

Г. Н. ИСАЕВА
Н. П. СИДОРОВА

**ОПЕРАЦИОННЫЕ
СИСТЕМЫ,
СРЕДЫ И ОБОЛОЧКИ
ПРАКТИКУМ**

Учебное пособие



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
имени дважды Героя Советского Союза, летчика-космонавта А.А. Леонова

**Г. Н. ИСАЕВА
Н. П. СИДОРОВА**

ОПЕРАЦИОННЫЕ СИСТЕМЫ, СРЕДЫ И ОБОЛОЧКИ

ПРАКТИКУМ

Учебное пособие



**Москва
2022**

УДК 004.451
ББК 32.972.111я73
И85

Рецензенты:

Стреналюк Ю. В., д-р тех. наук, профессор кафедры информационных технологий и управляющих систем;
Логачёва Н. В., канд. тех. наук, доцент кафедры информационных технологий и управляющих систем

Исаева, Г. Н.

И85 Операционные системы, среды и оболочки. Практикум : учебное пособие / Г. Н. Исаева, Н. П. Сидорова. — Москва : Директ-Медиа, 2022. — 51 с.

ISBN 978-5-4499-3324-9

Учебное пособие предназначено для ознакомления с теоретическими и практическими основами работы в двух наиболее популярных в настоящее время линейках операционных систем (ОС): MS Windows и Linux.

Пособие включает восемь практических работ, раскрывающих основные функции ОС по управлению вычислениями и эффективному распределению ресурсов в вычислительной системе (ВС).

Пособие составлено в соответствии с требованиями федерального государственного образовательного стандарта высшего образования (ФГОС ВО) для обучающихся по направлению бакалавриата 09.03.03 «Прикладная информатика», 09.03.02 «Информационные системы и технологии», 01.03.02 «Прикладная математика и информатика», а также может быть использовано при подготовке студентов других направлений бакалавров, таких как 10.03.01 «Информационная безопасность», 27.03.04 «Управление в технических системах» при изучении дисциплин, связанных с работой основного программного компонента ВС – операционной системы.

УДК 004.451
ББК 32.972.111я73

ISBN 978-5-4499-3324-9 © Исаева Г. Н., Сидорова Н. П., текст, 2022
© Издательство «Директ-Медиа», оформление, 2022

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ.....	5
ВВЕДЕНИЕ.....	6
Практическая работа 1. УСТАНОВКА ГОСТЕВЫХ ОС С ПОМОЩЬЮ ORACLE VM VIRTUAL BOX.....	8
1. Основные понятия.....	8
2. VirtualBox.....	9
3. Методические указания.....	10
4. Последовательность выполнения практической работы.....	11
5. Требования к оформлению отчёта.....	13
Контрольные вопросы.....	13
Практическая работа 2. ПОЛУЧЕНИЕ СВЕДЕНИЙ О ПАРАМЕТРАХ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ В ОС MICROSOFT WINDOWS.....	15
1. Основные понятия.....	15
2. Методические указания.....	16
3. Последовательность выполнения практической работы.....	16
4. Требования к оформлению отчёта.....	18
Контрольные вопросы.....	18
Практическая работа 3. ФАЙЛОВАЯ СИСТЕМА В ОС MICROSOFT WINDOWS.....	19
1. Основные понятия.....	19
2. Методические указания.....	20
3. Последовательность выполнения практической работы.....	21
4. Требования к оформлению отчёта.....	23
Контрольные вопросы.....	23
Практическая работа №4 ОСНОВЫ РАЗРАБОТКИ ПАКЕТНЫХ ФАЙЛОВ В ОС MICROSOFT WINDOWS.....	24
1. Основные понятия.....	24
2. Методические указания.....	26
3. Последовательность выполнения практической работы.....	26
4. Требования к оформлению отчёта.....	28
Контрольные вопросы.....	28
Практическая работа №5 НАЧАЛЬНОЕ ЗНАКОМСТВО С ОС ЛИНЕЙКИ LINUX.....	30
1. Основные положения.....	30
2. Методические указания.....	31
3. Последовательность выполнения практической работы.....	33
4. Требования к оформлению отчёта.....	34
Контрольные вопросы.....	34
Практическая работа №6 ОСОБЕННОСТИ ОРГАНИЗАЦИИ СТРУКТУРЫ ДАННЫХ В ОС LINUX.....	35
1. Основные понятия.....	35
2. Методические указания.....	36

3. Последовательность выполнения практической работы	36
4. Требования к оформлению отчёта	37
Контрольные вопросы	38
Практическая работа №7 УПРАВЛЕНИЕ ПРОЦЕССАМИ В ОС LINUX	39
1. Основные понятия.....	39
2. Методические указания.....	40
3. Последовательность выполнения практической работы	42
4. Требования к оформлению отчёта	43
Контрольные вопросы	43
Практическая работа №8 ОСНОВЫ ПРОГРАММИРОВАНИЯ В КОМАНДНОЙ СРЕДЕ В ОС LINUX	44
1. Основные понятия.....	44
2. Методические указания.....	45
3. Последовательность выполнения практической работы	45
4. Требования к оформлению отчёта	47
Контрольные вопросы	48
ЛИТЕРАТУРА	49
ПРИЛОЖЕНИЕ 1	50

СПИСОК СОКРАЩЕНИЙ

АС	аппаратные средства
ВС	вычислительная система
ПК	персональный компьютер
ПС	программные средства
ПО	программное обеспечения
ОЗУ	оперативное запоминающее устройство
ОС	операционная система
ФС	файловая система
ЦП	центральный процессор
ЭВМ	электронно-вычислительная машина
HTTP	Hyper Text Transfer Protocol – протокол передачи гипертекста
FTP	File Transfer Protocol – протокол передачи файлов
PID	Process Identifier – Идентификатор процесса
VM	Virtual Machine – виртуальная машина

ВВЕДЕНИЕ

Согласно компетенциям, которые должны освоить обучаемые в рамках курса «Операционные системы, среды и оболочки», они должны хорошо представлять себе работу приложений в изменяющихся условиях функционирования, которые, с одной стороны, определяются совершенствованием и развитием аппаратных средств (АС) ВС, а с другой – работают под управлением главного программного компонента – операционной системы. Поэтому знание материала, касающегося системного ПО – определяющегося функционированием ОС и тех сред, которые она поддерживает для выполнения пользовательских приложений, является ключевым аспектом данного пособия.

Теория операционных систем начинает свой отсчёт с середины прошлого столетия и насчитывает на сегодняшний день сотни версий и концепций построения этого комплекса программ, обслуживающего ВС. Разобраться в этом потоке информации не просто, поэтому целью данного пособия является донести до обучаемых характерные черты и функции двух популярных в нашей стране и признанных мировым компьютерным сообществом линеек ОС: Microsoft System Windows и ОС Linux; и на их примере показать, как функционируют и взаимодействуют основные модули операционной системы. Кроме того, на наш взгляд, более правильные знания формируют у обучаемых умение самому понять, какие команды выполняет ОС и с каким набором параметров, что при работе с графическим интерфейсом не совсем ясно, поэтому практические работы связаны и с освоением команд Интерпретатора командной строки (Command line interpretation – cmd.exe), и с написанием сценариев, позволяющих понять логику и ход операций ОС.

В настоящее время нет прочно устоявшейся терминологии в теории операционных систем, поэтому в качестве базовых, было выбрано несколько фундаментальных работ в этой сфере, и трактовка основных определений и названий соответствует изложенным базовым понятиям по управлению вычислениями, распределению ресурсов ВС, управлению вводом-выводом, рассмотренным в данных учебниках, рекомендуемых для высшей школы [2], [5-6], [9-10].

Практикум включает необходимые краткие начальные сведения по основным понятиям курса «Операционные системы, среды и оболочки» и восемь практических работ, половина из которых ориентирована на изучение основных команд и функций Windows-подобных ОС, другая – на освоение возможностей Linux-подобных ОС.

Для выполнения практических работ по данному курсу желательно иметь специально выделенный парк персональных компьютеров (ПК), с установленными операционными системами из ряда перечисленных выше и соответствующими правами. Понятно, что не каждое учебное заведение

может позволить себе такую организацию учебного процесса по курсу «Операционные системы, среды и оболочки». Поэтому давно используется другой подход – устанавливать гостевые операционные системы с правами администратора на виртуальных компьютерах, генерируемых с помощью платформы виртуализации – виртуальной машины [3]. В качестве доступного ПО можно использовать свободно распространяемую виртуальную машину – Oracle VM VirtualBox.

Учебное пособие предназначено для студентов, обучающихся по направлению «Прикладная информатика», «Информационные системы и технологии», и может быть полезно студентам других направлений подготовки при изучении дисциплин, связанных с ОС, операционными средами и разработкой приложений для различных платформ.

Практическая работа 1. УСТАНОВКА ГОСТЕВЫХ ОС С ПОМОЩЬЮ ORACLE VM VIRTUAL BOX

ЦЕЛЬ РАБОТЫ: освоить установку различных гостевых ОС с помощью платформы виртуализации.

1. Основные понятия

Не зависимо от того, в каком устройстве находится комплекс программного обеспечения – его принято классифицировать по четырём уровням – *базовое, системное, служебное и прикладное*. Такая иерархия отражает суть программных средств, необходимых для работы ВС, с одной стороны, и выполнения вычислительной системой основных своих функциональных задач – с другой. Каждый нижний уровень программного обеспечения является базовым для построения и функционирования следующего, более высокого уровня ПО (Рис 1) [9].

