

Департамент образования администрации г. Липецка  
Муниципальное образовательное учреждение дополнительного образования детей  
Центр детского (юношеского) технического творчества «Городской» г. Липецка

**«УТВЕРЖДЕНО»**  
на педагогическом совете  
**26.03.2009**

# Лаборатория ЮНОГО ЛИНУКСОИДА

<http://www.younglinux.info>

## Программа дополнительного образования детей

Возраст обучающихся — 11-16 лет  
Срок обучения — 1- 4 года

Автор: Шапошникова Светлана Вячеславовна,  
педагог дополнительного образования  
по компьютерным технологиям



Липецк, 2009

## Оглавление

1	Пояснительная записка.....	3
1.1	Цель и задачи программы.....	4
1.2	Ожидаемые результаты .....	4
1.3	Формы подведения итогов реализации программы.....	5
2	Учебно-тематический план.....	6
2.1	Первый год обучения.....	6
2.2	Второй год обучения.....	6
2.3	Третий год обучения.....	7
2.4	Четвертый год обучения.....	8
3	Содержание программы.....	9
3.1	Первый год обучения.....	9
3.2	Второй год обучения.....	10
3.3	Третий год обучения.....	11
3.4	Четвертый год обучения.....	13
4	Программное обеспечение.....	15
5	Информационное обеспечение.....	16
6	Методическое обеспечение.....	17
6.1	Информационно-методическая поддержка.....	17
6.2	Тест по теме «Обработка текстовой информации».....	17
6.3	Разработка занятия к теме «Принципы работы в графических средах».....	19
	Новый материал.....	19
	Что такое файл и каталог?.....	19
	Получаем дерево?!.....	20
	Каждому файлу – адрес!.....	21
	Файловый менеджер – ваш помощник.....	21
	Навигация и создание файлов.....	22
	Практическая работа. Создание файлов и каталогов.....	23
	Вопросы.....	23
	Выводы.....	24

## 1 Пояснительная записка

Образовательная программа «Лаборатория юного линуксоида» разработана для детских (юношеских) объединений, реализующих программы дополнительного образования, связанные с информационными технологиями. Программа ориентирована на возраст обучающихся 11 — 16 лет (5 — 10 классы включительно) и рассчитана на период обучения от 1 года до 4 лет.

Компьютерная техника и информационные технологии стали неотъемлемой частью жизни большинства людей. В настоящее время уже мало актуально считать целью обучения знакомство с компьютерными технологиями, т.к. сегодняшнее поколение детей уже в младших классах нередко владеет компьютерной техникой на уровне пользователя. Однако, часто эти знания отрывочны, не систематизированы, не имеют под собой теоретических основ. Известно, что большой объем неупорядоченной информации может приводить к информационному стрессу. Поэтому для педагога, преподающего дисциплины связанные с информационными технологиями, все более становится актуальной проблема обучения основополагающим принципам и направлениям ИТ, систематизация знаний учащихся. С другой стороны, не будет ошибкой утверждение, что объем знаний, связанных с информационными технологиями, превосходит любую другую науку. При этом данные знания не являются сильно зависящими друг от друга (так например, программисту на языке С не обязательно знать принципы web-дизайна). Разнообразие «компьютерных» направлений создает проблему выбора предмета изучения или последовательности изучения различных направлений. Очевидно, выбор будет зависеть от цели программы.

В данной программе дополнительного образования решается задача постепенного углубления и систематизации знаний учащихся. Другими словами, воспитанник на каждом цикле обучения (длительность цикла 1 год) получает законченный объем знаний определенного уровня по различным направлениям информационных технологий. В следующем цикле (в следующем году) происходит углубление и расширение знаний. Каждый год обучения состоит из пяти блоков (модулей), которые можно условно назвать: «Системно-административный», «Текстово-публикационный», «Графика и моделирование», «Программирование», «Дополнительный». Таким образом, программу можно было бы назвать «Модульное обучение информационным технологиям».

