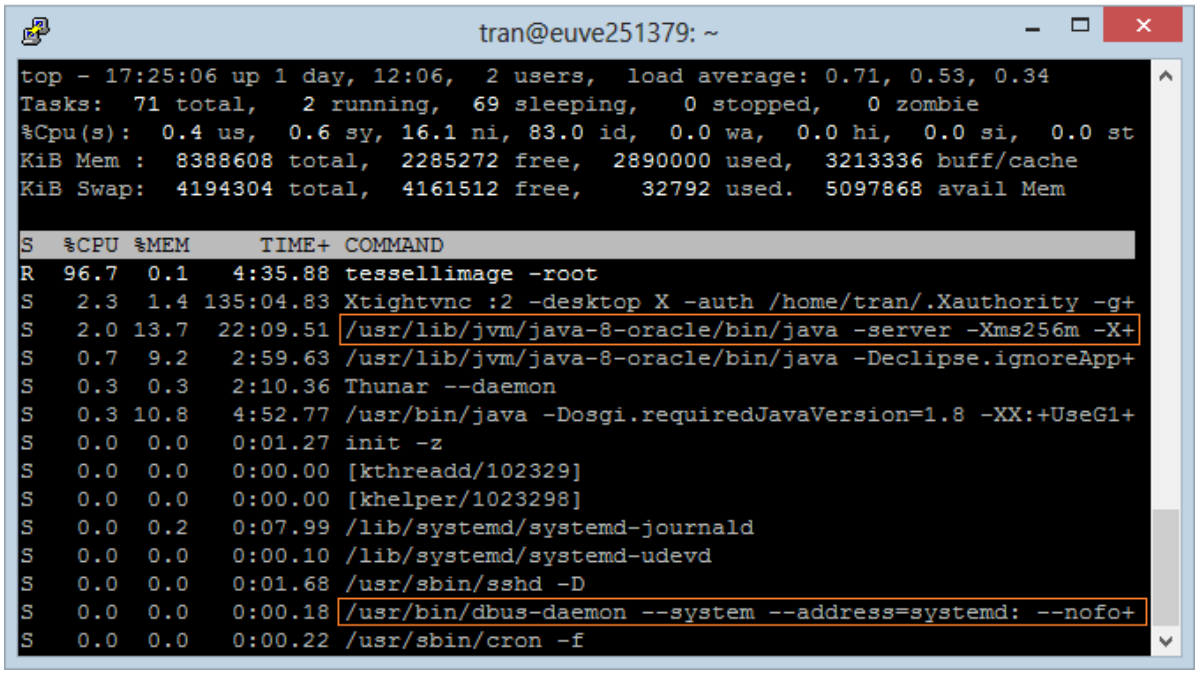
Нажмите 'Z', программа " **Top Command**" покажет процессы, запущенные с цветами. Это позволяет легко определить запущенные процессы.



9- Смотреть абсолютный путь процессов

При работе процессов, если вы хотите, чтобы увидеть абсолютный путь файла, выполняющего этот процесс? Просто нажмите **'c'**





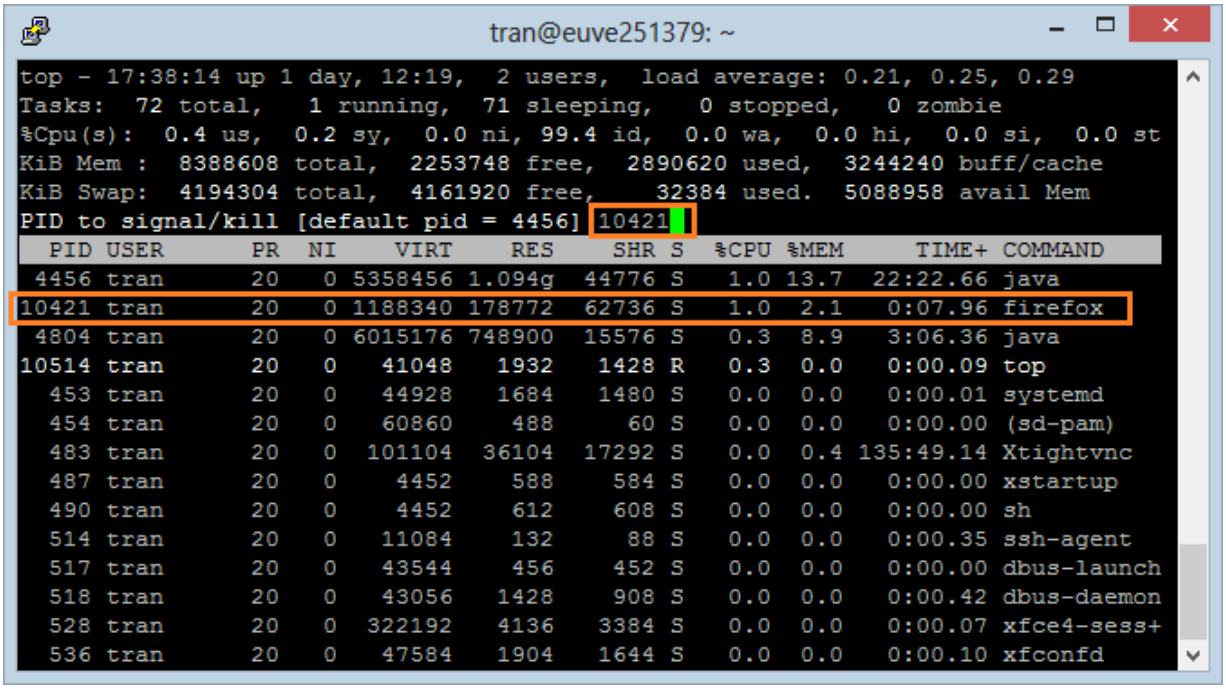
10- Убить процесс

В **Windows** вы можете закончить задание которое выполняется, используя " **End Task**".

В **Ubuntu** вы можете использовать " **Kill**", чтобы убить процесс

1. Нажмите 'к'.
2. Введите PID (Process ID).

3. Нажмите кнопку ENTER



Другие команды мониторинга процессов Ubuntu

ps – команда выдачи списка процессов Ubuntu

Команда **ps** выдаст краткий список текущих процессов. Вывод команды **ps** схож с выводом команды **top**, однако он отображает статический снимок процессов. Для того, чтобы выбрать все процессы, используете параметр **- A** или **- e**

Вывод большего количества данных по процессам

ps -Al

Для того, чтобы включить выдачу всех данных (будут показаны аргументы командной строки, переданные в процесс):

ps -AlF

Вывод списка всех процессов Ubuntu

ps ax

ps axu

Отображение потоков (LWP и NLWP)

ps -AlFH

Вывод информации о параметрах безопасности Ubuntu

ps -eo euser,ruser,suser,**fuser**,f,**comm**,label
ps axZ
ps -eM

Вывод дерева процессов

ps -ejH
ps axjf
pstree

Отображение потоков после процессов

ps -AlLm

Настраиваемая выдача данных

Позволяет выводить данные в последовательности, определяемой пользователем

ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,**stat**,wchan:14,**comm**
ps axo **stat**,euid,ruid,tty,tpgid,sess,pgrp,ppid,pid,pcpu,**comm**
ps -eopid,tt,user,fname,tmout,f,wchan

Вывод процессов, запущенных пользователем User

ps -U User -u User u

Вывод ID процессов, запущенных под apache

ps -C apache -o pid=

или

pgrep apache

Вывод имени для PID 30470

ps -p 30470 -o **comm**=

Вывод 10 процессов, потребляющих наибольшее количество памяти

ps -auxf | **sort** -nr -k 4 | **head** -10

Вывод 10 процессов, потребляющих наибольший ресурс процессора

ps -auxf | **sort** -nr -k 3 | **head** -10

По умолчанию, команда ps выводит только информацию о процессах, запущенных в текущей сессии терминала bash. Для вывода информации по всем процессам необходимо ввести команду **ps** с параметром **-e**. Для отображения желаемых полей необходимо ввести команду ps с параметром -o поле1,поле2,…, где через запятую перечисляются поля, которые необходимо отобразить.

free – использование памяти

Команда free показывает общее количество свободной и используемой системой физической памяти и памяти свопинга, а также размеры буферов, используемые ядром.

free

uptime – сообщает, как долго работает система

Команду uptime можно использовать с тем, чтобы определить, как долго работает сервер. Выдаются: текущее время, сколько времени работает система,

сколько в текущий момент зарегистрировано пользователей и какова средняя нагрузка на систему в последние 1, 5 и 15 минут.

uptime

w – определяем, кто зарегистрирован и что они делают

Команда w выдает информацию о том, какие пользователи сейчас находятся в системе и какие процессы запущены от их имени.

w username

w User

pmap – использование процессами оперативной памяти

Команда pmap выдает данные о распределении памяти между процессами. Использование этой команды позволит найти причину узких мест, связанных с использованием памяти.

pmap -d PID

Для того, чтобы получить информацию об использовании памяти процессом с pid # 26321, введите:

pmap -d 26321

vmstat – активность системы, информация о системе и аппаратных ресурсах

Команда vmstat выдает информационный отчет об активности процессов, памяти, свопинга, поблочного ввода/вывода, прерываний и процессора.

vmstat 3

Выдача статистики использования памяти

vmstat -m

Получение данных об активности / неактивности страниц памяти

vmstat –a

mpstat – использование мультипроцессора

Команда mpstat выводит данные об активности каждого имеющегося в наличие процессора, процессор 0 будет первым. Команда mpstat -P ALL выводит данные о среднем использовании ресурсов для каждого из процессоров:

mpstat -P ALL

iostat – средняя загрузка процессора, активность дисков

Команда iostat выдает статистику использования процессора, а также статистику ввода/вывода для устройств, разделов и сетевых файловых систем (NFS).

iostat

Задание:

1. Выполнить импорт конфигурации виртуальной машины – сервера UbuntuServer01 в Oracle VM VirtualBox
2. Пока идёт импорт (около 15 минут) изучить теоретический материал. Сомостоятельно выполнить поиск статей в Интернет, посвящённых Ubuntu Server.
3. Запустить виртуальную машину UbuntuServer01. Войти в терминал с логином admin1 и паролем 1.
4. Добавьте себя в список пользователей системы командой sudo adduser <*ваша фамилия*>, (латиницей) ,система потребует пароль – введите текущий – 1, затем введите свой новый пароль (при вводе символы не

отображаются!) и информацию о себе.

```
admin1@ubuntuL01:~$ sudo adduser tkachuk
Добавляется пользователь «tkachuk» ...
Добавляется новая группа «tkachuk» (1001) ...
Добавляется новый пользователь «tkachuk» (1001) в группу «tkachuk» ...
Создаётся домашний каталог «/home/tkachuk» ...
Копирование файлов из «/etc/skel» ...
Введите новый пароль UNIX:
Повторите ввод нового пароля UNIX:
passwd: password updated successfully
Changing the user information for tkachuk
Enter the new value, or press ENTER for the default
        Full Name []: Евгений Ткачук
        Room Number []: 205
        Work Phone []:
        Home Phone []:
        Other []:
chfn: name with non-ASCII characters: 'Евгений Ткачук'
Данная информация корректна? [Y/n] Y
admin1@ubuntuL01:~$ _
```

5. Для того, что вы могли выполнять команды администратора, введите себя в группу sudo командой usermod -aG sudo <*ваша фамилия*>

```
admin1@ubuntuL01:~$ sudo usermod -aG sudo tkachuk
admin1@ubuntuL01:~$
```

6. Сохраните скриншот выполненных команд и включите его в отчет. Для того, чтобы сделать скриншот, надо выйти из виртуальной машины нажатием правой кнопки Shift.
7. Выполните перезагрузку системы командой sudo shutdown -r now
8. Зайдите систему по своим именем, выполните все команды, описанные в теоретическом материале, скриншоты включите в отчёт.
9. В отчете в свободной форме опишите свои действия, сопроводив их скриншотами.