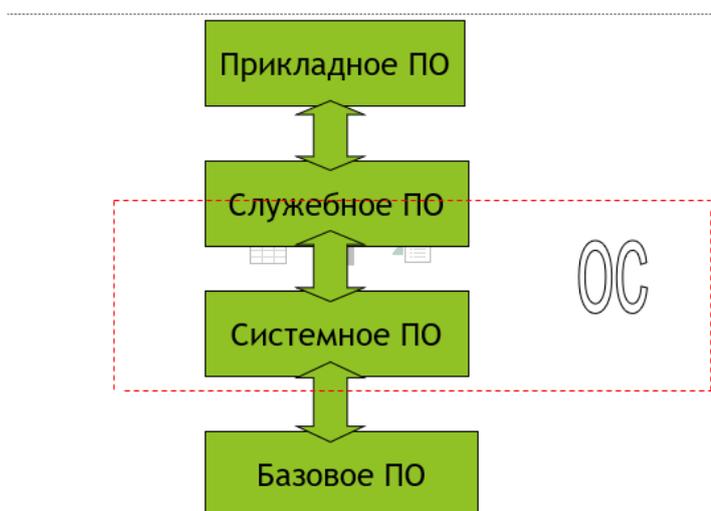


Рисунок 1 – Структура ПО ПК

Кроме того, при объединении всех уровней ПО ВС, за исключением базового, и рассмотрении их в одной плоскости, как software (англ. Программное обеспечение), можно выделить три группы – по критерию назначения программного обеспечения: *системное ПО, прикладное ПО и инструментальное ПО* (Рис 2).



Рисунок 2 – Схема распределения по видам ПО

Операционная система относится к наиболее общей части программного обеспечения вычислительной системы – к системной, которая обеспечивает функционирование остальных видов ПО.

Из ряда определений ОС, наиболее полно отражает суть этого программного компонента следующее: *ОС – это комплекс взаимосвязанных программных модулей, обеспечивающий пользовательский интерфейс и определяющий эффективную организацию всех вычислений и эффективное распределение всех ресурсов в вычислительной системе* [6].

2. VirtualBox

Виртуализация в теории вычислительных систем бывает разная: операционных систем, прикладных программ, систем хранения данных, компонентов аппаратного и программного обеспечения.

Наиболее востребована в настоящее время виртуализация аппаратных средств ВС с целью создания на реальной ЭВМ дополнительных аппаратных конфигураций (эмуляторов) и установка на них различных системных и прикладных программ. Силами современных виртуальных машин (Virtual Machine – VM) можно эмулировать работу процессора, ОЗУ, внешней памяти, сети и др. [3].

Основными примерами использования виртуальных машин являются такие как: проверка программных средств на

кроссплатформенность, тестирование опасного ПО, использование для обучения новым ОС. Для этого с помощью VM создают на реальном компьютере несколько виртуальных компьютеров с различными установленными на них операционными системами. Перспективным, бесплатным ПО в этой области VM является VirtualBox от Oracle – программное обеспечение, предназначенное для создания виртуальных машин.

Реальную операционную систему, установленную на ЭВМ, принято называть хостовой. В этой ОС и работает программная система – виртуальная машина, с помощью которой осуществляется генерация виртуального АО. И уже на эти аппаратные ресурсы устанавливается желаемая (или требуемая в ходе исследований) ОС. Эта операционная система называется гостевой. Понятно, что с помощью виртуальной машины можно создать несколько схем установки гостевых ОС на виртуальном компьютере. Самая простая – один виртуальный компьютер – одна гостевая ОС (Рис 3).



Рисунок 3 – Компоненты виртуализации

3. Методические указания

1. Установку гостевой Windows-подобной ОС следует производить, придерживаясь следующих основных шагов.

- Осуществить копирование требуемого образа ОС в свою директорию на локальный диск.
- Активировать приложение Oracle VM VirtualBox.
- Далее выбрать вкладку «создать».

- Указать имя виртуальной машины. Тип и версию.
- Объём оперативной памяти оставить 512Мб или изменить на 1Гб, в зависимости от типа загружаемого дистрибутива Windows-подобной ОС. Например, для Windows XP достаточно 512Мб, для Windows 10 – 1Гб.
- Создать новый виртуальный жесткий диск.
- Тип диска – VHD.
- Размер виртуального диска может составлять 15–20Гб. Здесь необходимо ориентироваться на ресурсы реальных АС ПК.
- Далее выбрать вкладку «создать».
- Выделить созданную виртуальную машину и выбрать кнопку «Настроить».
- Перейти во вкладку «Система» и снять «галочку» с гибкого диска.
- Перейти во вкладку «Носители» и выбрать в окне «Носители информации» диск с надписью «Пусто».
- В «Атрибутах» нажать на диск со стрелочкой.
- Далее нажать «Выбрать образ оптического диска» и выбрать iso-образ устанавливаемой гостевой операционной системы (скопированный образ ОС на начальном шаге).
- Закрывать окно настроек и загрузить ОС согласно правилам установки выбранной версии линейки операционной системы. Установить требуемые параметры системы, отвечая на вопросы в окнах загрузки. Рекомендуется пользоваться источником информации на официальном сайте от Microsoft [7].

2. Установку гостевой Linux-подобной ОС следует производить, придерживаясь таких же основных шагов, как и в п.1, но особенности загрузки данной ОС можно уточнить на официальном сайте [11]. Кроме того, следует помнить, что при задании пароля в Linux он не отображается никакими символами на консоли, но запомнить его нужно обязательно!

4. Последовательность выполнения практической работы

1. Установка гостевой ОС с помощью VM Virtual BOX.

Осуществить копирование требуемого образа ОС в свою директорию на локальный диск D:\ из папки ОС на сетевом ресурсе.

Активировать виртуальную машину VM Virtual BOX и согласно инструкции, приведённой в методических указаниях выше (3. Методические указания), произвести установку гостевой Windows-подобной ОС на сгенерированной с помощью VM платформе (виртуальном ПК).

Активировать виртуальную машину VM VirtualBox и согласно методическим указаниям развернуть гостевую Linux-подобную ОС (например, Ubuntu 16.4 как наиболее стабильную и проверенную версию ОС) на виртуальном ПК.

Рекомендуется сгенерировать два виртуальных компьютера с помощью VM VirtualBox. Результатом установки двух гостевых операционных систем на платформе виртуализации будет два независимых файла, с названиями пользовательских ОС указанных выше линеек, отображаемых в стартовом окне VM (Рис 4).



Рисунок 4 – Окно VM VirtualBox с гостевыми ОС

2. Проверьте работоспособность Windows-подобной ОС, для этого войдите в режим командной строки (cmd.exe) и выполните команду получения справки по названиям команд данного интерфейса (команда help).

Далее получите справку по одной из выбранных команд из списка (например dir). Выполните для этого Help dir или dir/?

Более развёрнутые начальные сведения о командном интерпретаторе и его командах получите, используя источники информации [5], [7-8].

3. Проверьте работоспособность Linux-подобной ОС, для этого войдите сначала в графический режим установленной ОС.

Осуществите сёрфинг по папкам и файлам. Ознакомьтесь с пиктограммами начальной страницы и ответьте на установочные вопросы. Затем проверьте режим терминала. Терминал – это имитация многопользовательского режима данной ОС. В каждом терминале параллельно может быть запущена своя задача, не только относящаяся к данному пользователю с выделенными правами, но и совершенно другому пользователю, например, с правами администратора системы.

Дополнительную информацию по архитектуре Linux-подобной ОС можно получить в ходе читаемых лекций и доступных на портале МГОТУ «Технологический университет» (ies.unitech-mo.ru) или в рекомендуемой литературе по изучению курса ОС [2], [5-6], [10], а также на сайтах [8], [11].

4. Оформить отчёт по практической работе.

5. Требования к оформлению отчёта

Отчёт по практической работе должен включать следующие разделы.

1. Титульный лист (Приложение 1).
2. Цель работы и постановку каждого выполняемого задания.
3. Скриншот стартового окна виртуальной машины с названиями файлов установленных гостевых ОС двух линеек: Windows-подобной и Linux-подобной ОС.
4. Скриншоты стартовых окон ОС, подтверждающие их работоспособность; можно привести скриншоты элементарных выполненных команд в системах.
5. Выводы по работе.
6. Ответы на контрольные вопросы.

Контрольные вопросы

1. Какие компоненты можно виртуализировать в современных ВС?
2. Что обозначает виртуализация ОС?
3. Перечислите достоинства и недостатки виртуализации.
4. Приведите список из 5-6 наименований виртуальных машин и дайте краткую характеристику каждой по критериям: стоимость, возможности, год выпуска.

5. Кратко охарактеризуйте установленные гостевые ОС: к какому классу они относятся, с какой архитектурой, на какие АС могут быть установлены.

6. Что означает термин «авторизация» по отношению к ОС? Что означает термин «аутентификация»? Какая из этих операций выполняется раньше и почему?

7. Как переключиться из гостевой ОС в хостовую ОС?

8. Можно ли копировать файлы из гостевой ОС в хостовую ОС? Если «да», то каким образом?