Модульная структура программы позволяет преподавать материал, соответствующий возрасту обучающихся, выявлять потенциал учащихся в том или ином направлении на раннем этапе реализации программы, заинтересовать большее количество детей и подростков, разнообразить учебный процесс, реализовывать в рамках программы различные элективные курсы. Однако, программа не подразумевает узкой специализации (например, не предполагается подготовка исключительно будущих программистов или специалистов по компьютерной графике). Скорее программа направлена на разностороннее развитие личности, формирование единой картины «информационного» мира.

Другой особенностью программы является ее ориентация на использование свободного программного обеспечения (СПО) в качестве средства обучения и предмета изучения. Использование СПО позволяет гарантировать равные возможности участникам образовательного процесса.

Кроме того, в программе сделан акцент на проектный подход. Так, уже в первый год обучения предусмотрена проектная деятельность по таким темам как «Создание компьютерных презентаций», «Введение в композицию и дизайн», «Основы компьютерной 2D-анимации». В последующих годах обучения доля проектной деятельности увеличивается; кроме того, предусматривается реализация проектов как по компьютерной графике, так и программированию.

## 1.1 Цель и задачи программы

Цель программы — формирование системы знаний обучающихся об основных направлениях информационных технологий, информации и способах ее обработки, программировании, а также формирование навыков работы в современных программных средах.

### Обучающие задачи программы:

- сформировать представления об операционной системе, прикладной программе, глобальной сети, о возможностях информационных технологий в отношении обработки графической и текстовой информации;
- обучить навыкам работы с современным программным обеспечением;
- познакомить с основами программирования, объектно-ориентированным программированием, инструментами для создания графического интерфейса пользователя.

### Развивающие задачи программы:

- развитие образного и логического мышления;
- развитие творческого подхода к решению различных задач.

### Воспитательные задачи программы:

- формирование умения планировать деятельность, ставить цели и выделять главное для решения той или иной задачи в условиях избыточности информации;
- воспитание самостоятельности и стрессоустойчивости;
- формирование представления о мире как системе разнообразных взаимодействующих объектов;
- воспитание культуры взаимодействия с другими людьми в условиях информационного общества.

## 1.2 Ожидаемые результаты

### В конце первого года обучения воспитанник должен

знать название и назначение основных частей компьютера, иметь представление об операционной системе как управляющей ресурсами компьютера программе и уметь работать в ней на уровне пользователя; владеть навыками работы в текстовом процессоре и графических редакторах, уметь подготовить компьютерную презентацию; иметь представление об алгоритме, знать основные алгоритмические конструкции, уметь «читать» блок-схемы; уметь составлять простейшие программы в структурном стиле на языке программирования, знать основные инструкции языка; иметь представление о компьютерной анимации.

### В конце второго года обучения воспитанник должен

уметь работать в текстовом режиме, знать файловую структуру операционной системы на базе ядра Linux; знать язык гипертекстовой разметки и принципы создания Web-узла; владеть навыками обработки растровых изображений, создавать текстовые эффекты; иметь представление о различных парадигмах программирования, знать основные принципы объектно-ориентированного программирования, уметь создавать простейшие сценарии в объектно-ориентированном стиле на языке программирования; знать форматы звуковых и видеофайлов, владеть навыками обработки мультимедийной информации.

**В конце третьего года обучения воспитанник должен**

иметь представление о прошлом и настоящем состоянии информационных технологий; иметь навыки работы в консольных редакторах; уметь пользоваться компьютерными технологиями при создании публикаций; иметь представление о моделировании, трехмерном моделировании, уметь создавать простейшие трехмерные модели в среде 3D-моделирования; знать об инструментах создания графического интерфейса пользователя, уметь его конструировать и программировать; владеть основными сведениями о базах данных и способах работы с ними.

**В конце четвертого года обучения воспитанник должен**

знать, что такое проприетарная и свободно-распространяемая программа, открытый и закрытый код; иметь представление о многообразии операционных систем; уметь создавать сложные публикации; владеть навыками создания трехмерной графики и анимации; уметь реализовывать простейшие проекты, связанные с программированием; владеть навыками работы с электронными таблицами.