8. Лабораторная работа № 4 Управление пользователями в ОС UBUNTU

Цель работы:
Изучить возможности Linux при работе с пользователями и управлении правами доступа.

Задачи работы:

- Рассмотреть концепцию Linux при работе с пользователями;
- Изучить управление базами данных пользователей;
- Рассмотреть возможности манипулирования доступом к данным.

8.1.Порядок выполнения работы

- Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
- Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- Выполните все действия, описанные в теоретической части (для тренировки).
- Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.
- Экспортируйте образ сервера на свой носитель или в своё облако для последующей работы.

8.2. Теоретическая часть

Работа с пользователями

Linux - это многопользовательская операционная система. Каждый пользователь в Linux принадлежит одной основной группе и одной или нескольким дополнительным группам. В Linux, как и в большинстве других операционных системах, работа с пользователями заключается в наборе следующих манипуляций: добавление пользователя/группы, удаление пользователя/группы, модификация настроек пользователя/группы. Данные манипуляции производятся с помощью команд: useradd, groupadd, userdel, groupdel, usermod, groupmod, а так же passwd, gpasswd, id. Существуют так же и графические средства администрирования пользователями, обычно они расположены в оболочке X в разделе Администрирование - Пользователи и группы. Однако, при администрировании Linux использование графических оболочек не приветствуется.

UID, GID

Каждый пользователь в системе имеет свой уникальный идентификационный номер (user-ID, или UID). Также пользователи могут объединяться в группы, которые в свою очередь имеют group-ID, или GID. Чтобы узнать свой UID и GID, т.е. уникальный номер пользователя и номер группы, к которой вы принадлежите, необходимо ввести команду id (Рис. 8.1).

```
work@work:~$ id
uid=1000(work) gid=1000(work) группы=4(adm)
,20(dialout),24(cdrom),46(plugdev),105(lpad
min),119(admin),122(sambashare),1000(work)
```

Рис. 8.1. Отобразить UID и GID

Пример добавления пользователя (Рис. 8.1.):

```
work@work:~$ sudo groupadd test
work@work:~$ sudo useradd -c "Test Test" -g test -m test
work@work:~$ sudo passwd test
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
work@work:~$ sudo it test
sudo: it: command not found
work@work:~$ sudo id test
uid=1001(test) gid=1002(test) groups=1002(test)
work@work:~$ sudo ls -ld /home/test/
drwxr-xr-x 2 test test 4096 2011-12-17 15:13 /home/test/
```

Рис. 8.2. Добавление нового пользователя

В примере мы добавляем группу для нового пользователя (groupadd), далее создаем нового пользователя с полным именем Test Test, имеющего основную группу test и логин test, далее задаем пароль для пользователя test (passwd test) и проверяем параметры созданного пользователя (id и созданный каталог пользователя /home/test/). На Рис. 8.2 видно, что UID и GID - более 1000. Данная особенность является признаком обычного пользователя. Значения ниже (меньше) 1000 (а в некоторых версиях - меньше 500) указывают на то, что пользователь является системным пользователем.

В соответствии с соглашением, системные пользователи обычно имеют id меньше, чем 100, а пользователь root имеет id, равный 0. Автоматическая нумерация обычных пользователей начинается со значения UID_MIN, установленного в файле /etc/login.defs. Это значение обычно установлено в 500 или 1000.

Помимо учетных записей обычных пользователей и учетной записи пользователя root, в системе бывает несколько учетных записей специального назначения для демонов, таких как FTP, SSH, mail, news и т.д. Такие учетные записи часто управляют файлами, но к ним невозможно получить доступ путем обычной регистрации в системе. Поэтому обычно они имеют login shell, определенный как /sbin/nologin или /bin/false, чтобы попытки зарегистрироваться в системе терпели неудачу.

Управление базами данных пользователей и групп в Linux

Основные файлы, содержащие информацию о пользователях и группах, - это четыре файла в каталоге */etc*.

1. /etc/passwd - файл паролей, содержащий основную информацию о пользователях;
2. /etc/shadow - файл теневых шифрованных паролей, содержащий зашифрованные пароли;
3. /etc/group - файл групп, содержащий основную информацию о группах и принадлежащих этим группам пользователях;
4. /etc/gshadow - файл теневых групп, содержащий шифрованные пароли групп.

Данные файлы редактировать обычным текстовым редактором крайне не рекомендуется. Они, обновляются при выполнении вышеуказанных команд, при этом при изменении - блокируются и синхронизируются.

Если все же есть острая необходимость в редактировании указанный файлов, то при помощи команды vipw можно безопасно редактировать файл /etc/passwd, а при помощи команды vigr безопасно редактировать файл /etc/group. Эти команды заблокируют необходимые файлы на то время, пока при помощи редактора vi будут производиться изменения. Если вы вносите изменения в файл /etc/passwd, команда vipw подскажет, что необходимо проверить, не нужно ли обновить и файл /etc/shadow. Подобным образом, если вы обновляете файл /etc/group при помощи команды vigr, вы получите подсказку, что необходимо обновить и файл /etc/gshadow. Если необходимо удалить администраторов группы, необходимо использовать команду vigr, поскольку команда gpasswd позволяет только добавлять администраторов.

В современных системах, файлы passwd и group не хранят пароли в открытом виде. Это сделано из соображений безопасности. Сами файлы passwd и group должны быть доступными для чтения для всех, а зашифрованные пароли - недоступными для чтения для всех. Поэтому зашифрованные пароли хранятся в теневых файлах, и эти файлы доступны для чтения только пользователю root. Необходимый доступ для изменения аутентификационных данных обеспечивается при помощи suid-программы, которая имеет полномочия пользователя root, но может быть запущена любым пользователем.

Права доступа

Операционная система Linux - это многопользовательская система, которая дает огромные возможности манипулирования доступом к данным для каждого пользователя отдельно. Это позволяет гибко регулировать отношения между пользователями, объединяя их в группы, что позволит защитить данные одного пользователя от нежелательного вмешательства других.

Бессмысленно считать, что файловая система это не самая важная часть операционной системы, поскольку все данные пользователей хранятся именно в файлах.

В UNIX-подобных системах файлы также обеспечивают доступ к периферийным устройствам, дисковым накопителям, принтерам и т.п.

Права доступа к файлам

В свою очередь файлы имеют двух владельцев: пользователя (user owner) и группу пользователей (group owner). Для каждого файла есть индивидуальные права доступа, которые разбиты на три группы:

- Доступ для пользователя-владельца файла (owner).
- Доступ для группы-владельца файла (group).
- Доступ для остальных пользователей (others).

Для каждой категории устанавливаются три вида доступа: (x) - право на запуск файла, (r) - право на чтение файла, (w) - право на изменение (редактирование) файла.

Для того, чтобы увидеть права доступа к файлам необходимо ввести команду ls с ключом –l (Рис. 8.3).

```
work@work:~$ ls -l Картинки/Kubuntu_leaflet.jpg
-rw-r--r-- 1 work work 658825 2010-03-26 15:21
Картинки/Kubuntu_leaflet.jpg
```

Рис. 8.3. Просмотр прав доступа к файлу

Для данного примера мы видим, что владелец имеет права на чтение, запись (первые две буквы rw), группа пользователей может лишь читать этот файл (следующая r--), остальные пользователи могут также только читать данный файл (r--).

Изменение прав доступа

Права пользователя могут быть изменены только владельцем файла или пользователем с правами администратора системы. Для изменения прав используется команда:

chmod[u|g|o|a] [+|-|=] [r|w|x] name1 [name2 ...]

В качестве аргументов команда принимает указание классов доступа («u» - владелец-пользователь, «g» - владелец-группа, «o» - остальные пользователи, «a» - все вышеперечисленные группы вместе), права доступа («r» - чтение, «w» - запись, «x» - выполнение) и операцию, которую необходимо произвести («+» - добавить, «-» -убрать, «=» - присвоить).

Таким образом, чтобы разрешить выполнение файла ip, который находиться в директории /home/work/Загрузки всем пользователем необходимо выполнить команду (Рис. 8.4):

```
work@work:~$ chmod a+x Загрузки/ip
```

Рис. 8.4. Команда, выдающая права на исполнение файла

Далее, чтобы оставить права записи только для владельца файла необходимо выполнить (рис.5.5):

```
work@work:~$ chmod go-w Загрузки/ip
```

Рис. 8.5. Команда, повзоляющая оставить права записи только для владельца файла

Рассмотрим еще несколько примеров:

- chmod go=w ip - установить право на запись для всех пользователей кроме владельца;
- chmod a+x ip - предоставить право на запись для всех пользователей;
- chmod g+x-w ip - добавить для группы право на выполнения файла, но снять право на запись.

Права доступа можно представить в виде битовой строки, в которой каждые 3 бита определяют права доступа для соответствующей категории пользователей, как представлено в таблице 5.1:

Таблица 5.1

Представление прав доступа в виде битовой строки

rwx	rwx	rwx
421	421	421
user	group	others
владелец	группа	остальные

Таким образом, для команды chmod 666 ip имеем (Рис. 8.6):

```
work@work:~$ chmod 666 Загрузки/ip
work@work:~$ ls -l Загрузки/ip
-rw-rw-rw- 1 work work 226568 2010-01-18 11:11 Загрузки/ip
work@work:~$ chmod 644 Загрузки/ip
work@work:~$ ls -l Загрузки/ip
-rw-r--r-- 1 work work 226568 2010-01-18 11:11 Загрузки/ip
```

Рис. 8.6. Пример испоьлзования команды chmod

Команда:

chmod 644 имя_файла

устанавливает «обычные» права доступа, т.е. владелец может читать и записывать в файл, а все остальные пользователи - только читать.

Особенности прав доступа для каталогов

Права доступа для каталогов не столь очевидны. Это в первую очередь связано с тем, что система трактует операции чтения и записи для каталогов отлично от остальных файлов. Право чтения каталога позволяет Вам получить имена (и только имена) файлов, находящихся в данном каталоге. Чтобы получить дополнительную информацию о файлах каталога (например, подробный листинг команды ls -l), системы придется «заглянуть» в метаданные файлов, что требует права на выполнения для каталога. Право на выполнение также потребуется для каталога, в который Вы захотите перейти (т.е. сделать его текущим) с помощью команды cd.

T-бит, SUID и SGID

Помимо стандартных «rwx» значений существуют еще и буквы «s» и «t». В действительности, битовая маска прав доступа к файлам содержит 4 группы по 3 бита в каждой. Таким образом, команда chmod 755 это всего лишь краткая запись полной формы команды: chmod 0755.

T-бит обычно используется с каталогами. Обычно, когда t-бит для каталога не установлен, файл в данном каталоге может удалить любой пользователь, имеющий доступ на запись к данному файлу. Устанавливая t-бит на каталог мы меняем это правило таким образом, что удалить файл из каталога может только владелец этого каталога или файла.

Установить t-бит можно при помощи команд:

chmod a+tw имя_файла

chmod 1777 имя_файла

Атрибуты SUID и SGID позволяют изменить права пользователя при запуске на выполнения файла, имеющего эти атрибуты.

Запускаемая программа получает права доступа к системным ресурсам на основе прав доступа пользователя, запустившего программу. Установка же флагов SUID и SGID изменяет это правило таким образом, что назначает права доступа к системным ресурсам исходя из прав доступа владельца файла. Т.е.

запущенный исполняемый файл, которым владеет суперпользователь, получает права доступа к системным ресурсам на уровне суперпользователя (фактически неограниченные). При этом установка SUID приведет к наследованию прав владельца-пользователя файла, а установка SGID -владельца-группы.

Пользоваться такими мощными атрибутами как SUID и SGID нужно с крайней осторожностью, особенно подвергать пристальному вниманию программы и скрипты, владельцем которых является root (суперпользователь), т.к. это потенциальная угроза безопасности системы.