Практическая работа 2. ПОЛУЧЕНИЕ СВЕДЕНИЙ О ПАРАМЕТРАХ ФУНКЦИОНИРОВАНИЯ СИСТЕМЫ В ОС MICROSOFT WINDOWS

ЦЕЛЬ РАБОТЫ: исследование возможности получения сведений о функциональности и параметрах системы, работающей под управлением ОС линейки Microsoft Windows.

1. Основные понятия

Основные программы ОС, которые позволят получить сведения об аппаратной и программной конфигурации ВС следующие.

Программа Сведения о системе – отображает данные о конфигурации системы, запуск её осуществляется, например, для Windows 10 следующими способами:

- 1) «Пуск» → «Выполнить» и вызвать файл msinfo32;
- 2) «Пуск» → «Средства администрирования» → «Сведения о системе».

Программа Диспетчер задач позволяет осуществить мониторинг за активностью процессов и задач в ОС, посмотреть журналы приложений, и т.д., вызвать её можно комбинацией клавиш «Ctrl+Shift+Esc».

Программа Системный монитор – компонент, позволяющий осуществлять оценку производительности локального компьютера и устанавливать соединение с объектом, выполняющимся на этом компьютере, а также отображать данные с него. Вызвать эту программу можно с помощью perfmon в «Пуск» → «Выполнить» или найти и запустить файл perfmon.exe в папке C:\Windows\System32.

Кроме того, имеется и набор других важных программ, которые при необходимости пользователь может получить во вкладках из списка, выпадающего при активации «Пуск».

Например, программа Командный интерпретатор, работающая на протяжении всего сеанса работы ОС и позволяющая пользователю в режиме командной строки осуществлять диалог с системой на языке, который понимает командный интерпретатор. Получить доступ к данной программе в режиме командной строки можно следующим образом:

- 1) «Пуск» → «Служебные – Windows» → «Командная строка»;
- 2) Активировав программу cmd.exe.

2. Методические указания

1. При установке гостевых ОС Windows 10 и Windows 7 порядок вызова вкладок и внешний вид вызываемых соответствующих окон может отличаться.

2. При работе с командным интерпретатором пользователь вызывает командную строку и осуществляет набор команды после введённого приглашения и получает ответ от системы в том же окне на экране монитора, либо в виде активации других окон. В качестве основного устройства взаимодействия пользователя в данном режиме с системой выступает *терминал* (или консоль, как принято называть окно ввода-вывода в программных системах, работающих под Windows). Физически терминал представляет собой монитор и клавиатуру. Логически это два файла – ввода и вывода. В некоторых ОС это один файл. Так в Windows-подобных системах файл консоли (или логическое имя терминала) называется CON.

3. Изучаемые команды:

Getmac, Chkdsk, Cmd, Ipconfig, Powercfg, Prompt, Set, Sfc, Shutdown, Systeminfo, Tasklist, Tracert.

4. Команда *set* формирует символьные переменные окружения в резервируемой памяти ОС, доступной для команд ОС и программ пользователей. Следует помнить, что имя создаваемой переменной не должно совпадать с ключевыми резервируемыми словами.

3. Последовательность выполнения практической работы

1. Проведите мониторинг основных показателей работы ОС линейки Windows (установленной гостевой Windows 10 или Windows 7). Активируйте программу «Сведения о системе» и ознакомьтесь с её возможностями. Получите общие сведения о системе, данные об адаптерах сети, связанные с сетевыми подключениями, и другие данные, на Ваш взгляд, важные для понимания работы ОС.

2. Активируйте программу «Диспетчер задач» и ознакомьтесь с её возможностями. Зафиксируйте в отчёте моменты перепланировки задач и загруженность ЦП в этих случаях.

3. Активируйте программу «Системный монитор» и выполните операции со счётчиками, добавив их с помощью панели инструментов «Добавить». Просмотрите в окне список объектов и список счётчиков. Например, для объекта «Сетевой интерфейс» или другого сетевого объекта, подключите счётчик сетевой активности каналов передачи данных (счётчик «Всего Байт в секунду»). Проверьте работу счётчика, осуществив выход в сеть (начните взаимодействие с знакомым сайтом). Как изменится диаграмма счётчиков?

4. Для программы «Системный монитор» создайте и проследите ещё за двумя дополнительными счётчиками. Например, откройте папку «Группы сборщиков данных». Во вкладке «Особый» создайте группу сборщиков данных. В контекстном меню выберите «Создать: группа сборщиков данных». Выполните создание группы двумя способами – из шаблона и вручную. Затем приступите к добавлению счётчиков для группы сборщиков данных, созданных Вами. В свойствах первого счётчика выберите двоичный тип журнала (позже для другого счётчика можно будет выбрать другой тип журнала), а в дереве контекстного меню выберите папку, в которую будет собираться информация. Активируйте группу с помощью вкладки в верхней части окна. Найдите папку и файл с записанными протоколами и оцените его.

5. Приступите к работе со вторым счётчиком и другим типом журнала. Если сохранить шаблон группы сборщиков данных (правая кнопка мыши), то его можно будет использовать для новых групп сборщиков данных.

6. Активируйте программу «Командная строка». Опираясь на учебные материалы, рекомендованные в пособии и лекции по курсу [5], [7], [10], выберите и проверьте в командном режиме команды, обеспечивающие информацию о системе. Приведите пример их выполнения (3-4 команды).

7. Проверьте работу команды *prompt*. Включите в строку приглашения сообщение о текущей дате. Включите в строку приглашения текст «Привет, <собственное имя>».

8. Выполните команду *set*. Приведите её ключи и атрибуты.

9. Исследуйте команду: `set TMP=D:\GARBAGE`. Что будет выполнено в системном реестре по этой команде?

10. Покажите, как команда `set` может являться обработчиком формул. Приведите пример 2-3-х арифметико-логических выражений.

11. Исследуйте команду управления системой: просмотр и изменение размера файла подкачки.

12. Как получить такие характеристики ВС, как размер физической и виртуальной памяти с помощью командного интерпретатора?

4. Требования к оформлению отчёта

Отчёт по практической работе должен включать следующие разделы.

1. Титульный лист (Приложение 1).
2. Цель работы и постановку каждого выполняемого задания.
3. Скриншоты работы с окнами основных Программ по отслеживанию параметров и конфигурации ВС.
4. Скриншоты и ответы на вопросы по результатам исследования команд ОС в режиме Командной строки.
5. Фрагменты текстовых файлов (где необходимо), в которых фиксируется результат выполнения перенаправления ввода/вывода команды или её результат.
6. Выводы по работе.
7. Ответы на контрольные вопросы.

Контрольные вопросы

1. Опишите функционал клавиш «стрелка вверх», «стрелка вниз», «стрелка вправо/влево» дополнительной клавиатуры.
2. В чём преимущества режима командной строки по сравнению с графическим режимом ОС? Какие аспекты характеризуют недостатки данного режима?
3. Для каких целей используется команда `set`? Перечислите все случаи.
4. Как проследить динамику изменения текущих размеров физической и виртуальной памяти.
5. Какие команды в ОС линейки MS Windows наиболее подходят для сетевого администрирования.

Практическая работа 3. ФАЙЛОВАЯ СИСТЕМА В ОС MICROSOFT WINDOWS

ЦЕЛЬ РАБОТЫ: исследовать функции ОС линейки Microsoft Windows по управлению файлами и устройствами в режиме командной строки.

1. Основные понятия

Файловая система (ФС – File System) определяет и контролирует, как будут храниться и именоваться данные на внешнем носителе/накопителе информации: флеш-накопителе, жестком или ssd диске и других. От ФС зависит способ хранения данных на накопителе, сам формат данных и то, как они будут записываться/читаться в дальнейшем.

Выделяют следующие функции *ОС по управлению файлами и устройствами:*

- организация параллельной работы устройств ввода-вывода и процессора;
- согласование скоростей обмена и кэширование данных;
- разделение устройств и данных между процессами;
- предоставление удобного программного интерфейса к устройствам;
- поддержка широкого спектра драйверов с возможностью простого включения в систему нового драйвера;
- динамическая загрузка и выгрузка драйверов;
- поддержка нескольких файловых систем;
- поддержка синхронных и асинхронных операций ввода-вывода [2].

Для линейки Windows и других ОС в настоящее время стандартом является ФС NTFS – New Technology File System, она поддерживается практически всеми устройствами и не имеет лимита на размер файлов в 4 Гб. Была разработана на смену FAT (File Allocation Table), обладает более высокой производительностью, защитой, механизмом хранения информации. Данные располагаются в главной таблице MFT – Master File Table. Файлы можно именовать на любом языке в стандарте юникода UTF – Unicode transformation format.

Несмотря на сложность подсистемы ввода-вывода в любой ОС, система представляет для пользователя упрощённый механизм взаимодействия с хранимыми данными. В этих целях ОС подменяет физическую структуру хранящихся данных некоторой удобной для пользователя логической моделью: иерархически организованных каталогов и файлов.

Другой задачей, возлагаемой на ФС, является организация ввода-вывода с устройствами. Каждому устройству ввода-вывода ставится в соответствие отдельный специальный файл, как было упомянуто в предыдущей практической работе по отношению к монитору и клавиатуре. Этот файл с конкретным именем и используется пользователями и системой для операций с устройством, вместо создания сложных процедур и запросов ввода-вывода для физических устройств. Привлекательность модели файла-устройства состоит в её простоте и универсальности для устройств любого типа [6].