### **1.3 Мониторинг. Формы подведения итогов реализации программы**

<i>Год обучения</i>	<i>Начальная диагностика</i>	<i>Промежуточная аттестация</i>	<i>Итоговая аттестация</i>
1	Собеседование	Тестирование. Творческие работы по темам «Создание компьютерных презентаций», «Введение в композицию и дизайн», «Основы компьютерной 2D-анимации».	-
2	Тестирование	Тестирование. Творческие работы по темам «Использование профессиональных инструментов для создания растровой графики», «Изменение изображений», «Создание текстовых эффектов» и «Практика объектно-ориентированного программирования».	Разработка сайта-визитки и/или простого приложения, исходный код которого написан в стиле ООП.
3	Тестирование	Тестирование. Творческие работы по темам «Практика создания публикаций», «Практика 3D моделирования», «Разработка приложений с графическим интерфейсом», «Разработка баз данных»	Разработка простого приложения с графическим интерфейсом пользователя.
4	Тестирование	Тестирование. Творческие работы по темам «Практика использования LaTeX», «Практика программирования».	Участие в конкурсах, выставках и олимпиадах.

## 2 Учебно-тематический план

### 2.1 Первый год обучения

Модуль	№ темы	Название темы	Всего часов	Теория	Практика
I*	1	Введение в информационные технологии	10	6	4
	2	Принципы работы в графических средах	10	4	6
II	3	Обработка текстовой информации	20	10	10
	4	Создание компьютерных презентаций	12	4	8
III	5	Компьютерная графика	4	4	0
	6	Создание растровой графики	12	4	8
	7	Создание векторной графики	12	4	8
	8	Введение в композицию и дизайн	8	4	4
IV	9	Алгоритмика	12	8	4
	10	История и разнообразие языков программирования	4	4	0
	11	Введение в программирование	16	8	8
V	12	Основы компьютерной 2D-анимации	24	10	14
Итого:			144	70	74

\* Название модулей:

I – «Системно-административный»

II – «Текстово-публикационный»

III – «Графика и моделирование»

IV – «Программирование»

V – «Дополнительный»

### 2.2 Второй год обучения

Модуль	№ темы	Название темы	Всего часов	Теория	Практика
I	1	Основы работы в режиме командной строки	12	6	6
	2	Файловая система Linux	12	6	6
	3	Введение в локальные и глобальные сети	6	4	2
II	4	Знакомство с языком гипертекстовой разметки	16	8	8
	5	Введение в технологии Web	38	18	20
III	6	Использование профессиональных инструментов для создания растровой графики	24	8	16

	7	Изменение изображений	18	6	12
	8	Создание текстовых эффектов	12	4	8
IV	9	Парадигмы программирования	12	6	6
	10	Основные принципы объектно-ориентированного программирования	12	8	4
	11	Практика объектно-ориентированного программирования	30	12	18
V	12	Обработка звуковой информации	12	6	6
	13	Обработка видеoinформации	12	6	6
Итого:			216	98	118

### 2.3 Третий год обучения

Модуль	№ темы	Название темы	Всего часов	Тео-рия	Прак-тика
I	1	История развития информационных технологий	6	4	2
	2	Принципы работы в консольных редакторах	12	4	8
	3	Знакомство с расширенными функциями консольных редакторов	12	6	6
II	4	Основы верстки публикаций	20	10	10
	5	Практика создания публикаций	24	8	16
III	6	Виды компьютерного моделирования	6	4	2
	7	Особенности трехмерной графики	4	4	0
	8	Практика 3D-моделирования	26	8	18
IV	9	Инструменты для создания графического интерфейса пользователя	6	4	2
	10	Знакомство с программированием основных элементов графического интерфейса	24	12	12
	11	Разработка приложений с графическим интерфейсом	30	12	18
V	12	Введение в базы данных	6	6	0
	13	Инструменты для работы с данными	26	10	16
	14	Разработка баз данных	14	4	10
Итого:			216	98	118