Управление файлами

В ОС Linux следует различать физическую файловую систему, которая отвечает за управление дисковым пространством и размещение файлов в физических адресах диска и логическую файловую систему, которая обеспечивает логическую структуру хранения файлов - пространство имен файлов. ОС Unix и Linux могут работать с различными физическими файловыми системами (Ext2, ext3, ufs), логическое же представление файловой системы в Unix/Linux структурировано. Все файлы в логической файловой системе располагаются в виде дерева, промежуточные вершины которого соответствуют каталогам, и листья - файлам и пустым каталогам. Реально на каждом логическом диске (разделе физического дискового пакета) располагается отдельная иерархия каталогов и файлов. Для получения общего дерева в динамике используется «монтирование» отдельных иерархий к фиксированной корневой файловой системе в качестве ветвей общего дерева. Самым верхом иерархии является корень, который имеет предопределенное имя «/» (слэш). Этот же символ используется как разделитель имен в пути. Далее в корне находятся папки с определенными для каждого дистрибутива именами (etc, home, bin, mnt, proc и т.д.).

Полное имя файла, например, /bin/sh означает, что в корневом каталоге должно содержаться имя каталога bin, а в каталоге bin должно содержаться имя файла sh. Коротким или относительным именем файла называется имя, задающее путь к файлу от текущего рабочего каталога. В каждом каталоге содержатся два специальных имени, имя «.» - ссылка на текущий каталог, и имя «...» - ссылка «родительский» каталог данного текущего каталога, т.е. каталог,

непосредственно предшествующий данному в иерархии каталогов. Так, например, для структуры, показанной на Рис. 8.7 доступ к файлу file2 из текущего каталога (laba) возможен по полному имени: /home/myvar/file2 или по относительному имени: .../.../.../myvar/file5.

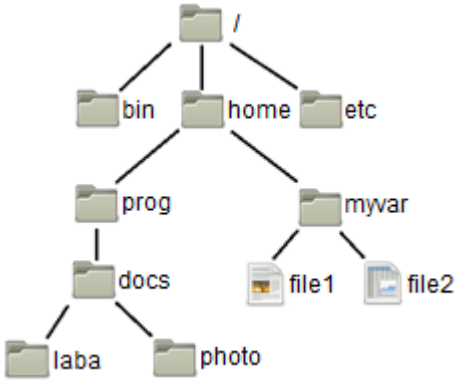


Рис. 8.7. Пример дерева каталогов

Типы файлов

ОС LINUX поддерживают несколько типов файлов:

- Обычные файлы (или регулярные) - представляют собой последовательность байтов. Это текстовые, исполняемые файлы и т.д. Данный тип файла отображается командой ls -l в виде «-» (черточки).
- Каталоги - представляют собой особый вид файлов, которые хранятся во внешней памяти подобно обычным файлам, но их структура поддерживается самой файловой системой. Данный тип файла отображается командой ls -l в виде символа «d».
- Специальные файлы устройств, бывают блочные и символьные. Данный тип файла отображается командой ls -l в виде символа «b» или «c» соответственно. Специальные файлы не хранят данные. Они обеспечивают механизм отображения физических внешних устройств в имена файлов файловой системы. Каждому устройству, поддерживаемому системой, соответствует, по меньшей мере, один специальный файл. При выполнении чтения или записи по отношению к специальному файлу, производится прямой вызов соответствующего драйвера устройства. При этом имена специальных файлов можно использовать практически всюду, где можно использовать имена обычных файлов.

- Ссылка (link). Данный тип файла отображается командой ls -l в виде символа «l». Файловая система UNIX/LINUX обеспечивает возможность связывания одного и того же файла с разными именами.
- Именованный программный канал (pipe) - одно из средств межпроцессных взаимодействий (IPC) в ОС UNIX/LINUX. Данный тип файла отображается командой ls -l в виде символа «p». Именованному программному каналу обязательно соответствует элемент некоторого каталога.
- Сокет (socket)- предоставляют весьма мощный и гибкий IPC. Данный тип файла отображается командой ls -l в виде символа «s». Они могут использоваться для организации взаимодействия программ на одном компьютере, по локальной сети или через Internet, что позволяет создавать распределённые приложения различной сложности. Кроме того, с их помощью можно организовать взаимодействие с программами, работающими под управлением других операционных систем.

Ссылки

Существуют жесткие и мягкие ссылки.

Жесткая ссылка является просто еще одним именем для исходного файла и не является типом файла. Она прописывается в индексном дескрипторе исходного файла (в структуре, хранящей метаданные файла). После создания жесткой ссылки невозможно различить, где исходное имя файла, а где ссылка. Если вы удаляете один из этих файлов (точнее одно из этих имен), то файл еще сохраняется на диске (пока у него есть хоть одно имя - жесткая ссылка). Очень трудно различить первоначальное имя файла и позже созданные жесткие ссылки на него. Поэтому жесткие ссылки применяются там, где отслеживать различия и не требуется. Одно из применений жестких ссылок состоит в том, чтобы предотвратить возможность случайного удаления файла. Особенностью жестких ссылок является то, что они прямо указывают на номер индексного дескриптора, а, следовательно, такие имена могут указывать только на файлы внутри той же самой файловой системы (т. е., на том же самом носителе, на котором находится каталог, содержащий это имя).

Мягкие (символические) ссылки тоже могут рассматриваться как дополнительные имена файлов, но в то же время они представляются отдельными файлами - файлами типа мягких ссылок и являются самостоятельным типом файла. Однако блоки данных файла в системе представляются в одном экземпляре, у файла-ссылки адреса блоков данных те же, что и у исходного файла. В отличие от жестких ссылок мягкие ссылки могут указывать на файлы, расположенные в другой файловой системе, например, на монтируемом носителе, или даже на другом компьютере. Если исходный файл удален, мягкая ссылка не удаляется, но становится бесполезной. Используйте мягкие ссылки в тех случаях, когда хотите избежать путаницы, связанной с применением жестких ссылок.

Создание любой ссылки внешне подобно копированию файла, но фактически как исходное имя файла, так и ссылка указывают на один и тот же реальный файл на диске. Поэтому, например, если вы внесли изменения в файл, обратившись к нему под одним именем, вы обнаружите эти изменения и тогда, когда обратитесь к файлу по имени-ссылке.

Для создания ссылки, используется команда ln (Рис. 8.8):

ln [-f] файл1 [файл2 ...] целевой_файл

Команда ln делает целевой_файл ссылкой на файл1. Файл1 не должен совпадать с целевым_файлом. Если целевой_файл является каталогом, то в нем создаются ссылки на файл1, файл2,... с теми же именами. Только в этом случае можно указывать несколько исходных файлов. Если целевой_файл существует и не является каталогом, его старое содержимое теряется. Аргументы: -f - удаление существующего целевого файла; -s – создание мягкой ссылки (по умолчанию создается жесткая ссылка).

```
work@work:~$ ls
examples.desktop  site1      Музыка
mysite            Видео      Общедоступные
Navicat           Документы  Рабочий стол
PHP редактор      Загрузки   Шаблоны
shares            Картинки
work@work:~$ ln -s Картинки/Kubuntu_leaflet.jpg
work@work:~$ ls
examples.desktop     shares     Картинки
Kubuntu_leaflet.jpg  site1      Музыка
mysite               Видео      Общедоступные
Navicat              Документы  Рабочий стол
PHP редактор         Загрузки   Шаблоны
```

Рис. 8.8. Пример создания ссылки

Файловый менеджер

Управление файлами также можно выполнять с помощью Midnight Commander (mc) - один из файловых менеджеров с текстовым интерфейсом типа Norton Commander для UNIX-подобных операционных систем. Запуск mc из консоли выполняется с помощью команды mc.

Установить файловый менеджер mc так, как показано на Рис. 8.9 , делать этого не надо, он уже установлен в виртуальной машине Ubuntu Server01.

```
work@work:~$ sudo apt-get install mc
```

Рис. 8.9. Установка mc

Достоинство mc том, что есть встроенные средства редактирования и просмотра текстовых файлов (какими являются конфигурационные файлы). При работе с mc необходимы права суперпользователя. Общая последовательность действий по редактированию конфигурационного файла:

- запустить программу (ввод команды mc);
- используя клавиши управления курсором и клавишу «Enter», добраться до нужного файла и выбрать его;
- нажатием клавиши «F4» открыть файл для редактирования;
- внести необходимые изменения;
- сохранить их (клавиша «F2»);
- выйти из режима редактирования (клавиша «F10»).

Просмотр файла выполняется аналогично, только клавишей «F3».

Выйти из mc можно тоже клавишей «F10». Общий вид mc приведен на Рис. 8.10.

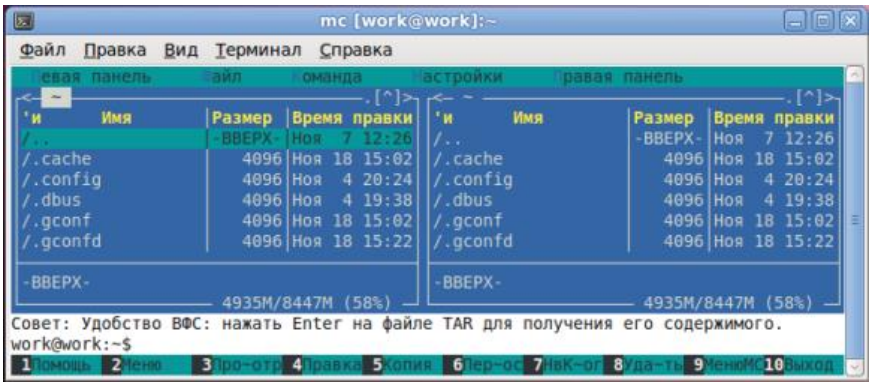


Рис. 8.10. Midnight Commander

8.3.Задания к лабораторной работе:

1. Выведите на экран UID и GID своего пользователя;
2. Создайте группу пользователей с именем usersGroup;
3. Создайте пользователя myUser в группе usersGroup;
4. Выведите на экран UID и GID пользователя myUser;
5. В своем пользователе work создайте текстовый файл, и ограничьте к нему доступ на чтение для всех других групп пользователей. Затем, зайдите в систему от имени пользователя myUser и проверьте возможность открыть этот текстовый файл;
6. От имени пользователя work изменить права доступа к данному файлу, и проверьте изменения;
7. Для одного и тоже файла создайте мягкую и жесткую ссылку в домашнем каталоге. Попробуйте создать ссылки одновременно для нескольких файлов;
8. Проверьте работу файловог менеджера mc.

8.4.Контрольные вопросы

1. Расскажите про идентификационные номера пользователей и групп в Linux.
2. Расскажите о файлах Linux, содержащих информацию о пользователях и группах системы.
3. Как система Linux хранит пароли пользователей и групп?
4. Как организуется разграничение доступа к файлам в Linux?
5. Как изменить права доступа к файлу? Как сделать это с помощью битовой строки?
6. Расскажите о T-бит, SUID и SGID.
7. Расскажите про файловые системы Linux.
8. Какие типы файлов существуют в Linux?
9. Расскажите о мягких и жестких ссылках.

9. Практическое занятие 5. Работа с пакетами и репозиториями Ubuntu

Цель работы:
Изучить возможности Linux при работе с пользователями и управлении правами доступа.

Задачи работы:

- Рассмотреть концепцию Linux при работе с пользователями;
- Изучить управление базами данных пользователей;
- Рассмотреть возможности манипулирования доступом к данным.

9.1.Порядок выполнения работы

- Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
- Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- Выполните все действия, описанные в теоретической части (для тренировки).
- Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.
- Экспортируйте образ сервера на свой носитель или в своё облако для последующей работы.

9.2.Теоретическая часть

Первое, с чем сталкивается пользователь, впервые опробовав линуксную ОС после Windows, - необычность понятия "пакет". Однако ничего заумного в нём нет. Пакетная система работы с линуксным софтом порождена философией UNIX "делай только одну вещь". Грубо говоря, пакет - это архив, который содержит версию формата пакета, сжатую заархивированную управляющую информацию и сжатые заархивированные устанавливаемые файлы. Короче, архив в архиве. При этом пакет содержит только оригинальные файлы, а всякого рода часто встречающиеся общеупотребительные файлы включать в пакет не принято.