2. Методические указания

1. При выполнении операций с данными, выводимыми на консоль или терминал используются буферы ввода-вывода – промежуточные хранилища данных в ОЗУ, где собираются данные перед непосредственной выдачей на монитор или ввода с клавиатуры. Согласно концепции управления файлами и устройствами в ФС, доступ к этим буферам и работа с ними аналогична доступу к файлам. По умолчанию ФС предоставляет доступ к трём виртуальным файлам, называемых потоками ввода-вывода (в Windows-подобных ОС за этими стандартными файлами закреплены стандартные *файловые дескрипторы* n , где $n=\{1,2,3\}$):

stdout – стандартный поток вывода – для cmd Windows дескриптор 1 – по умолчанию экран терминала (консоль);

stderr – стандартный поток вывода ошибок – для cmd Windows дескриптор 2 – по умолчанию экран терминала (консоль);

stdin – стандартный поток ввода – для cmd Windows дескриптор 0 – по умолчанию клавиатура.

Таким образом, если потоки данных в результате работы команд ОС большие или они нужны для дальнейшего анализа, то их целесообразно поместить во внешние файлы, используя при этом операцию перенаправления вводимых или выводимых данных

со стандартных устройств во внешние файлы, как описано в учебной литературе [2], [5-6], [10].

2. Современные файловые системы поддерживают различные типы файлов, среди них есть и такие специфические, как символьные связи, именованные конвейеры, отображаемые на память файлы. Существует такое понятие, как *конвейер ввода-вывода* – перенаправление выходного потока одной команды (или программы) в поток другой. В командной строке конвейер обозначается символом «|» и записывается как показано в примере на рисунке 5. В основу конвейера положен принцип декомпозиции заданий и операций, выполняемых как системными программными модулями ОС, так и приложениями, работающими под управлением ОС.

```
C:\Users\denna>dir | find "файлов"
      12 файлов                5 572 байт

C:\Users\denna>dir | find "байт"
      12 файлов                5 572 байт
      34 папок   30 719 856 640 байт свободно

C:\Users\denna>dir | find "папок"
      34 папок   30 719 840 256 байт свободно

C:\Users\denna>
```

Рисунок 5 – Пример конвейера ввода-вывода

3. Изучаемые команды:

Assoc, Attrib, Cd, Comp, Compact, Convert, Copy, Diskpart, Dir, Find, Format, Fsutil, Mkdir, Mklink, Subst, Tree.

3. Последовательность выполнения практической работы

1. Опираясь на учебные материалы, информационные справочные материалы по работе с ОС Windows 10 от фирмы Microsoft, выберите и проверьте в режиме командной строки команды, обеспечивающие информацию о ФС, о действиях и операциях по работе с файлами и каталогами.

2. Исследуйте команду *dir* с разными ключами.

3. Исследуйте команду *dir* в конвейере с командой *more, find, sort*. Получите данные, которые нельзя получить простой суперпозицией команды *dir* и выше упомянутых.

4. Создайте 2-3 каталога, удалите один из них. Поместите в другой 2 исполняемых файла и проследите путь к ним, записав информацию в отдельный файл.

5. Для команды смены каталога приведите наиболее короткую запись запроса в случае: а) перехода из каталога нижнего уровня в корневой каталог диска; в) перехода из дочернего каталога в родительский каталог.

6. В каких целях используется команда *subst*? Что она даёт пользователю? Продемонстрируйте на примере.

7. Используя команду *tree*, определите дерево текущего каталога. Есть ли ключи у данной команды, проверьте.

8. Команда *copy* – сложная команда. Кроме копирования файлов, данная команда позволяет выполнить иные действия с файлами и устройствами. Перечислите эти действия и ключи.

9. Как указать в запросе данной команды для копирования в имени файла произвольное сочетание символов? Произвольный одиночный символ? Приведите демонстрационные примеры.

10. Произведите копирование любого файла из выбранной директории на псевдоустройство.

11. Объедините три файла в один файл 1.doc в текущем каталоге текущего диска с проверкой записи. Если спецификация приёмника при копировании отсутствует, проверьте, какая будет спецификация у получившегося файла, если объединяются файлы из разных каталогов? Объединяются файлы с различными расширениями? Возможно ли это в ОС Windows 10?

12. *Copy CON + t.dat t1.dat*- выполните команду, объясните полученный результат. (Завершение ввода в файл определяется нажатием клавиш *<Ctrl>+Z* и *<Enter>*). Что будет происходить по команде: *copy CON PRN*?

13. Дополните список команд работы с ФС, файлами и каталогами другими командами этой темы для ОС Windows 10 и сохраните его в виде таблицы, содержащей саму команду, её интерпретацию и ключи (если есть).

14. Приведите примеры различных сочетаний работы с командами перенаправления ввода/вывода для каждого стандартного дескриптора.

15. Запишите команду перенаправления данных, выводимых некоторой командой как на стандартное устройство вывода, так и

на стандартное устройство вывода ошибок одновременно. Проанализируйте результат.

16. Оформить отчёт по практической работе.

4. Требования к оформлению отчёта

Отчёт по практической работе должен включать следующие разделы.

1. Титульный лист (Приложение 1).
2. Цель работы и постановку каждого выполняемого задания.
3. Скриншоты с результатами выполнения исследуемых команд операционной системы.
4. Фрагменты текстовых файлов(где необходимо), в которых фиксируется результат выполнения перенаправления ввода/вывода команды или её результата.
5. Выводы по работе.
6. Ответы на контрольные вопросы.

Контрольные вопросы

1. Какие компоненты содержит файловая система?
2. Перечислите основные функции, возлагаемые на ФС в однопользовательских и однопрограммных ОС.
3. Какая основная задача возлагается на ФС в многопользовательских ОС, в отличие от однопользовательских?
4. Чем отличается файл от каталога в линейке ОС Microsoft Windows?
5. Файловые системы поддерживают несколько функционально различных типов файлов, перечислите их.
6. Перечислите возможные атрибуты файлов в ФС, работающей под управлением Windows 10 (Windows 7).
7. При физической организации файловой системы ключевым является способ размещения файла на диске. Перечислите на сегодняшний день известные структуры.
8. Приведите символы (команды) перенаправления ввода-вывода потоков в ФС Windows-подобной ОС.
9. Что такое конвейер команд в теории ОС?
10. Как удалить несколько файлов одновременно в режиме командной строки Windows 10?

Практическая работа №4 ОСНОВЫ РАЗРАБОТКИ ПАКЕТНЫХ ФАЙЛОВ В ОС MICROSOFT WINDOWS

ЦЕЛЬ РАБОТЫ: исследование средств автоматизации работы пользователей в командном режиме и средств управления ходом выполнения заданий.

1. Основные понятия

Командный (пакетный) файл – это группа последовательных команд настройки компьютера на определённый режим или выполнение определённых операций. Командный файл может иметь фактические и формальные параметры. В линейке Unix (Linux)-подобных ОС, вместо термина «командные файлы», принятого в линейке Windows, чаще используется термин «скрипты». Механизмы и средства реализации в двух линейка рассматриваемых ОС очень похожи. Последовательность записанных команд составляет задание, которое шаг за шагом выполняется командным интерпретатором (языки командных интерпретаторов часто называют языками управления заданиями) [5], [10]. Задания создаются в виде текстовых файлов, но с расширениями bat (сокращение от слова batch – пачка) или cmd, что является индикатором их отношения к исполняемым файлам. Вызов таких файлов в режиме cmd, как правило, состоит из двух частей: имя файла и фактические параметры (если имеются) (Рис. 6).

```
29.03.2021 12:02 <DIR> Videos
      8 файлов             1 499 байт
     18 папок      386 275 479 552 байт свободно

C:\Users\User> 1.bat c 10 *.txt

C:\Users\User>echo 1.bat
1.bat

C:\Users\User>find /c "10" *.txt

----- 1.TXT: 0
----- 11.TXT: 0
----- 111.TXT: 5
----- 333.TXT: 0
----- RR.TXT: 0

C:\Users\User>
```

Рисунок 6 – Вызов командного файла с параметрами в ОС Windows 10

Есть определённые правила, которых надо придерживаться при разработке данных файлов. Так, число фактических параметров ограничено десятью, и эти параметры не обязательны. Данные фактические параметры замещают формальные параметры, присутствующие в тексте исполняемого файла. Ограничение на количество 10 может быть снято командой Shift.

Командный файл может содержать любые внешние и внутренние команды ОС, а также специальные внутренние команды. Результат выполнения командного файла (задания) может состоять в изменении содержимого и состояния файлов и каталогов, настроек ВС, в управлении другими заданиями, в активации различных прикладных и системных программ. Чтобы диагностировать состояние исполняемого пакетного файла, при его завершении формируется код возврата – числовое значение, характеризующее успешность выполнения задания. При помощи специальных средств командного интерпретатора возможен доступ к кодам возврата задания после его завершения.

Большинство программ, работающих под управлением ОС линейки Windows, ориентированы на диалоговый режим – пользователь непосредственно взаимодействует с программой до её окончания. Но пакетный режим в эпоху больших данных и аналитическо-статистической отчётности в прикладных областях экономики и бизнеса является надёжным средством автоматизации рутинных действий и повышения эффективности работы пользователя с ЭВМ, поэтому поддерживается всеми современными ОС.