Управляющая информация может включать в себя ссылки на другие пакеты, необходимые для работы, т.е. оговаривать так называемые "зависимости". Пример пакета: opera_11.01.1190_i386.deb. Здесь opera - наименование пакета, 11.01.1190 - его версия, i386 - признак, что пакет предназначен для компьютера с чипсетом архитектуры Intel не ниже 386, deb указывает, что пакет предназначен для ОС, совместимых с Debian GNU/Linux.

Нередко интересуются, будет ли пакет, помеченный как i586 или i686, работать на конкретном интеловском компьютере. Узнайте класс своего процессора командой

cat /proc/cpuinfo

и обратите внимание на значение параметра cpu family. Это число должно быть больше или равно первой цифре, следующей за символом i в обозначении пакета. С процессорами AMD проще: пакеты помечаются как AMD64 или AMD32, в зависимости от разрядности процессора. И ещё: не устанавливайте 64-разрядные пакеты, если ваша ОС 32-разрядная: могут быть проблемы. Впрочем, при установке пакетов из репозиториев этого не должно случиться. Вот, собственно, и всё, что на первых порах надо знать о понятии "пакет".

ОС Ubuntu, потомок ОС Debian, обеспечивает пользователю несколько инструментов для управления пакетами. Основу составляют утилита apt (акроним от "Advanced package tool") и низкоуровневая утилита dpkg (акроним

от "Debian package tool"). Кроме того, Ubuntu предлагает и другие интерфейсы к системе пакетов Debian GNU/Linux. К ним относятся в порядке возрастания уровня: псевдографический aptitude, графический Synaptic и, наконец, "Центр Приложений Ubuntu", который, в отличие от остальных, работает только с приложениями. Не буду рассматривать графические интерфейсы, их освоить легко, но возможности "не те". Также оставлю в покое aptitude как производное от apt. Порою встречающиеся утверждения, что aptitude лучше удовлетворяет зависимости, чем apt, считаю просто недоразумениями. Кроме кого, начиная с версии Ubuntu 10.10, aptitude удалён из дистрибутива. По моему убеждению, чтобы быть уверенным пользователем Ubuntu, осваивать следует apt и dpkg.

Репозиторий в Ubuntu — это архив программ, расположенный в интернете. Удобство установки программы из репозитория заключается в том, что пользователю не нужно беспокоиться о совместимости и безопасности устанавливаемого пакета.

apt

Начиная с версии Ubuntu 16.04 доступна новая утилита apt, которая содержит наиболее часто используемые команды из apt-get и apt-cache. Утилита apt предназначена для пользователей, в то время как apt-get можно рассматривать как как инструмент более низкого уровня.

apt-get

apt-get это утилита управления пакетами. apt-get требует прав суперпользователя для своей работы.

Основные команды:

- update - Обновить информацию о пакетах, содержащихся в репозиториях.
- install foo - Установить пакет foo. Скачивание, установка и настройка происходят автоматически. Если для настройки пакета foo нужны дополнительные сведения, будет показан запрос к пользователю.
- upgrade - Обновление пакетов, для которых в репозитории доступны новые версии.
- dist-upgrade - Обновление пакетов, требующих разрешения зависимостей (установка дополнительных или удаление конфликтующих пакетов).

- remove foo - Удаление пакета foo из системы.
- purge foo - Удаление пакета foo и очистка системы от его конфигурационных файлов. Файлы настроек в домашних каталогах пользователей удалены не будут.
- autoremove - Удаление пакета, который более не нужен в системе. Используется для очистки системы от ненужных пакетов. Факт, что пакет более не нужен, определяется следующим образом: если пакет был установлен не сам по себе, а как зависимость для другого пакета, который впоследствии был удалён, значит этот пакет тоже уже не нужен в системе.
- source foo - Получение исходных текстов пакета foo.
- build-dep foo - Получение зависимостей для сборки пакета foo.

Починка базы пакетов

- apt-get также используется для устранения сбоев в базе пакетов вызванных нарушенными зависимостями. Разрешение зависимостей производится командой:
- sudo apt-get install -f

apt-cache

apt-cache – утилита, позволяющая выполнять запросы к кешу APT.

Основные команды:

- search - Поиск пакета по части названия или описания. Поддерживает регулярные выражения.
- show - Информация о пакете: версия, размер, описание и т. п.
- depends - Зависимости указанного пакета.
- rdepends - Обратные зависимости пакета.

apt-key

apt-key служит для добавления ключей от репозиториев в систему. Ключи защищают репозитории от возможности подделки пакета. Подробнее смотрите в статье Репозитории.

Основные команды:

- add - Добавление ключа в базу доверенных ключей.
- del - Удаление ключа из доверенных.

Возможности этого семейства утилит огромны. Основная команда семейства - это apt-get install _наименование_пакета_. Можно устанавливать и сразу несколько пакетов, перечислив их наименования через пробелы. Тут важно понимать, что указываются только имена, ОС вытягивает из репозиториев именно то, что нужно для вашего компьютера и установленной сборки Ubuntu. При установке нового пакета инсталлятор проверяет, были ли ранее установлены пакеты, от которых зависит новый пакет, если нет, то пытается взять такие пакеты из доступных репозиториев. При недоступности таковых выдаёт ошибку. Но иногда следует указывать имена расширенные, которые включают в себя мажор версии пакета. Так поступают, если в репозитории находятся несколько версий и пользователю предоставляется возможность выбирать. Например, команда

sudo apt-get install virtualbox-4.0

сначала определит, не установлена ли другая версия virtualbox, если да, то с санкции пользователя "снесёт" её и установит именно virtualbox-4.0. А вот команда

sudo apt-get install virtualbox

просто попытается установить из репозиториев самую свежую версию этой замечательной виртуальной машины. При установке пакеты сначала попадают в кэш /var/cache/apt/archives, затем устанавливаются куда надо. Если очень интересно, куда, то это всегда можно узнать командой whereis, например, whereis virtualbox. По мере взросления вашей сборки Ubuntu в кэше накапливается много нужного и ненужного. Последнее занимает место и может быть удалено, о чём чуть позже.

Нередко требуется удалить установленный пакет - ну, погорячились, установили не то, что надо, или же новый пакет в управляющей информации не содержит сценария удаления своего предшественника. Команда

sudo apt-get remove virtualbox

"снесёт" установленную версию пакета virtualbox, однако сохранит все конфигурационные файлы с тем, чтобы ими могла воспользоваться другая версия пакета. При этом сам пакет останется в кэше, и при необходимости его можно будет установить оттуда повторно. А вот команда

sudo apt-get --purge remove virtualbox
"снесёт" и конфигурационные файлы, и сам пакет в кэше.

Многие пакеты ОС Ubuntu обновляет в порядке штатной процедуры обновления, но только те, которые поддерживает компания Canonical, а также те, для которых обновления предоставляются сообществом Ubuntu. Остальные пакеты в пределах мажора версии можно обновлять самостоятельно. Например, команда

sudo apt-get upgrade opera -u
обновит ваш ранее установленный браузер Opera, повысив минор версии, если, конечно, таковая существует в репозитории, прописанном в /etc/apt/sources.list. При обновлении приложения строится дерево зависимостей, проверяются обновления для них и нередко обновляется не основной пакет, а только пакеты, от которых он зависит. Необязательная опция -u позволяет видеть, что именно обновляется. Кстати, теперь для добавления репозиториев нет необходимости редактировать файл /etc/apt/sources.list, достаточно воспользоваться командой add-apt-repository _источник_. Например, можно добавить источник deb-пакетов от Google вот такой командой:

sudo add-apt-repository "deb http://dl.google.com/linux/deb/ stable main"
После добавления нового источника пакетов никогда не забывайте проиндексировать список источников командой

sudo apt-get update
Если вы не знаете или забыли, какая версия пакета установлена и какие библиотеки он использует, то не стесняйтесь задать системе вопрос, например,

командой apt-cache search _имя_пакета_, при этом мажор версии пакета, естественно, не указывается, например,

apt-cache search *sysstat*

В ответ вы получите исчерпывающую информацию о пакетах, непосредственно касающихся вашего приложения, в данном случае утилиты сбора статистики о работе вычислительной системы *sysstat* . Теперь, зная версию пакета, вы можете определить все зависимости, которые он за собой тянет, командой

sudo apt-cache depends *sysstat* -4.0

Это очень полезная команда, так как с её помощью можно получить рекомендации, какие пакеты следует доустановить для более полноценной работы приложения. Дело в том, что иногда не все зависимости указываются в основном пакете приложения; такие непрописанные зависимости называются вынесенными. Их можно, конечно, и проигнорировать, приложение будет работать, однако функциональность его останется неполной.

Чтобы узнать, какие приложения установлены, необходимо воспользоваться командой

sudo apt-cache search all

При этом на экран будет выведен только список установленных приложений с краткими комментариями о них; зависимости, библиотеки и прочее выведены не будут. Иногда бывает нужным знать, какие ветви и файлы создала или затронула установка пакета. Для этой цели служит команда apt-file search _строка_, где _строка_ - это набор символов, последовательность которых встречается в полном пути. Правда, apt-file по умолчанию отсутствует, и эту утилиту сначала надо установить командой

sudo apt-get install apt-file

Имея в распоряжении apt-file, нетрудно найти директории, затронутые установкой конкретного приложения, например, virtualbox-4.0:

sudo apt-file search virtualbox-4.0

Для очистки кэша от устаревших пакетов, которые больше не скачиваются и не нужны, имеется очень полезная команда

sudo apt-get autoclean

Чтобы очистить весь кэш от пакетов, если срочно требуется место на диске, на работоспособности системы это как будто не сказывается, но если вам потребуется переустановить какой-либо ранее установленный пакет, то придётся его заново скачивать. Делается такая чистка следующей командой, хотя я бы подобное не рекомендовал:

sudo apt-get clean

Если надо убедиться, всё ли в порядке с пакетами, установленными в системе, то это проверяют командой

sudo apt-get check

Если не получите никаких сообщений о неудовлетворённых зависимостях, то всё отлично. Правда, вынесенных зависимостей это не касается. Чтобы уладить проблемы с обнаруженными невынесенными зависимостями, просто введите

sudo apt-get -f install

Я перечислил apt-минимум, которым должен владеть пользователь Ubuntu, желающий свободно управлять пакетным хозяйством своей системы, пополняемым из репозиториев. Однако, помимо пакетов из репозиториев, имеется огромное количество и других deb-пакетов, которые разработчики предлагают на своих сайтах. Чтобы воспользоваться этим богатством, нужно спуститься уровнем ниже.

dpkg

Полную информацию об этой утилите можно почерпнуть в debianadmin.com/debianubuntu-package-management-using-dpkg.html. Я не рекомендую новичкам пользоваться этой утилитой для установки-удаления пакетов. dpkg не разрешает зависимости: если для устанавливаемого пакета требуются другие пакеты, то их нередко приходится устанавливать вручную. Кроме того, всякий раз при установке пакета требуется указывать путь к нему:

sudo dpkg -i /путь/пакет.deb
и выявлять зависимости командой:

dpkg --info /путь/пакет.deb
Неудобно! Проще скачанный пакет переместить из директория загрузок в вышеупомянутый кэш и установить его оттуда утилитой apt-get install, не указывая путь. Тем не менее, dpkg - весьма достойная утилита. Вот как без aptitude проще всего узнать, какие пакеты установлены в системе? Нет проблем:

dpkg -l
С помощью dpkg можно легко узнать, какая именно версия пакета установлена в системе и установлен ли пакет вообще; например, команда

dpkg -l 'virtualbox*'
выводит установленную версию виртуальной машины virtualbox, предваряя её суффиксом ii, но также выводит и старую версию пакета, если он не удалялся из кэша, предваряя его имя суффиксом rc. Более подробную информацию о версиях пакетов, по сравнению с предыдущей командой, можно получить, привлекая grep:

dpkg -l | grep 'virtualbox*'
Чтобы определить ширину колонок при выводе информации о пакетах, укажите её явно, например:

COLUMNS-150 dpkg -l 'virtualbox*'

Утилита dpkg также при желании позволяет посмотреть содержимое пакета:

dpkg -content /путь/пакет.deb

О репозиториях

Владельцы репозитория скомпилировали исходный код, оформили его в виде пакета для установки и протестировали программу на совместимость с операционной системой. Для программного обеспечения из репозитория имеется определенная гарантия, что программа A, версии B, заработает в дистрибутиве версии C. Кроме этого, получение программы из репозитория гарантирует, что в программу не внесены несанкционированные изменения третьей стороной.

Виды репозиториев

В Ubuntu имеется штатных 4 репозитория:

1. **Main** — это основной репозиторий, содержащий официально поддерживаемые приложения.
2. **Restricted** — здесь содержится официально поддерживаемый софт, но который не лицензирован на условиях открытой лицензии GPL.
3. **Universe** — для программ, не имеющих официальной поддержки. Этот репозиторий поддерживается силами сообщества.
4. **Multiverse** — содержит ПО, которое не является свободным.

Добавление и удаление репозиториев

Кроме этих четырех репозиториев существует огромное количество сторонних, которые еще иногда называют репозиториями третьей стороны. Любой желающий может создать свой частный репозиторий, который потом можно будет подключить к Ubuntu.

По умолчанию, репозитории Universe и Multiverse доступны, но если вы хотите отключить их, измените /etc/apt/sources.list и закомментируйте следующие строки:

deb http://archive.ubuntu.com/ubuntu precise universe multiverse

deb-src http://archive.ubuntu.com/ubuntu precise universe multiverse

deb http://us.archive.ubuntu.com/ubuntu/ precise universe

deb-src http://us.archive.ubuntu.com/ubuntu/ precise universe

deb http://us.archive.ubuntu.com/ubuntu/ precise-updates universe

deb-src http://us.archive.ubuntu.com/ubuntu/ precise-updates universe

deb http://us.archive.ubuntu.com/ubuntu/ precise multiverse

deb-src http://us.archive.ubuntu.com/ubuntu/ precise multiverse

deb http://us.archive.ubuntu.com/ubuntu/ precise-updates multiverse

deb-src http://us.archive.ubuntu.com/ubuntu/ precise-updates multiverse

deb http://security.ubuntu.com/ubuntu precise-security universe

deb-src http://security.ubuntu.com/ubuntu precise-security universe

deb http://security.ubuntu.com/ubuntu precise-security multiverse

deb-src http://security.ubuntu.com/ubuntu precise-security multiverse

Добавить новый источник пакетов очень просто. Делается это всего одной командой:

sudo apt-add-repository ppa:user/repository

Здесь ppa:user/repository — имя зарегистрированного репозитория. Удаляется источник пакетов аналогично, с помощью программы apt-add-repository, но с указанием ключа -r:

sudo apt-add-repository –r ppa:user/repository

Следует отметить, что просто подключить новый репозиторий недостаточно, нужно попросить операционную систему загрузить из него список доступных программ актуальных версий:

sudo apt-get update

После проделанных манипуляций можно приступать к установке, которая осуществляется самым привычным образом:

sudo apt-get install имя-программы

9.3.Задания к лабораторной работе:

1. Обновите индекс пакетов вашего виртуального сервера (…update).

2. Обновите вашу систему (…upgrade).
3. Установите утилиту измерения производительности вашей ОС sysstat.
4. Проверьте её версию.
5. Запустите утилиту командами iostat , затем mpstat.
6. Самостоятельно найдите описание и изучите команды sysstat, проверьте их.

9.4.Контрольные вопросы

10. Расскажите про пакеты ubuntu.
11. Расскажите о репозиториях ubuntu.
12. Как добавить репозитарий к стандартному списку Ubuntu?
13. Как обновит индекс пакетов Ubuntu?
14. Как обновить все пакеты в Ubuntu?
15. Как обновить только одну программу в Ubuntu?
16. Как установить новую утилиту в Ubuntu?
17. Как удалить утилиту из Ubuntu?
18. Как полностью удалить утилиту Ubuntu?
19. Как очистить кэш пакетов Ubuntu?
20. Как система Linux хранит пароли пользователей и групп?
21. Для чего служит утилита sysstat?
22. Перечислите команды утилиты sysstat.
23. Как запустить демон сбора статистик и sysstat?

10. Лабораторная работа №5 Управление каталогами.

Цель работы:
Изучение структуры файловой системы и возможностей командного языка UBUNTU по управлению каталогами.

10.1. Порядок выполнения работы

- Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
- Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- Выполните все действия, описанные в теоретической части (для тренировки).
- Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.
- Экспортируйте образ сервера на свой носитель или в своё облако для последующей работы.

10.2. Теоретическая часть

Изучаются команды:

mkdir, rmdir - для создания и уничтожения каталогов;

ls - вывод листинга каталога. «что здесь есть?»;

pwd - вывод на экран полного имени текущего каталога, «где я нахожусь?»;

cd - смена текущего каталога, «перейти в»;

find, grep - поиск файлов в системе каталогов;

>маршрутное – имя - файла - создание пустого файла.

Эта работа посвящена изучению структуры файловой системы и возможностей командного языка Ubuntu по управлению каталогами. Для начала рассмотрим основные команды, используемые в ОС UBUNTU при работе с каталогами..

Разделителем элементов пути в системах UBUNTU служит символ V. В отличие от таких систем, как, например, MSDOS и Windows, в которых каждому тому соответствует отдельный корневой каталог, обозначаемый именем тома, в системе UBUNTU есть только один корневой каталог. Он обозначается ‘/’ - Все дополнительные тома, подключаются к основному дереву каталогов так, что корневой каталог каждого из этих томов становится просто одним из каталогов в файловой системе. Каталог для подключения может быть выбран произвольно. Операция подключения тома к файловой системе носит название монтирования, и может производиться в любой момент во время работы системы.

В системе обычно присутствуют следующие каталоги:

/ — корневой каталог;

/bin— каталог с пользовательскими программами;

/sbin — каталог программ для администрирования системы;

/etc — каталог с конфигурационными файлами программ;

/home — каталог, в котором создаются домашние каталоги пользователей;

/lib — каталог с динамическими и статическими библиотеками;

/boot — каталог, содержащий файлы системного загрузчика;

/mnt — каталог, в который, как правило, производится монтирование;

/dev — каталог, содержащий специальные файлы устройств;

/opt — каталог, в который устанавливается ПО сторонних производителей;

/usr — каталог, в котором хранятся в режиме доступа только для чтения разделяемые данные, такие как исполняемые файлы программ, документация, библиотеки и другие системные ресурсы;

/root— каталог, являющийся домашним для пользователя roo1;

/var — каталог, содержащий журналы, файлы баз данных, кеши разного рода; .

/tmp — каталог для хранения временных файлов.

Для печати текущего каталога:

pwd

Для смены каталога:

cd <путь к каталогу>

Пример: переход к каталогу var.

```
rogdy@ubuntu:~$ cd /var
rogdy@ubuntu:/var$ pwd
/var
rogdy@ubuntu:/var$ ls
backups  cache  crash  games  lib  local  lock  log  mail  opt  run  spool  tmp
rogdy@ubuntu:/var$
```

Попробуйте объяснить смысл каждой строки из данного примера. Какие операции запрашивает пользователь, и какие ответы даёт система?

Если команда запущена без указания каталога, то переход производится в домашний каталог пользователя. Вообще, для указания домашнего каталога пользователя можно использовать специальный символ ‘ ~’. Так, для перехода в папку tmp, находящуюся в домашнем каталоге можно воспользоваться следующей командой:

```
rogdy@ubuntu:/var$ cd ~ /tmp
```

Домашний каталог пользователя обычно располагается в каталоге /home и называется по имени пользователя. Например, для пользователя user1 домашний каталог будет таким: /home/user1.

Для создания каталога:

mkdir <список имен каталогов>

```
rogdy@ubuntu:~$ mkdir KaTaLoG
```

Если требуется создать сразу несколько вложенных друг в друга каталогов, можно воспользоваться ключом -p:

```
rogdy@ubuntu:~$ mkdir -p vheshny/vnutrenny
```

Для удаления каталога:

```
rogdy@ubuntu:~$ rmdir KaTaLoG
```

Пример: создание каталогов и работа с ними.

```
rogdy@ubuntu:~$ mkdir abc
rogdy@ubuntu:~$ cd abc
rogdy@ubuntu:~/abc$ mkdir ABC
rogdy@ubuntu:~/abc$ ls
ABC
```

Попробуйте объяснить смысл каждый строки из данного примера. Какие операции запрашивает пользователь, и какие ответы даёт система?

Команда удаляет только пустые каталоги. Ключ -p подобен такому же ключу команды mkdir, и позволяет удалить сразу несколько каталогов, вложенных друг в друга, если все они пусты.

/ Корневой каталог. Это родительский каталог для всех каталогов и файлов в файловой системе UBUNTU.

/bin Каталог исполняемых модулей командной строки. Данный каталог содержит все исполняемые модули «родных» команд UBUNTU.

/dev Каталог устройств, содержащий специальные файлы для байт-ориентированных и блок-ориентированных устройств, таких как принтеры и клавиатуры. В данном каталоге существует файл под называнием null, который называется bit bucket и который может использоваться для перенаправления вывода в никуда.

/etc Файлы системной конфигурации и каталог исполняемых файлов. Большая часть административных файлов, а также файлов, связанных с командами, хранится здесь.

/lib В каталоге хранятся библиотеки компилятора C.

/lost+found Данный каталог содержит обрабатываемые файлы, если система отключилась ненормально. Система использует данный каталог для восстановления файлов. В каждом разделе диска есть только один каталог lost+found.

/usr Данный каталог имеет несколько подкаталогов, таких как adm, bin, etc и include**.** Например. /usr/include содержит файлы заголовков для компилятора C. /home содержит домашние каталоги пользователей.

Для создания каталогов используется команда mkdir. Можно указывать как полный так и относительный путь. Поэтому можно создавать дерево каталогов: определить относительно или абсолютно корень, после чего создать относительно нового каталога новые поддирективы.

Команду ls (с ее многочисленными опциями) можно использовать для получения информации об одном или нескольких файлах или каталогах системы. Используйте ls для генерации списка файлов и каталогов в различном порядке, например по имени или по времени. Возможно распечатывать лишь отдельные детали о файлах и каталогах, например только имя файла.

10.3. Задание

1. Определите уникальное имя вашего головного личного каталога. Объясните структуру полного маршрутного имени каталога.
2. Создайте два поддерева из одного и из двух каталогов.
3. С использованием команды ls проверьте факт построения дерева подкаталогов.
4. Посмотрите содержимое пустых подкаталогов, т.е. новых подкаталогов, не содержащих файлов. Объясните их содержание.
5. Сделайте текущим последний каталог меньшего дерева.
6. Определите полное маршрутное имя.
7. Смените текущий последний каталог на подкаталог большего дерева.
8. Определите его полное маршрутное имя.
9. Поместите в созданные подкаталоги по 2-3 пустых файла не выходя из текущего. Используйте при этом разные способы задания маршрутного имени подкаталогов.
10. Просмотрите содержимое каталогов. Объясните содержания каждого поля каталога.
11. Установите в качестве текущего HOME-каталога.

12. Найдите обычные файлы с определением их полных маршрутных имен. Выполните то же для различных комбинаций известных вам условий поиска файлов.

13. Проделайте предыдущие задания для файлов типа каталог.

14. Выведите на экран принадлежащую вам регистрационную запись с использованием команды grep.