Команды в задании могут выполняться последовательно и параллельно, для этого существуют специальные разделители, такие как ; () & | && || и другие. Кроме того, имеется возможность задавать различные переменные и формальные параметры при написании задания, как было указано выше. Формальные параметры имеют вид %0, %1 и т.д. до %9. Фактические значения параметров вводятся в строке вызова командного файла; вводимые параметры подставляются на место формальных параметров %1, %2 и т.д. по порядку. На место формального параметра %0, если он встречается в тексте задания, подставляется имя самого командного файла (задания).

При создании заданий пользователь может оперировать тремя видами переменных окружения: встроенные системные и встроенные пользовательские и локальные. Для обращения к значениям переменных окружения, их имена следует заключать в знаки %, например, %ТЕХТ%, для задания значений переменной – пользоваться знаком = и командой *set*:

Set <имя переменной> = <значение переменной>

2. Методические указания

1. Если текущим является каталог (папка), содержащий командный файл, то полный путь к данному файлу можно не указывать.

2. Рекомендуется, во избежание ошибок, располагать отдельные команды в задании на отдельных строках, для локализации команд – пользоваться круглыми скобками.

3. Прекращение выполнения задания можно осуществить вызовом <Ctrl>+<Break> или Ctrl >+C.

4. Перед выполнением очередной строки командного файла (задания), её содержимое выводится на экран. Вывод любой строки командного файла на экран подавляется, если строка начинается с символа @.

5. Изучаемые команды

Echo, Goto, Find, For, If, Mode, Pause, Rem (remark – примечание), Shift, Sort.

3. Последовательность выполнения практической работы

1. Научитесь создавать командный файл в графическом режиме, а затем в режиме командной строки. Воспользуйтесь текстовым редактором в первом случае и вводом с консоли содержимого файла во втором (в командном режиме). Затем попробуйте редактировать полученные файлы.

2. Проверьте работу команды *echo*. Что обозначает: *echo off*, *echo on*, *echo* (без параметров), *echo*+текстовое сообщение, *echo %переменная окружения%*. Исследования поясните. Попробуйте выполнить команду «*echo off*» таким образом, чтобы она не отображалась на консоли. Что обозначает появление в тексте командного файла сообщения: «*rem echo off*»?

3. Используя команду *copy CON* или *edit*, создайте текстовый файл <имя файла>.bat со следующим содержимым:

```
echo off
Cls
echo Hello!
echo:
echo Hello again!
```

Выполните его, затем замените первую команду *echo off* на *echo on*, дайте пояснения, если выявились изменения.

Ту же самую операцию сделайте и со вторым bat – файлом со следующим содержимым:

```
echo off
rem Это первый комментарий
cls
rem Это второй комментарий
echo Hello!
rem Это третий комментарий
echo:           (создание «пустой строки»)
echo Hello again!
rem Это четвёртый и последний комментарий
```

4. Создайте текстовый файл <имя файла>.txt из 4-6 строк текста, в котором есть повторяющиеся слова и конструкции, например:

```
«Help, I need somebody,
Help, not just anybody,
Help, you know I need someone, help.»...из известной песни
«The Beatles»
```

Сформируйте один командный файл: *poisk.bat*, который бы позволял при вызове с различными параметрами осуществлять различные виды поиска в созданном Вами текстовом файле <имя файла>.txt. Например:

```
Find /%1 «%2» %3
```

Посмотрев в справочнике (help) команду *Find*, определите, какие значения формальных параметров Вы должны задать при запуске данного .bat файла?

5. Создайте командный файл с параметром, реализующий следующий ветвящийся алгоритм.

При определённой комбинации символов (например, метка 1), соответствующей формальному параметру, Вы должны активировать командный интерпретатор и выполнить замену приглашения в этой программе на текущую дату.

В случае другой комбинации символов (или метка 2) Вы должны перенаправить выходной поток одной из команд в текстовый файл и при успешном выполнении данной команды открыть этот текстовый файл, активировав соответствующее приложение.

При отсутствии параметра (не заданной метке) или неправильном его задании – обработать ошибку, чтобы на консоль было выдано сообщение для пользователя.

6. Создайте командный файл из которого будет вызван ранее созданный и проверенный другой командный файл. Сделайте это двумя способами. В чём отличие вызовов?

7. Оформить отчёт по практической работе.

4. Требования к оформлению отчёта

Отчёт по практической работе должен включать следующие разделы.

1. Титульный лист (Приложение 1).

2. Цель практической работы и постановку каждого выполняемого задания.

3. Скриншоты с результатами исследования по изучаемым командам операционной системы.

4. Командные файлы и их реализацию в среде командного интерпретатора. Текстовые файлы с результатами или исходными данными для выполняемых заданий.

5. Выводы по работе.

6. Ответы на контрольные вопросы.

Контрольные вопросы

1. В каких случаях целесообразно применение командных файлов?

2. Опишите механизм реализации увеличения числа формальных параметров исполняемых файлов.

3. При помощи специальных средств командного интерпретатора возможен доступ к кодам возврата задания после его завершения. Какие это средства?

4. Перечислите все способы запуска задания на исполнение из другого задания. Чем они отличаются?

5. В чём отличие встроенных системных переменных от встроенных пользовательских? Чем от встроенных переменных отличаются локальные переменные?

6. Приведите список возможных модификаторов параметров командной строки. Для чего они используются?

7. Какие формы логических условий могут быть использованы в условном операторе при составлении заданий?

8. Приведите общую запись оператора цикла для командной строки и командного файла. Есть ли отличие в синтаксисе?

Практическая работа №5 НАЧАЛЬНОЕ ЗНАКОМСТВО С ОС ЛИНЕЙКИ LINUX

ЦЕЛЬ РАБОТЫ: изучить основные принципы работы в одной из ОС линейки Linux, проверить работу команд по управлению системой в режиме терминала (командной строки).

1. Основные положения

ОС Linux относится к мультизадачным многопользовательским операционным системам и продолжает традиции и идеологию линейки ОС UNIX [5], [10]. Код данной операционной системы является открытым и свободно распространяемым. ОС Linux Ubuntu – это одна из версий многочисленного семейства операционных систем, основанных на ядре Linux. На сегодняшний день самая новая версия идентифицируется как Ubuntu 20.04.3 LTS (Focal Fossa), но пользователям доступны и другие дистрибутивы ОС, выполненные на основе Ubuntu, такие как Ubuntu Budgie, Kubuntu, Ubuntu MATE, Xubuntu, Lubuntu, Ubuntu Kylin, Ubuntu Studio [8]. Версии могут отличаться графическим интерфейсом, набором предустановливаемых программ, но они все бесплатные, и недостающие компоненты можно получить на официальных интернет-страницах сообщества Ubuntu Linux. Здесь же можно найти дистрибутивы и более старых надёжных и хорошо зарекомендовавших себя версий этой линейки, таких как Ubuntu 16.04 – достаточных для начального знакомства с Linux.

В данной линейке ОС, как и в Windows, предусмотрено два режима работы: графический и режим командной строки, и многие команды Windows совпадают по имени и частично по функциям с командами Linux. Это такие команды, как `cd`, `copy`, `date`, `echo`, `find`, `for`, `goto`, `if`, `mkdir`, `more`, `rmdir`, `sort` и др. Для получения первичных знаний по командам можно воспользоваться многочисленной литературой по Linux или интернет-источниками [1], [4], [8], [11]. В данной работе предполагается использование режима командной строки, поэтому справку по командам и утилитам получаем с помощью команды `man <название команды>`, для краткой справки `<название команды>--help`.

Работу всех команд операционной системы упрощённо можно представить следующим образом: команда является каким-либо устройством, способным выполнять преобразование входного потока в выходной – это положение кратко было освещено при рассмотрении ОС Windows (Практическая работа 3).

Входной поток в команде можно задать следующими способами:

- указать адрес входного потока справа от команды;
- получить поток по умолчанию, без прямого указания на адрес входного потока;
- воспользоваться перенаправлением входного потока (<);
- передать во входной поток с помощью конвейера (|) (или серфинга) выходной поток предыдущей команды.

В свою очередь, выходной поток может быть задан:

- 1) по умолчанию, без явного указания на выходной поток;
- 2) явным указанием выходного потока;
- 3) перенаправлением выходного потока одной команды на вход другой (|);
- 4) перенаправлением в начало файла (>);
- 5) перенаправлением в конец файла (>>).

2. Методические указания

1. В ОС Linux можно переключиться в текстовый режим комбинацией клавиш <Alt>+<Ctrl>+<F1>, а <Alt>+<Ctrl>+<F7> – обратное переключение в графический режим. Комбинации от Alt-Ctrl-F1 до Alt-Ctrl-F6 загружают шесть отдельных приглашений на вход в систему в текстовом режиме, Alt-Ctrl-F7 – в графическом режиме.

2. Если результат выполнения команды содержит много текста, то как и в предыдущих работах с ОС Windows, можно сохранить этот текст в файле, путём перенаправления вывода (операторы перенаправления >, >>).

3. Приглашение к вводу команды в системе может выглядеть по-разному: # – это приглашение для суперпользователя (root);

\$ или [имя@localhost имя]\$ – для обычных пользователей.