15. Уничтожьте все построенные вами подкаталоги. Получите подтверждение выполнения команд по содержимому домашнего каталога.

16. Проанализируйте с использованием команды history содержание лабораторной работы, продумайте ответы на нижеприведенные контрольные вопросы и сдайте выполненную работу преподавателю. После получения зачета по работе – уничтожьте все созданные файлы и корректно выйдите из системы.

10.4. Контрольные вопросы.

1. Какие системные имена каталогов вам известны?

2. Каким образом можно построить отдельный каталог или цепочку каталогов?

3. Для чего и каким образом переопределяются текущие каталоги?

4. Как обратиться к файлам параллельных ветвей дерева каталогов? К вышележащему каталогу?

5. Какие условия поиска файлов вы знаете? Как комбинируются условия поиска? Как осуществляется поиск по дереву каталогов?

6. Какова последовательность действий при удалении одного каталога? Цепочки каталогов?

7. Объясните назначение и содержание каждого поля каталога.

8. Как отличить по содержимому каталога типы файлов, содержащихся в ваших каталогов.

9. Какую информацию содержит «пустой» вновь созданный каталог?

10. Как осуществить поиск файлов в системе каталогов по фрагментам текста файлов?

11. Практическое занятие № 6. Текстовый редактор vi ОС UBUNTU

Цель работы

Изучение основных возможностей встроенного текстового редактора vi – наиболее распространённого средства для построения текстовых файлов, исходных текстов программ и shell-процедур.

11.1. Порядок выполнения работы

- Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
- Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- Выполните все действия, описанные в теоретической части (для тренировки).
- Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.
- Экспортируйте образ сервера на свой носитель или в своё облако для последующей работы.

11.2. Теоретическая часть

Изучаются вопросы:

- вход в редактор и выход, сохранение файлов;
- ввода текста;
- удаления фрагментов текста;
- копирования фрагментов текста в буферную
область памяти;
- вставки содержимого буфера в текст файла;
- редактирования (изменения) текста;
- поиска строк файлов по их фрагментам;
многострочных операций с файлом (префиксные
команды)

vi — текстовый редактор командной строки в Ubuntu и других ОС UBUNTU. При запуске vi не открывается новое графическое окно, просмотр и редактирование файла производится в текущем окне терминала. За счет того, что vi может работать и без Графического пользовательского интерфейса (GUI), он может использоваться для редактирования файлов.

Запуск редактора:

а) vi   myfile (одно или несколько имен файлов через пробелы для последовательного вызова их на редактирование).

Если такого файла нет, то появится начало пустого файла; курсор установится в начале первой строки.

б) vi + myfile

На экране конец файла; а курсор - в начале последней строки.

Пример:

в) vi +10 myfile

На экране - часть файла и строка 10 - в центре экрана;

курсор - в начале этой строки.

Режимы работы.

Редактор работает в нескольких режимах:

- В режиме вставки (редактирования). Нажатие на клавишу приводит к вставке соответствующего символа в редактируемый текст.
- В командном режиме
  - нажатие на любую клавишу воспринимается как команда редактору, которая немедленно исполняется;
  - в этом режиме можно ввести команду с параметрами в командной строке.

Поэтому при работе с Vi пользователю всегда нужно обращать внимание на то, в каком режиме находится редактор.

Командный режим.

Редактор Vi всегда начинает работу в командном режиме. В этом режиме есть два способа отдавать команды редактору.

1. нажатие практически на любую клавишу редактор воспринимает как определенную команду. Команды не отображаются, а сразу выполняются. Например, нажатие стрелок не перемещает курсор, а выводит на экран символы, соответствующие командам, расположенным по этим клавишам. Переключиться из режима вставки в командный режим - при помощи Esc.
2. для ввода более сложных многословных команд используется командная строка, вызов которой осуществляется нажатием клавиши ":" .

Режим вставки (редактирования).

Для перехода к этому режиму следует использовать команду insert, выполнение которой осуществляется при нажатии на клавишу "i" в командном режиме. При переключении в этот режим внизу экрана появляется надпись Insert, можно вносить изменения в текст документа.

Выход из vi

а) с сохранением изменений: переключиться в командный режим (Esc), ввести команду

: w q

б) без сохранения изменений: переключиться в командный режим (Esc), ввести команду

: q !

Перемещение курсора

0 - в начало строки

$ - в конец строки

w - в начало слова

b - в начало предыдущего слова

Удаление

dw - удаление слова

d$ - удаление до конца строки

d0 - удаление до начала строки

d7w - удаление 7 слов

Изменение (замена)

c$ (или C)   <текст замены (может быть из нескольких строк)> Esc - замена конца строки (от курсора);

c^ <текст замены> Esc - замена от начала строки до курсора

cc < текст > Esc - замена одной строки;

5cc < текст > Esc - замена пяти строк.

Создание новой строки

o - пустая строка после текущей строки;

O - пустая строка перед текущей строкой.

Использование буфера обмена

Занести в буфер:

yw - сохранить слово (курсор - в начале слова);

yy - сохранить одну строку ;

5yy - сохранить 5 строк; и т.п.

При выполнении команд ndd (где n - число) удаляемые n строк заносятся в буфер.

Вставка текста из буфера:

- p - после текущей строки;

- P - перед ней.

Пример работы с текстовым редактором

Vi + myfile.txt

```
i
helo!
How are you?
```

```
i
helo!
hOW ARE YOU?
C$ JJJ
```

```
i
helo!
hOW ?
C$ JJJ
```

Попробуйте объяснить смысл каждый строки из данного примера. Какие операции запрашивал пользователь? Какие запросы он вводил с клавиатуры, чтобы получить тот или иной результат?

11.3. Задание

1.Войдите в редактор с созданием нового пустого файла с произвольным именем и расширением -.1.

2. Поместите в созданный файл текст, включающий не менее четырех строк с несколькими словами в каждой.

3. Вставьте по одной пустой строке до и после одной из строк файла.

4. Заполните пустые строки произвольным текстом.

5.Вставьте еще по одной строке в середину текста файла без предварительного резервирования пустых строк.

6. Перейдите в режим редактирования и выполните произвольное редактирование отдельных слов и строк файла с использованием всех команд из групп «Команды изменения текста» и «Команды отмены произвольных изменений в текущей строке».

7. Выйдите из редактора с сохранением файла. Убедитесь в сохранении созданного файла.

8. Войдите опять в vi для редактирования созданного файла с использованием команд из групп "Команды копирования в буфер" и "Команды вставки буфера в текст".

9. Поменяйте местами несколько слов в строках файла.

10. Поменяйте местами несколько строк.

11. Поменяйте местами последовательно начало строки с её концом и наоборот.

12. Выполните операцию поиска строк файла по заданным их фрагментам с различными направлениями поиска (см. «Команды поиска строки файла по фрагменту её текста»).

13. Не выходя из vi, перепишите полученную в результате редактирования версию файла в файл с тем же именем, но с расширением .new.

14. Добавьте первые 3-и строки редактируемого файла к файлу .1.

16. Не выходя из vi, загрузите в буфер файл с расширением .1.

17. Последовательно удалите части строк и несколько строк с использованием команд из групп «Команды удаления текста».

18. Выйдите из редактора без сохранения файла.

19. Просмотрите и проанализируйте содержимое редактируемых файлов в текущем каталоге.

20. Уничтожьте созданные файлы

11.4. Контрольные вопросы

1. В чем особенности и преимущества встроенного редактора vi ОС UBUNTU?

2. Какие два основных режима работы использует редактор? Как осуществляется переключение режимов?

3. Какую структуру имеет экран при редактировании файла? Назначение полей экрана?

4. Как в vi организованно редактирование открытого в нем файла?

5. Как организованна работа с клавишными командами редактора?

6. Что такое «префиксные команды» и их назначение? Как организована работа с командами этого вида?

7. Какие функции редактора вы использовали при выполнении лабораторной работы?

12. Практическое занятие 7. Введение в shell-программирование

Цель работы

Ознакомление со средствами языка shell, для создания процедур обработки данных. Изучаются вопросы оформления shell-процедур.

12.1. Порядок выполнения работы

- Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью
- Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- Выполните все действия, описанные в теоретической части (для тренировки).
- Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.
- Экспортируйте образ сервера на свой носитель или в своё облако для последующей работы.

12.2. Теоретическая часть

Изучение команд:

Set – присваивание значения параметрам, передаваемым процедурам;

Echo, read, banner – вспомогательные команды для ввода и вывода информации;

If, then, else – команды проверки условий и ветвления вычисления в процедуре;

Test – проверка файлов, числовых величин, строк символов;

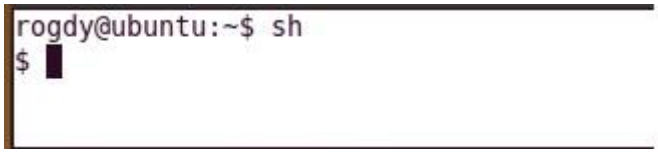
While, until, for – команды построения циклических процедур;

- правила построения и постановки значений переменных;
- вычисление арифметических выражений;
- обработка символьных строк.

Командный язык shell (в переводе - раковина, скорлупа) фактически есть язык программирования очень высокого уровня. На этом языке пользователь осуществляет управление компьютером. Обычно, после входа в систему вы начинаете взаимодействовать с командной оболочкой.

Программирование на языке Shell происходит в окне терминала, а сами операции выполняются сразу, как только нажимается клавиша Enter после введения текста функции. (В отличие, скажем, от языка C++, где для того, чтобы компьютер выполнил введённые вами функции, необходимо запускать написанную программу нажатием клавиши F9. В Shell такого нет, а значит, нужно быть внимательным при вводе функций и команд, т.к. в случае ошибки придётся переписывать всю программу заново).

Для того, чтобы, работая в терминале, переключиться в среду Shell, необходимо ввести команду sh. Признаком того, что оболочка (shell) готова к приему команд служит выдаваемый ею на экран промптер("$").



Имя shell-переменной - это начинающаяся с буквы последовательность букв, цифр и подчеркиваний.

Значение shell-переменной - строка символов.

Для присваивания значений переменным может использоваться оператор присваивания "=".

var_1=13 - "13" - это не число, а строка из двух цифр. (аналог из C++: char var_1[3] = “13”))

var_2="ОС UBUNTU" - здесь двойные кавычки (" ") необходимы, так как в строке есть пробел.

ВАЖНО: Обратим внимание на то, что, как переменная, так и ее значение должны быть записаны без пробелов относительно символа "=".

Возможны и иные способы присваивания значений shell-переменным. Так например запись,

```
$ DAT=`date`
```

приводит к тому, что сначала выполняется команда "date" (обратные кавычки говорят о том, что сначала должна быть выполнена заключенная в них команда), а результат ее выполнения, вместо выдачи на стандартный выход, приписывается в качестве значения переменной, в данном случае "DAT".

```
$ echo $DAT
Mon Oct 17 16:46:28 MSD 2011
```

Можно присвоить значение переменной и с помощью команды "read", которая обеспечивает прием значения переменной с (клавиатуры) дисплея в диалоговом режиме (аналог из C++: scanf или cin). Обычно команде "read" в командном файле предшествует команда "echo", которая позволяет предварительно выдать какое-то сообщение на экран. Например:

```
$ echo -n "vvedite znachenie x"
vvedite znachenie x$ read x
5
$ echo $x
5
```

При выполнении этого фрагмента командного файла, после вывода на экран сообщения

Введите трехзначное число:

интерпретатор остановится и будет ждать ввода значения с клавиатуры. То число, которое пользователь введёт с клавиатуры, станет значением переменной "x".

Одна команда "read" может прочитать (присвоить) значения сразу для нескольких переменных. Если переменных в "read" больше, чем их введено (через пробелы), оставшимся присваивается пустая строка. Если передаваемых значений больше, чем переменных в команде "read", то лишние игнорируются.