Выход из системы: *logout*(<Ctrl>+<Alt>+<Backspace>) – прекращение сеанса работы с данным пользователем, но система не завершает свою работу; <Ctrl>+<Alt>+ – завершение работы,

а командой *shutdown* можно спланировать выключение. Начать работу с системой можно с команды «*ls*» (*list*) – просмотра содержимого текущего каталога.

Большинство команд соответствуют шаблону: $\$<команда> <ключи> <параметры>$.

Например, для команды *ls*: $\$ls -l a.txt$, если ключей много и они однобуквенные, то можно записать компактно $\$ls -lg a.txt$, а не $\$ls -l -g a.txt$. Ключи начинаются с дефиса (-); есть ключи, требующие наличия параметров, которые должны быть записаны сразу после ключа, поэтому ключи с параметрами не объединяются. Есть и исключения среди команд, которые не соответствуют этим правилам.

4. Предусмотрена, так же как и в Windows, организация конвейера, устанавливающего отношения между выходным потоком одной команды и входным потоком другой, получающей результаты работы первой команды. Например, $ls -l | more$ – выходной поток команды *ls -l* знаком конвейера *|* направляется на вход команды *more*, таким образом информация детально может быть рассмотрена пользователем на экране монитора.

5. Изучаемые команды:

Awk, Cat, Grep, Logout, Lpr, Ls, Man, More, Pr, Shutdown, Sort, Who, Whoami и др.

6. Изучаемые утилиты:

Sysctl – осуществляет доступ к данным о производительности системы, таким как */proc/cmdline* – информация о параметрах ядра, выставленных при загрузке системы; */proc/cpuinfo* – информация о ЦП, на котором выполнено ядро ОС; */proc/meminfo* – информация об ОЗУ ЭВМ; */proc/loadavg* – данные о средней загрузке ЦП, в том числе информацию за разные интервалы (1, 5, 10 мин); */proc/devices* – отображаются различные устройства, подключённые к системе и др.

В системе ведутся различные журналы, такие, например, как системный журнал событий (*/var/log/syslog*), системный журнал событий ядра ОС (*/var/log/dmesg*) и др.

Наиболее популярные утилиты для мониторинга ресурсов системы, её быстродействия объединены в пакет системных утилит *sysstat*. С помощью *sysstat* можно получить информацию о ВС, скорости операций ввода-вывода, объёмах использования файла

подкачки для процессов, информацию о системе прерываний, о сети, ОЗУ, ЦП, и др. В её состав входят утилиты: отчёты с информацией о системе (*sar*); отчёты об использовании процессора и дискового ввода/вывода (*iostat*); отчёты об использовании процессора как полностью, так и по ядрам (*mpstat*); отчёты о задачах (процессах) (*pidstat*), и др.[8], [11].

3. Последовательность выполнения практической работы

1. Найдите справочные данные для команд ОС Linux, название которых совпадает с командами Windows-подобных ОС. Попробуйте их выполнить. Сформулируйте, чем они отличаются.

2. Исследуйте следующие конвейеры команд, опишите их функционал:

- 1) `Cat *.txt | grep problem | lp`
- 2) `Sort textfile | ls`
- 3) `Sort data.txt |pr | lpr`
- 4) `Who | grep tty15`
- 5) `Who | sort`
- 6) `Ls -l | awk '{print $2}'`
- 7) `Ls -l | awk '{print $5}' | sort -n | tail -1`
- 8) `Ls | pr -t -15 -4`
- 9) `Ls | pr -3 | lpr`

3. Выполните мониторинг основных показателей и характеристик о работе установленной гостевой Linux-подобной ОС и проанализируйте полученные данные. Для этого детально изучите содержимое `/proc`.

4. Получите доступ к папкам `/proc/cpuinfo`, `/proc/meminfo` и добавьте из них информацию о системе.

5. Получить доступ к журналам ядра и выбрать из него информацию о ЦП: `dmesg|grep CPU`.

6. Воспользовавшись командой `iostat`, получить информацию о состоянии блочных устройств системы.

7. С какой целью используется команда `sar`? Выполните её с разными ключами, например, `sar -uR 1 5` и объясните полученный результат. Измените параметры и также объясните полученные данные.

8. Оформите отчёт по практической работе.

4. Требования к оформлению отчёта

Отчёт по практической работе должен включать следующие разделы.

1. Титульный лист (Приложение 1).
2. Цель практической работы и постановку каждого выполняемого задания.
3. Скриншоты с результатами исследования по изучаемым командам и утилитами операционной системы.
4. Фрагменты текстовых файлов (где необходимо), в которых фиксируется результат выполнения перенаправления ввода/вывода команды или её результат.
5. Выводы по работе.
6. Ответы на контрольные вопросы.

Контрольные вопросы

1. Какие преимущества даёт пользователю режим терминала.
2. Определите состава пользователей, работающих в данный момент в многопользовательской ОС.
3. Как организовать вывод больших данных(больших файлов) на экран монитора с остановками в командном режиме?
4. Что такое /proc и её назначение?
5. Перечислите все возможности утилиты iostat.
6. Дайте краткую характеристику (по основным критериям деления ОС) последним версиям Ubuntu Linux, появившимся, начиная с 2010 года.
7. В чём ключевые отличия линейки ОС Linux от Windows?

Практическая работа №6 ОСОБЕННОСТИ ОРГАНИЗАЦИИ СТРУКТУРЫ ДАННЫХ В ОС LINUX

ЦЕЛЬ РАБОТЫ: изучить особенности построения файловых систем Linux-подобных ОС и работу с файлами и каталогами в данной линейке ПО.

1. Основные понятия

Файловая подсистема, работающая под управлением ОС, управляет файлами, размещает записи файлов в отведённые для них места, управляет свободным пространством, доступом к файлам и поиском данных для пользователей.

Отличие файловых систем Linux-подобных ОС от файловых систем линейки Windows в том, что в Linux отсутствует такое понятие, как логическое устройство. При указании пути к файлу – имя устройства не указывается, дерево каталогов строится из одного корня. Корневой каталог имеет predetermined имя «/», остальные подкаталоги, ведущие к файлу, подсоединяются к корневому через разделитель, в качестве которого выступает тот же символ «/»:

/каталог1/каталог2/каталог/.../файл.

Физически разные компоненты дерева каталогов в системе могут размещаться на разных дисках, но логически они принадлежат одной древовидной структуре с одним корневым узлом. Для объединения файловых систем различных устройств в одну структуру используется операция монтирования.

Сущность этой операции заключается в том, что каждое физическое устройство можно рассматривать, как свою собственную файловую систему (файловую систему устройства) с корневым каталогом /. Если этот раздел диска объявлен в операционной системе, как корневой раздел (root), его каталог становится корневым каталогом всей файловой системы (файловой системы ЭВМ). ФС остальных устройств должны быть смонтированы в каталогах файловой системы ЭВМ.

Операция монтирования связывает корневой каталог монтируемого раздела (устройства) с выбранным каталогом файловой системы ЭВМ – точкой монтирования. В результате монтирования корневой каталог файловой системы устройства

получает имя каталога, являющегося точкой монтирования, благодаря чему файловая система устройства «привязывается» к файловой системе ЭВМ в точке монтирования.

Работа с файлами и каталогами в режиме командной строки происходит в специальных программах (или программных средах), которые могут идентифицировать файлы и папки (каталоги), выполнять различные операции над ними и исполнять файлы. К таким программам относятся *Sell*, *Bash*, и др.

2. Методические указания

1. Для получения справки о свободном месте на диске, необходимо воспользоваться командой *df*. Эта команда позволяет определить:

- общий объём раздела;
- размер занятого данными дискового пространства;
- объём свободного места на диске;
- сведения о точках монтирования и т.д.

2. По отношению к файлам приняты следующие популярные команды и утилиты: *open* (открыть файл на чтение или запись), *close*, *read*, *write*, *stat* (запросить атрибуты файла), *chown* (изменить запись с информацией о владельце файла) и *chmod* (изменить права доступа к файлу).

3. Создание пустого файла производится с помощью знака *>*, перенаправляющего стандартный вывод с экрана в файл.

4. В ОС *Linux Ubuntu* имеется механизм создания жёстких и символических ссылок на файлы и папки. Синтаксис исполнения ссылок отличается лишь ключом.

5. Изучаемые команды:

cat, *chgrp*, *chown*, *cp*, *df*, *du*, *ln*, *mkdir*, *more*, *move*, *rm*, *touch*, *wc* и др.

3. Последовательность выполнения практической работы

1. Определите структуру каталогов на Вашем ПК, перенаправьте иерархическое дерево в отдельный файл и поясните цветовую гамму изображения папок и файлов полученной структуры.

2. Выведите в отдельный файл календарь на три месяца зимы.

3. Приведите различные способы создания файлов (пустых и не пустых) – объясните механизм команд.

4. Определите права доступа к одному из файлов и директории. Измените их. Продемонстрируйте на примере. Сколько групп пользователей файлов и папок имеется в системе?

5. Выберите или создайте директорию и разрешите всем группам пользователей в данной директории – всё. Разрешите всем группам пользователей в данной директории только читать файлы. Разрешите user и group только читать и писать, а others – ничего в данной директории.

6. Выберите любой файл в текущей директории. Создайте на него «мягкую» (символическую) и жёсткую ссылку. В чём их отличие? Продемонстрируйте аналогичную операцию для папки (директории).