ПРЕДУПРЕЖДЕНИЕ. На самом деле интерпретатор для продолжения работы ждет лишь нажатия клавиши. Введенное вами число воспринимается им не как число, а как последовательность символов(!). Интерпретатор не проверяет, что вы ввели. Поэтому в качестве значения переменной может оказаться любая введенная абракадабра или просто нажатие , как значение пустой строки. (Для обеспечения проверки формата ввода следует написать свою команду).

При обращении к shell-переменной необходимо перед именем ставить символ "$" для получения Значения этой переменной. Сравните две команды, по-разному набранные с клавиатуры и заметьте разницу в результатах, распечатываемых программой на экране.

```
$ echo x
x
$ echo $x
5
```

Запись echo x печатает исключительно те буквы, которые введены после команды echo. (Аналог в C: printf(“x”) и printf(“%d”, x)).

И еще один пример. Фрагмент командного файла:

```
$ echo x = $x
```

выдаст на экран

```
x = 5
```

Что наглядно показывает, что команда echo может за один раз выводить на экран несколько переменных.

Команда "set" устанавливает значения параметров. Это бывает очень удобно. Например, команда "date" выдает на экран текущую дату, скажем, "Mon Oct 13 16:46:28 2011", состоящую из пяти слов, тогда

```
$ set `date`
$ echo $1 $3
```

выдаст на экран

```
Mon 17
```

Команда "set" позволяет также осуществлять контроль выполнения программы, например:

set -v на терминал выводятся строки, читаемые shell.

Set +v отменяет предыдущий режим.

Set -x на терминал выводятся команды перед выполнением.

Set +x отменяет предыдущий режим.

Команда "set" без параметров выводит на терминал состояние программной среды

Как во всяком языке программирования в тексте на языке shell могут быть комментарии. Для этого используется символ "#". Все, что находится в строке (в командном файле) левее этого символа, воспринимается интерпретатором как комментарий. Например,

```
$ # ja kommentariy!
$ ## ja tozhe
$ ### i ja =P
```

Как во всяком процедурном языке программирования в языке shell есть операторы. Ряд операторов позволяет управлять последовательностью выполнения команд. В таких операторах часто необходима проверка условия, которая и определяет направление продолжения вычислений.

Команда test ("[ ]")

Команда test проверяет выполнение некоторого условия. С использованием этой (встроенной) команды формируются операторы выбора и цикла языка shell.

Два возможных формата команды:

test условие

или

[условие]

Пробелы должны быть и между значениями и символом сравнения или операции (как, кстати, и в команде "expr") . Про последнюю следует сказать подробнее. Именно команда expr даёт программе возможность воспринимать ту или иную переменную именно как число. (Иными словами, если сравнивать два ЯП, Shell и C, запись echo `expr $x` будет аналогична записи printf(“%d”, x).

Разница в том, что в языке Си тип данных (char, int и пр.) определяется и устанавливается сразу при объявлении переменной, а в shell тип данных определяется после того, как значение было записано в переменную и может меняться в теле программы).

Пример работы с командами и операторами языка shell.

```
$ echo date: `date`
date: Mon Oct 17 17:27:46 MSD 2011
$ s=`expr $6 - $3`
$ echo $s
1994
```

Попробуйте объяснить смысл каждой строки из данного примера. Какие операции запрашивает пользователь, и какие ответы даёт система?

В shell используются условия различных "типов".

Условный оператор "if"

В общем случае оператор "if" имеет структуру

```
if [условие] ; echo $0
  then список
       [elif условие
        then список]
  [else список]
Fi
```

Здесь "elif" сокращенный вариант от "else if" может быть использован наряду с полным, т.е. допускается вложение произвольного числа операторов "if" (как и других операторов). Разумеется "список" в каждом случае должен быть осмысленный и допустимый в данном контексте.

Конструкции

```
[elif условие
 then список]
```

и

```
[else список]
```

не являются обязательными (в данном случае для указания на необязательность конструкций использованы квадратные скобки - не путать с квадратными скобками команды "test"!).

пример использования оператора if:

```
$ z=7
$ if [  expr $z > 0`] ; echo $0
> then "POLOZITELNO"
> [ elif `expr &z<0` then echo "otrizatelno"]
> [else echo "null"]
> fi
sh
sh: POLOZITELNO:
```

Самая усеченная структура этого оператора

if условие

then список

fi

Обратите внимание, что структура обязательно завершается служебным словом "fi". Число "fi", естественно, всегда должно соответствовать числу "if".

Примеры.

Пусть написан расчет "if-1"

```
$ if [ $1 -gt $2]
> then pwd
> else echo $0: Hello!
> fi
```

Тогда вызов расчета

if-1 12 11

даст

/home/sae/STUDY/SHELL,

а

if-1 12 13

даст

if-1 : Hello!

Оператор вызова ("case")

Оператор выбора "case" имеет структуру:

```
case   строка  in
      шаблон)  список команд;;
      шаблон)  список команд;;
         ...
esac
```

Здесь "case" "in" и "esac" - служебные слова. "Строка" (это может быть и один символ) сравнивается с "шаблоном". Затем выполняется "список команд" выбранной строки. Непривычным будет служебное слово "esac", но оно необходимо для завершения структуры.

Пример.

```
###
# case-1: Структура "case".
#       Уже рассматривавшийся в связи со
#       структурой "if" пример проще и
#       нагляднее можно реализовать с
#       помощью структуры "case".
echo -n " А какую оценку получил на экзамене?: "
read z
case $z in
      5) echo Молодец !          ;;
      4) echo Все равно молодец !  ;;
      3) echo Все равно !        ;;
      2) echo Все !              ;;
      *) echo  !                 ;;
esac
```

Непривычно выглядят в конце строк выбора ";;", но написать здесь ";" было бы ошибкой. Для каждой альтернативы может быть выполнено несколько команд. Если эти команды будут записаны в одну строку, то символ ";" будет использоваться как разделитель команд.

Обычно последняя строка выбора имеет шаблон "*", что в структуре "case" означает "любое значение". Эта строка выбирается, если не произошло совпадение значения переменной (здесь $z) ни с одним из ранее записанных

шаблонов, ограниченных скобкой ")". Значения просматриваются в порядке записи.

```
###
# case-2:  Справочник.
#         Для различных фирм по имени выдается
#         название холдинга, в который она входит
case $1 in
        ONE|TWO|THREE) echo Холдинг: ZERO    ;;
              MMM|WWW) echo Холдинг: Not-Net ;;
      Hi|Hello|Howdoing) echo Холдинг: Привет! ;;
                  *) echo Нет такой фирмы  ;;
esac
```

Оператор цикла с перечислением ("for")

Оператор цикла "for" имеет структуру:

```
for имя in список значений
      do
        список команд
      done
```

где "for" - служебное слово определяющее тип цикла, "do" и "done" - служебные слова, выделяющие тело цикла. Не забывайте про "done"! Фрагмент "in список значений" может отсутствовать.

Пример работы с оператором done:

```
$ a=1
$ b=2
$ c=3
$ for i in $a $b $c
> do
> echo "I love you!"
> done
I love you!
I love you!
I love you!
```

Т.о. мы видим, что переменная i в цикле пробегает по всем значением, заданным в условии (т.е. по очереди принимает значения всех переменных, указанных в условии), и для каждого из них выполняет ту операцию, которая заказана в теле цикла.

Оператор цикла с истинным условием ("while")

Структура "while", также обеспечивающая выполнение расчетов, предпочтительнее тогда, когда неизвестен заранее точный список значений параметров или этот список должен быть получен в результате вычислений в цикле.

Оператор цикла "while" имеет структуру:

```
while условие
      do
         список команд
      done
```

где "while" - служебное слово определяющее тип цикла с истинным условием. Список команд в теле цикла (между "do" и "done") повторяется до тех пор, пока сохраняется истинность условия (т.е. код завершения последней команды в теле цикла равен "0") или цикл не будет прерван изнутри специальными командами ("break", "continue" или "exit"). При первом входе в цикл условие должно выполняться.

```
$ a=1
$ b=5
$ while [ $a -lt $b]
> do
> a=`expr $a +1`
> echo $a
> done
```

a=`expr $a + 1`

т.е. при каждой итерации значение "a" увеличивается на 1.

Оператор цикла с ложным условием ("until")

Оператор цикла "until" имеет структуру:

```
until условие
      do
         список команд
      done
```

где "until" - служебное слово определяющее тип цикла с ложным условием. Список команд в теле цикла (между "do" и "done") повторяется до тех пор, пока сохраняется ложность условия или цикл не будет прерван изнутри

специальными командами ("break", "continue" или "exit"). При первом входе в цикл условие не должно выполняться.

Отличие от оператора "while" состоит в том, что условие цикла проверяется на ложность (на ненулевой код завершения последней команды тела цикла) проверяется ПОСЛЕ каждого (в том числе и первого!) выполнения команд тела цикла.

Примеры.

```
$ until false
> do
> read x
> if [$x=5]; echo $0
> then echo enough; break
> else echo some more
> fi
> done
```

Здесь программа с бесконечным циклом ждет ввода слов (повторяя на экране фразу "some more"), пока не будет введено "5". После этого выдается "enough" и команда "break" прекращает выполнение цикла.

12.3. Задание

1. Разработайте текст процедуры с использованием по заданию (см. ниже), вариант задания назначается преподавателем.

2. Отладьте, при необходимости отредактируйте и выполните процедуру.

3. Оформите процедуру с использованием вспомогательных команд и комментариев так, чтобы она легко читалась и чтобы результаты её работы легко анализировались.

12.4. Контрольные вопросы.

1. Что такое shell-процедура? Назначение?

2. Какого типа команды могут быть включены в тело процедуры?

3. Чем отличается обработка процедуры при выполнении от обработки программы на языке высокого уровня?

4. Что такое параметры? Для каких целей они используются? Какое число параметров может быть передано процедуре?

5. Какие вспомогательные команды вы использовали при оформлении процедуры?

6. Какого вида значения и как могут быть присвоены переменным языка shell?

7. Что такое локальные переменные и для каких целей их надо экспортировать в среду?

8. Как осуществляется ветвление вычислительного процесса процедуры?

9. Какого типа цикла в процедурах могут быть построены средствами языка shell?

10. Какие способы вызова процедур на исполнение вы знаете?

12.5. Варианты заданий к лабораторной работе № 12 «Введение в shell-программирование»

Разработать shell-процедуру с комментариями, выполняющую ниже перечисленные функции.

1. Вводит последовательность из N слов и подсчитывает в каждом введенном слове число символов. Если число символов больше M, то слово выводится на экран. Значения N и M передаются в качестве параметров.
2. Вводит строку из заданного числа слов. Выделяет слова, начинающиеся на указанную параметром букву, подсчитывает число таких слов.
3. Вводит строку N слов, анализирует длину каждого слова, упорядочивает слова по их алфавиту и выводит список на экран. Значение N задается параметром.
4. Вводит заданное параметром число слов и выводит каждое слово на печать, сопровождая его порядковым номером.
5. Вводит произвольное число коротких символьных параметров, подсчитывает длину каждого из них и выводит на экран список значений длин и общее число введенных параметров.
6. Вводит несколько коротких чисел в виде параметров, подсчитывает их сумму и результат выводит на экран.
7. Запрашивает последовательно ввод нескольких чисел со знаками и выводит на экран два списка чисел – положительных и отрицательных.

8. Запрашивает ввод строки символов, разделенных пробелами и заданной параметром длины, разбивает символы на пересекающиеся пары и выводит их на экран.
9. Ищет в личном головном каталоге пользователя созданные им файлы, выводит список их имен и распечатывается текст файла, заданного пользователем.
10. Создает новый подкаталог и помещает туда новые файлы, создаваемые пользователем по запросам процедуры. Имена новых файлов указываются параметрами.
11. Создает новый подкаталог и копирует туда из родительского каталога файлы заданного параметром типа.
12. Анализирует указанный параметром каталога и выводит на экран число файлов различного типа (обычные, директория, скрытые). Тип задается параметром.