7. Сделайте выборку о всех ссылках данной директории и запишите информацию в отдельный файл перенаправлением вывода.

8. Приведите расшифровку каждого столбца в команде ls -l и объясните цветовую палитру (если таковая имеется), предназначенную для подсветки различных файлов.

9. Продолжите работу с виртуальной файловой системой /proc, начатую в Практической работе 5. Получите сведения о системе на файловых ресурсах: proc/sys/fs, proc/sys/fs/file-max, proc/sys/net. Поясните полученную информацию.

10. Оформите отчёт по практической работе.

4. Требования к оформлению отчёта

Отчёт по практической работе должен включать следующие разделы.

1. Титульный лист (Приложение 1).

2. Цель и постановку каждого выполняемого задания.

3. Скриншоты с результатами исследования по изучаемым командам операционной системы.

4. Фрагменты текстовых файлов (где необходимо), в которых фиксируется результат выполнения перенаправления ввода/вывода команды или её результат.

5. Выводы по работе.

6. Ответы на контрольные вопросы.

Контрольные вопросы

1. Объясните сущность понятия «монтирование» в применении к ОС Linux.
2. Приведите примеры создания, копирования и удаления файлов с рабочего стола.
3. Приведите пример, как команда *move* может быть использована для перемещения и переименования файла одновременно.
4. Можно ли удалить не пустой каталог? Если да – то приведите пример и дайте пояснения.
5. Чтобы результаты, выдаваемые командами *du*, *df*, были хорошо читаемы и понятны, какие ключи следует использовать с данными командами.
6. Приведите примеры графических оболочек для Ubuntu 16.04. Для других версий Ubuntu.

Практическая работа №7 УПРАВЛЕНИЕ ПРОЦЕССАМИ В ОС LINUX

ЦЕЛЬ РАБОТЫ: закрепить представление об основных возможностях подсистемы управления процессами ядра ОС Linux Ubuntu.

1. Основные понятия

Процессом называется программа в стадии выполнения. Управление процессами – задача ядра операционной системы. Для поддержания информации о процессах, ядро использует специальную структуру – таблицу процессов. Каждому процессу назначается уникальное число-идентификатор процесса, которое используется ядром для однозначного определения того, какому процессу соответствует запись в таблице процессов.

Подсистема управления процессами ядра ОС отвечает за синхронизацию процессов, их взаимодействие, распределение памяти и планирование выполнения процессов. По характеру выполнения процессы могут быть фоновыми и привилегированными. Любой запускаемый процесс по умолчанию будет выполняться как привилегированный (*foreground*). Это значит, что такой процесс постоянно связан с терминалом ЭВМ и делает невозможным выполнение ещё каких-либо действий с системой, пока он не завершится.

Фоновый процесс (*background*) после запуска освобождает терминал и позволяет перейти к другой задаче, не дожидаясь его завершения. Фоновая обработка наиболее пригодна для процессов, которые долго выполняются. Программы, выполняющиеся в виде фоновых процессов, называются демонами (*daemon*). В любой момент времени в системе существуют десятки процессов, которые были запущены при старте операционной системы, вызваны ядром для обслуживания каких-либо событий, добавлены пользователем при запуске какой-либо задачи.

Обычно большинство процессов находится в состоянии ожидания – сна (*sleep*), не мешая остальным и дожидаясь сигнала для активизации. Кроме того, в системе можно найти процессы, закончившие работу, но ещё не получившие разрешения на выгрузку из ОЗУ, эти процессы называются *зомби*.

В Linux-подобных ОС (например, ОС Ubuntu), между процессами поддерживаются отношения «родитель»-«потомок». Каждый процесс может породить произвольное число процессов-потомков, но каждый процесс имеет только одного родителя, и нет ни одного процесса, который не имел бы родителя. Самый первый процесс, запускаемый ядром в ходе «раскрутки» системы, называется *init* и служит для управления процессом раскрутки системы, в том числе запуска *процессов-демонов*.

Если характеризовать демоны, они также известны как системные сервисы или процессы, производящие не интерактивную обработку данных пользователя (т.е. процесс-демон не подключен к терминальному устройству). Примером процессов-демонов могут служить серверы протоколов HTTP и FTP *httpd* и *ftpd* или сервер системного журнала *syslogd*. Часто имена исполняемых файлов программ-демонов заканчиваются буквой *d*. Процессу *init* назначается идентификатор 1, первому порождённому *init* процессу – 2 и так далее. В ОС Ubuntu для идентификатора процесса общеупотребительным является сокращение PID.

После того, как процесс порожден, т.е. ему отведена запись в *системной таблице процессов*, он находится в состоянии готовности и может быть выбран планировщиком процессов операционной системы для выполнения. При выполнении операций ввода-вывода и в ряде других случаев процесс может перейти в состояние ожидания некоторого события; тогда до наступления этого события или поступления в процесс некоторого прерывания он не будет выбираться для выполнения. Кроме процессов, находящихся в состояниях готовности и ожидания, в ОС Ubuntu могут присутствовать остановленные процессы и процессы-зомби, о которых упоминалось ранее.

2. Методические указания

1. Для того чтобы узнать, какие процессы в данный момент существуют в системе, необходимо воспользоваться командой *ps*.

2. Часто используются специальные директивы объединения команд: конвейеры, перенаправления ввода-вывода, логические операторы.

3. Команда *kill* может передавать сигнал процессу. Сигнал может передаваться в виде его номера или символического

обозначения. По умолчанию (без указания сигнала) передаёт сигнал завершения процесса. Идентификация процесса для команды *kill* производится по PID. Перечень системных сигналов, доступных в Linux, с указанием их номеров и символьных обозначений можно получить с помощью команды *kill -l*.

4. Команда *killall* работает аналогично команде *kill*, но для идентификации процесса использует его символьное имя, а не PID.

5. К командам управления процессами можно отнести следующие: *pidof* – определяет PID процесса по его имени; *pgrep* – определяет PID процессов с заданными характеристиками (например, запущенные конкретным пользователем); *pkill* – позволяет отправить сигнал группе процессов с заданными характеристиками; *nice* – запускает процесс с заданным значением приоритета, причём, возможность инициации уменьшения или увеличения приоритета доступна только пользователю *root*; *renice* – изменяет значения приоритета для запущенного процесса, возможность инициации уменьшения или увеличения приоритета – только пользователю с правами *root*.

6. Следует указать и на команды: *at* – она осуществляет однократный отсроченный запуск команды; *cron* – демон, который занимается планированием и выполнением команд, запускаемых по определённым датам и в определённое время.

7. Команды, выполняемые периодически, указываются в файле */etc/crontab*, (не через команду *cron*, а путём внесения строк в файл *crontab* или с использованием одноименной команды *crontab*). Команды, которые должны быть запущены лишь однажды, добавляются при помощи *at*.

8. Каждая команда в файле *crontab* занимает одну строку и состоит из шести полей: минута час день_месяца месяц день_недели команда.

9. Для вывода последних *n* строк из файла служит команда *tail*, она позволяет также обнаруживать и выводить новые строки, появляющиеся в конце файла; *sleep* – задает паузу в выполнении скрипта.

10. Организация взаимодействия двух процессов осуществляется двумя способами: передачей данных через файл и передачей управления посредством сигнала.

11. Изучаемые команды:

at, cron, crontab, kill, killall, nice, pkill, ps, renice, sleep, tail, top.

3. Последовательность выполнения практической работы

1. Исследуйте команду *ps*: приведите её ключи и проверьте их действие. Приведите расшифровку столбцов таблицы, выдаваемой командой *ps -p* <список идентификаторов процессов>.

2. Выполните команду вывода приветствия «Hello World!» в 11:00 часов в каждые 6, 7, 8, 9 дни января, февраля и марта, а приветствие «Good day!» – каждый чётный час каждого понедельника.

3. Приведите команду *ps* таким образом, чтобы можно было посмотреть подробную информацию обо всех процессах, запущенных от имени текущего пользователя; список процессов должен быть упорядочен по степени использования процессорного времени.

4. Приведите пример использования данных команд:

nice [-n <число>] <команда>

renice -n <число> <список идентификаторов процессов>

kill -s

5. Организуйте взаимодействие двух процессов посредством передачи данных через файл. Для этого в разных консолях запустите каждый из скриптов. Тогда, переключаясь между консолями, можно вести управление и наблюдение за их работой. Тексты скриптов приведены в данном пункте ниже. Короткий скрипт считывает информацию с консоли. Вторым скрипт, более длинный, выводит её на экран, причём таким образом, чтобы считывание первым скриптом строки «QUIT» приводило к завершению работы обработчика.

```
#!/bin/bash while true; do read LINE echo $LINE >> data.txt done
```

```
#!/bin/bash (tail -n 0 -f data.txt) | while true; do read LINE; case $LINE in QUIT) echo "exit" killall tail exit ;; *) echo $LINE ;; esac done
```

6. Составьте конвейер команд, где происходит активация какого-либо процесса, затем по таблице идентификаторов процессов определите его PID и завершите этот процесс, например, таким образом: *kill -s SIGSTOP 123*. Проанализируйте действия и результаты по командам.

7. Оформите отчёт по практической работе.

4. Требования к оформлению отчёта

Отчёт по практической работе должен включать следующие разделы.