13. Лабораторная работа №7 Программирование shell-процедур.

Работа предусматривается выполнение индивидуального задания повышенной сложности по варианту согласно номеру в журнале. Работа предусматривает несколько самостоятельных выходов на машину для отладки процедуры.

Отчётом по работе является работающая процедура, продемонстрированная преподавателю с объяснениями её текста и алгоритма работы. Выполнение лабораторной работы в полном объеме является обязательным условием для получения по курсу в целом экзаменационной оценки – «отлично».

13.1. Порядок выполнения работы

- Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу
- Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01
- Пока идёт импорт (около 15 минут) ознакомьтесь с индивидуальным заданием
- Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.
- Выполните все действия, описанные в теоретической части (для тренировки).
- Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.
- Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.
- Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.
- Экспортируйте образ сервера на свой носитель или в своё облако для последующей работы.

13.2. Индивидуальные задания к лабораторной работе

Вариант 1.

Написать shell-процедуру, которая:

- вводит передаваемое в качестве 1-го параметра количество символьных строк;
- в каждой введенной строке ищет подстроку, передаваемую в качестве второго параметра;
- заменяет каждую найденную подстроку на строку, передаваемую в качестве третьего параметра;
- выводит на экран каждую введенную строку и соответствующую ей новую строку.

Вариант 2.

Написать shell-процедуру, которая:

- вводит 2 символьные строки;
- в каждой введенной строке ищет подстроку, передаваемую в качестве параметра;
- заменяет каждую найденную подстроку на пробел;
- образует из полученных строк третью строку так, чтобы в ней чередовались слова из первой и второй строк;
- выводит на экран введенные строки и новую строку.

Вариант 3.

Написать shell-процедуру, которая:

- вводит символьную строку;
- во введенной строке ищет подстроку, передаваемую в качестве первого параметра;
- вставляет после каждой найденной подстроки символ, передаваемый в качестве второго параметра;
- удаляет из полученной строки символ, передаваемый в качестве третьего параметра;
- выводит на экран введенную и новую строку.

Вариант 4.

Написать shell-процедуру, которая:

- вводит символьную строку;
- проверяет введенную строку на совпадение со строкой, переданной в качестве 1-го параметра;
- если строки совпадают, то выдает на экран приглашение повторить ввод;
- если не совпадают, то сравнивает длину введенной строки с длиной 2-го параметра, и, в случае их равенства, выводит на экран введенную строку в обратном порядке составляющих ее символов;

Вариант 5.

Написать shell-процедуру, которая:

- вводит символьную строку;
- проверяет введенную строку на совпадение со строками, содержащимися в файле, имя которого передается в качестве 1-го параметра;
- для всех найденных совпадений заменяет соответствующие строки в файле на строку, переданную в качестве 2-го параметра;
- выводит на экран старое и новое содержимое файла, а также число найденных совпадений;

Вариант 6.

Написать shell-процедуру, которая:

- вводит символьную строку, содержащую маршрутное имя некоторого файла; проверяет введенное маршрутное имя, если оно начинается с символа /, на совпадение его первой части с маршрутным именем домашнего каталога пользователя;
- если введенное маршрутное имя содержит маршрутное имя домашнего каталога или является относительным, то проверяет существование указанного файла, в противном случае выводит на экран сообщение об ошибке;
- если файл существует, то выводит на экран его содержимое;

- если файл не существует, то создает его и записывает в него строку, передаваемую в качестве параметра;

Вариант 7.

Написать shell-процедуру, которая:

- вводит символьную строку, содержащую имя некоторого файла;
- проверяет наличие файла в домашнем каталоге или в одном из подкаталогов;
- если файл существует, то выводит на экран его содержимое;
- если файл не существует, то создает его и записывает в него с консоли некоторый текст;
- устанавливает для файла права доступа, передаваемые в качестве параметра.

Вариант 8.

Написать shell-процедуру, которая:

- Читает содержимое первого файла, передаваемого в качестве первого параметра;
- Читает содержимое второго файла, передаваемого в качестве второго параметра;
- Находит в первом файле строку, содержащую заданное третьим параметром слово;
- Вставляет содержимое второго файла после найденной строки первого файла;

Выводит на экран содержимое полученного файла

Вариант 9.

Написать shell-процедуру, которая:

- Читает содержимое первого файла, передаваемого в качестве первого параметра;
- Читает содержимое второго файла, передаваемого в качестве второго параметра;
- Если число строк в первом и втором файлах одинаковое, то выводит на экран каждые 5 секунд попеременно строки из первого и второго файлов

Вариант 10.

Написать shell-процедуру, которая:

- Читает содержимое первого файла, передаваемого в качестве первого параметра;
- Читает содержимое второго файла, передаваемого в качестве второго параметра;

- Выводит на экран каждые 7 секунд попеременно 2 строки из первого и 1 строку из второго файла, перемещаясь по файлам циклически

Вариант 11.

Написать shell-процедуру, которая:

- удаляет из заданного первым параметром каталога и всех подкаталогов файлы, дата последней модификации которых предшествует текущей дате минус число дней, переданное в качестве второго параметра;
- изменяет дату последней модификации всех остальных файлов указанного каталога на текущую без изменения содержимого файлов.

Вариант 12.

Написать shell-процедуру, которая:

- выводит на экран список всех пользователей системы, включенных в заданную первым параметром группу пользователей;
- для каждого из заданных третьим и следующими параметрами имен пользователей выводит на экран права доступа к заданному вторым параметром файлу.

Вариант 13.

Написать shell-процедуру, которая:

- Вводит символьную строку, содержащую некоторое целое число;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые 6 секунд попеременно 2 строки из файла и 1 введенную строку, перемещаясь по файлу циклически

Вариант 14.

Написать shell-процедуру, которая:

- Вводит символьную строку;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые 6 секунд попеременно 2 строки из файла и 1 введенную строку, перемещаясь по файлу циклически

Вариант 15.

Написать shell-процедуру, которая:

- Вводит символьную строку;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые 6 секунд попеременно 2 строки из файла и 1 введенную строку, перемещаясь по файлу циклически

Вариант 16.

Написать shell-процедуру, которая:

- Вводит символьную строку, содержащую два целых числа m и n, разделенных пробелами;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые 5 секунд попеременно m строк из файла и n строк "Будь здоров"

Вариант 17.

Написать shell-процедуру, которая:

- Вводит символьную строку, содержащую два целых числа m и n, разделенных пробелами;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые m секунд попеременно n строк из файла и пустую строку

Вариант 18.

Написать shell-процедуру, которая:

- Вводит символьную строку, содержащую три целых числа k,m и n, разделенных пробелами;
- Читает содержимое файла, передаваемого в качестве первого параметра;
- Выводит на экран каждые m секунд попеременно m строк из файла и n пустых строк

Вариант 19.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Записывает через каждые 5 секунд в файл, имя которого передается в качестве второго параметра, попеременно строки из первого файла и текущее время и дату;
- Выводит на экран каждые 7 секунд текущее содержимое второго файла;
- При вводе с клавиатуры слова quit удаляет второй файл и завершает работу

Вариант 20.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Создает новый файл, имя которого передается в качестве второго параметра;
- Выводит на экран каждые 7 секунд очередную строку первого файла;
- Сортирует все выведенные на экран строки первого файла по длине и записывает их в новый файл;
- При вводе с клавиатуры слова quit удаляет новый файл и завершает работу

Вариант 21.

Написать shell-процедуру, которая:

- Читает содержимое файла, имя которого вводится при исполнении процедуры;
- Создает новый файл, имя которого передается в качестве параметра;
- Выводит на экран каждые 6 секунд очередные 2 строки файла;
- Сортирует выведенные на экран строки по длине и записывает их в новый файл;
- При вводе с клавиатуры слова end удаляет второй файл и завершает работу

Вариант 22.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Создает новый файл, имя которого передается в качестве второго параметра;

- Записывает в новый файл строки первого файла в обратном порядке, вставляя после каждого слова фразу "THAT IS ALL" столько раз , сколько задано третьим параметром

Вариант 23.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Создает в текущем каталоге цепочку подкаталогов с относительным маршрутным именем, повторяющим полное маршрутное имя текущего каталога;
- Создает в последнем подкаталоге новый файл, имя которого передается в качестве второго параметра;
- Записывает в новый файл строки первого файла в обратном порядке, вставляя после каждого слова фразу "THAT IS ALL" столько раз, сколько задано третьим параметром

Вариант 24.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра;
- Создает в текущем каталоге цепочку подкаталогов с относительным маршрутным именем, повторяющим полное маршрутное имя текущего каталога;
- Создает в последнем подкаталоге новый файл, имя которого передается в качестве второго параметра;
- Записывает в новый файл строки первого файла в обратном порядке, вставляя после каждого слова фразу "THAT IS ALL" столько раз, сколько задано третьим параметром

Вариант 25.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра (в файле в каждой строке представлено одно целое число);

- Проверяет правильность формата содержимого файла, при ошибках выводит соответствующие сообщения и завершает работу;
- Подсчитывает сумму всех содержащихся в файле чисел;
- Выводит на экран полученную сумму

Вариант 26.

Написать shell-процедуру, которая:

- Читает содержимое файла, передаваемого в качестве первого параметра (в файле в каждой строке представлено по два целых числа, разделенных пробелами);
- Проверяет правильность формата содержимого файла, при ошибках выводит соответствующие сообщения и завершает работу;
- Подсчитывает сумму чисел в каждой строке файла;
- Сортирует полученные суммы по убыванию и выводит их на экран

Вариант 27.

Написать shell-процедуру, которая:

- Читает содержимое двух файлов, передаваемых в качестве первых параметров (в файлах в каждой строке представлено по одному целому числу);
- Проверяет правильность формата содержимого файлов, при ошибках выводит соответствующие сообщения и завершает работу;
- Выбирает одинаковые числа в первом и втором файлах;
- Сортирует полученные числа по возрастанию и выводит их на экран

Вариант 28.

Написать shell-процедуру, которая:

- Читает содержимое двух файлов, передаваемых в качестве первых параметров (в файлах в каждой строке представлено по одному целому числу);
- Проверяет правильность формата содержимого файлов, при ошибках выводит соответствующие сообщения и завершает работу;
- Суммирует числа из первого и второго файлов, расположенные в строках с одинаковым номером;

- Сортирует полученные суммы по возрастанию и выводит их на экран

Вариант 29.

Написать shell-процедуру, которая:

- Читает содержимое двух файлов, передаваемых в качестве первых параметров (в файлах в каждой строке представлено не более, чем по три целых числа, разделенных пробелами);
- Проверяет правильность формата содержимого файлов, при ошибках выводит соответствующие сообщения и завершает работу;
- Вычисляет суммы чисел в каждой строке;
- Сортирует полученные положительные суммы по возрастанию и выводит их на экран

Вариант 30.

Написать shell-процедуру, которая:

- Читает содержимое трех файлов, передаваемых в качестве первых параметров (в файлах в каждой строке представлено не более, чем по четыре целых числа, разделенных пробелами);
- Проверяет правильность формата содержимого файлов, при ошибках выводит соответствующие сообщения и завершает работу;
- Вычисляет произведения чисел в каждой строке;
- Сортирует полученные произведения по возрастанию и выводит их на экран

Список использованной литературы

1. Таненбаум Э. Современные операционные системы. – Питер, 2015
2. Таненбаум А. Компьютерные сети. – Питер, 2006
3. Цикритзис Д., Бернстайн Ф. Операционные системы. – М.: Мир, 1977
4. Кальп Б. Администрирование Windows Vista. – БХВ, 2008
5. Драуби О., Моримото Р., Ноэл М., Амарис К., Мистри Р. Microsoft Windows Server 2008. – Вильямс, 2008
6. Майерс С. MacOS X 10.5 Leopard. – БХВ, 2008
7. Баррет Д. Дж. Linux: Основные команды. – Кудиц-Пресс, 2008