1. Титульный лист (Приложение 1).
2. Цель и постановку каждого выполняемого задания.
3. Скриншоты с результатами исследования по изучаемым командам операционной системы.
4. Фрагменты текстовых файлов (где необходимо), в которых фиксируется результат выполнения перенаправления ввода/вывода команды или её результат.
5. Выводы по работе.
6. Ответы на контрольные вопросы.

Контрольные вопросы

1. Перечислите все основные задачи по управлению процессами в ОС Linux Ubuntu.
2. Каков состав полей, задействованных в системной таблице процессов? Перечислите основные.
3. Опишите синтаксис строки в файле *crontab*.
4. Как можно получить перечень всех доступных системных сигналов процессов?
5. Какие функции возложены на команду *top*? Какую клавишу следует нажимать при выходе из команды?
6. В чём отличие управляющих символов, используемых при взаимодействии процессов: `||` и `|;` `&&` и `&`?

Практическая работа №8 ОСНОВЫ ПРОГРАММИРОВАНИЯ В КОМАНДНОЙ СРЕДЕ В ОС LINUX

ЦЕЛЬ РАБОТЫ: исследование отличий разработки командных файлов в среде ОС Linux от командных файлов в среде Windows. Создание элементарных скриптов в ОС Linux Ubuntu.

1. Основные понятия

Исполняемые файлы могут выполняться в средах различными оболочками (Практическая работа 7), одна из них – shell.

Командный язык shell (в переводе – раковина, скорлупа) – фактически есть язык программирования 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. Такой режим значительно ускоряет и упрощает работу с программами пользователя (сценариями и процедурами).

Для того чтобы, работая в терминале, переключиться в среду Shell, необходимо ввести команду sh. Признаком того, что оболочка (shell) готова к приёму команд, служит выдаваемый ею на экран символ приглашения («\$»). 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 «=».

Например, 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, т.к. в строке есть пробел.

Необходимо помнить, что и переменная, и её значение должны быть записаны без пробелов относительно символа "=".

Для того чтобы текстовый файл можно было использовать как командный, существует несколько возможностей. Пусть с помощью редактора создан файл с именем My, содержащий одну строку следующего вида: *Date; pwd; ls*.

Можно вызвать оболочку shell как команду, обозначаемую sh, и передать ей файл My, или как аргумент, или как перенаправленный вход: \$sh My, или \$sh < My,.

В результате выполнения любой из этих команд будет выдана дата, затем имя текущего каталога, а потом содержимое каталога.

Файл можно выполнять и в текущем экземпляре shell. Для этого существует специфическая команда «.» (точка).

.My

Другой вариант работы с командным файлом – это превращение его в выполняемый файл, т.е. просто сделать его командой (запускающей этот файл по его названию), что достигается изменением кода защиты. Для этого надо разрешить выполнение этого файла. Например: Chmod 711 My сделает у этого файла код защиты «RWX X». Тогда простой вызов My, приведёт к выполнению тех же трех команд, которые были записаны в файл.

Результат будет тот же, если файл с содержимым Date; pwd; ls будет представлен в виде:

Date

pwd

ls

так как переход на другую строку также является разделителем в последовательности команд.

Таким образом, выполняемыми файлами могут быть не только файлы, полученные в результате компиляции и сборки, но и файлы, написанные на языке shell. Их выполнение происходит в режиме интерпретации с помощью shell-интерпретатора. Начинать командный файл надо всегда с пустой строки или пустого оператора «:».

2. Методические указания

1. Создание исполняемых файлов возможно в редакторе vim (Vi Improved) или ином другом редакторе.

2. Изучаемые команды:

banner echo, else, if, for until, read, set, test, then, wc, while и др.

3. Последовательность выполнения практической работы

1. Составить командный файл, определяющий: кому принадлежит текущий каталог; дату его создания и состав

содержащихся в нём файлов; общее количество файлов в текущем каталоге; последние 10 раз, когда заходил в систему пользователь с текущим «логином».

2. Написать скрипт: записать ряд Фибоначчи в файл; выдать n -ое число Фибоначчи на консоль, определить сумму n чисел этого ряда. Значение параметра n должно передаваться при вызове скрипта в среде окружения.

3. Создать небольшой скрипт, в котором было бы предусмотрено ожидание ввода трёх переменных с консоли, а затем выдача их на экран. Провести эксперимент, задавая недостаточное количество значений переменных или избыточное. Описать, как реагирует командный интерпретатор на такие действия.

4. Задание по вариантам: разработайте процедуру и выполните её в среде командной оболочки в ОС Linux. Отладьте полученный сценарий, при необходимости отредактируйте и продемонстрируйте преподавателю.

Варианты процедур

1) Вводит последовательность из N слов и подсчитывает в каждом введённом 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 из N слов, анализирует длину каждого слова, упорядочивает слова по алфавиту и выводит список на экран. Значение N задаётся 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, 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, разделённых 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 каталог и выводит на экран число файлов различного типа (обычные, директория, скрытые). Тип задается параметром.

13) Ищет в заданном параметром каталоге все текстовые файлы – выводит их число и в найденных файлах далее ищет слово (строку), заданное вторым параметром и так же выводит число файлов, в которых найдена заданная строка.

14) Вводит несколько слов, число которых задано в виде параметра, составляет из них предложение и подсчитывает количество символов в предложении, включая пробелы-разделители. Результат выводит на консоль.

15) Выполняет организацию запуска фонового процесса и по истечении заданного параметром таймаута – останавливает его.

5. Оформите отчет по работе.

4. Требования к оформлению отчёта

Отчёт по практической работе должен включать следующие разделы.

1. Титульный лист (Приложение 1).

2. Цель практической работы и постановку каждого выполняемого задания.

3. Скриншоты с результатами исследования по изучаемым командам операционной системы.

4. Фрагменты текстовых и исполняемых файлов (где необходимо), в которых фиксируется результат выполнения перенаправления ввода/вывода команды или её результат.

5. Выводы по работе.

6. Ответы на контрольные вопросы.

Контрольные вопросы

1. В каких случаях целесообразно применение командных файлов?

2. Опишите разницу в написании, внешнем изображении и выполнении командных файлов в Linux-подобных ОС и в Windows-подобных ОС.

3. В чём достоинства и недостатки программирования в средах окружения?

4. Сколько параметров может иметь командный файл?

5. Найдите в учебной литературе или разработайте код скрипта, который позволяет расширять количество формальных параметров для исполняемых файлов в Linux-подобных ОС.

ЛИТЕРАТУРА

1. Арпачи-Дюрсо Р., Арпачи-Дюрсо А. Операционные системы. Три простых элемента / Арпачи-Дюрсо Р. – Изд-во ДМК Пресс, 2021. – 730с.-ISBN-978-5-9706-0932-3.

2. Гордеев А. В. Операционные системы: учеб. / А. В. Гордеев. – 2-е изд. – СПб., Питер, 2007. – 415 с. (Учебник для вузов).-ISBN 5-94723-632-3.

3. Исаева Г.Н. Виртуальные машины: современная реальность программных систем// Актуальные тренды и перспективы развития науки, техники и технологий/ сборник науч. труд. по материалам междуна. научно-прак. конф. 30 января 2019 г., под общ. ред. Е. П. Ткачевой. – Белгород: ООО Агентство перспективных научных исследований (АПНИ), 2019. – С.108-112.

4. Колисниченко Д. Н. Linux-сервер своими руками / Электронный ресурс [Текст]/ Код доступа: <https://it.wikireading.ru/12360/> [Дата обращения 17.01.2022].

5. Назаров С.В. Современные операционные системы: учебное пособие /С.В. Назаров, А.И. Широков. – М.: Национальный Открытый Университет «ИНТУИТ», 2012. – 367 с.: ил., табл. – (Основы информационных технологий). ISBN 978-5-9963-0416-5.

6. Олифер В., Олифер Н. Сетевые операционные системы: учебник для вузов. – 2-е изд. – СПб: Питер, 2011. – 669 с.:ил. – ISBN 978-5-91180-528-9.

7. Официальная документация от Microsoft/ Электронный ресурс// Код доступа <https://docs.microsoft.com/ru-ru/windows-insider/> [Дата обращения 17.01.2022].

8. Официальная документация Linux Ubuntu/ Электронный ресурс/ Код доступа <https://ubuntu.ru/> [Дата обращения 17.01.2022].

9. Симонович С.В. Информатика: Базовый курс: учеб./ С.В. Симонович. – Стандарт третьего поколения. – Изд. 3. – СПб., Питер, 2022. – 640 с. – (Учебник для вузов). – ISBN 978-5-4461-0842-8.

10. Таненбаум Э. Современные операционные системы : пер. с англ / Э. Таненбаум. – 2-е изд. – СПб.: Питер, 2005. – 1037 с. – (Классика Computer Science). – ISBN 5-318-00299-4.

11. Этапы загрузки ОС Linux Ubuntu// Электронный ресурс/ Код доступа <https://losst.ru/ustanovka-ubuntu-16-04/> [Дата обращения 17.01.2022].

ПРИЛОЖЕНИЕ



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
имени дважды Героя Советского Союза, летчика-космонавта А.А. Леонова

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ
СИСТЕМ И ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И УПРАВЛЯЮЩИХ СИСТЕМ**

Отчёт по практической работе №_
по дисциплине «Операционные системы, среды и оболочки»

Выполнил: студент гр. _____ .

Преподаватель: доцент КИТУС Исаева Г.Н

Королев, 202__г.