**2.4 Четвертый год обучения**

<i>Модуль</i>	<i>№ темы</i>	<i>Название темы</i>	<i>Всего часов</i>	<i>Теория</i>	<i>Практика</i>
I	1	Правовая охрана программ. Лицензирование ПО	12	10	2
	2	Разнообразие операционных систем	12	8	4
II	3	Знакомство с языком TeX	4	2	2
	4	Основы работы с издательской системой LaTeX	24	8	16
	5	Практика использования LaTeX	8	2	6
III	6	Использование профессиональных инструментов для создания трехмерной графики	24	12	12
	7	Основы 3D-анимации	24	10	14
	8	Практика использования трехмерной графики в моделировании	12	4	8
IV	9	Введение в практическое программирование	12	6	6
	10	Практика программирования	48	18	30
V	11	Основы работы с электронными таблицами	36	18	18
Итого:			216	98	118



## 3 Содержание программы

### 3.1 Первый год обучения

#### 1. Введение в информационные технологии

Что такое информация. Компьютер как средство обработки информации. История информационных технологий. Устройство и типы компьютеров. Понятия: операционная система, программа.

---

#### 2. Принципы работы в графических средах

Текстовый и графический режимы работы. История возникновения графического интерфейса. Разнообразие графических сред. Основные элементы графической среды (Desktop, Главное меню, окно). Понятия: файл, директория (папка, каталог), ярлык, формат файла. Операции копирования, перемещения, переименования и удаления. Представление о файловой системе как древовидной структуре. Файловые менеджеры.

---

#### 3. Обработка текстовой информации

Особенности хранения текста в памяти компьютера. Разнообразие текстовых редакторов и процессоров. Ввод и редактирование текста. Форматированный и «плоский» текст. Добавление в текстовый файл таблиц и графики. Списки и их типы. Логическая структура документа, стили. Дополнительные возможности современных текстовых процессоров.

---

#### 4. Создание компьютерных презентаций

Понятия презентации, компьютерной презентации, слайда. Программы для создания презентаций. Структура и оформление слайда. Логическая структура презентации. Спецэффекты и интерактивность. Публикация презентаций.

Проектная деятельность.

---

#### 5. Компьютерная графика

Компьютерная графика, векторная и растровая графика. Цветовые модели, RGB. Распространенные форматы графических файлов и области их применения (GIF, JPEG, PNG). Методы сжатия изображений.

---

#### 6. Создание растровой графики

Программы для создания растровой графики. Инструменты (выделение, примитивы, заливка и др.) и палитра. Основные принципы работы в графическом редакторе (перемещение частей изображения, изменение размеров, дублирование, поворот и др.).

---

#### 7. Создание векторной графики

Программы для создания векторной графики. Понятие холста. Фигуры и их трансформация. Инструмент «Кривая Безье». Булева алгебра в графике. Работа с цветом. Контур. Графическое оформление текста. Эффекты.

---

#### 8. Введение в композицию и дизайн

Виды дизайна. Особенности формирования изображения. Композиция. Передний и задний планы, равновесие, замкнутое и открытое пространство, динамика и статичность.

Проектная деятельность.

---

#### 9. Алгоритмика

Понятия алгоритма, исполнителя. Признаки алгоритма. Исходные данные и

---

результат. Способы описания алгоритмов. Линейный и разветвляющийся алгоритмы. Циклы.

---

### **10.История и разнообразие языков программирования**

Понятия программирования, компьютерной программы, языка программирования. Исходный и исполняемый код. Транслятор, компилируемые и интерпретируемые языки. История развития языков программирования.

---

### **11.Введение в программирование**

Способы выполнения программы, интерактивный режим и файлы. Среды разработки. Типы данных: числа (целые, дробные и др.) и строки. Операторы и выражения. Представление о переменной. Ключевые слова. Инструкции: условие, циклы. Комментарии. Структуры данных: строки, кортежи, списки и словари. Значение функций, их определение и вызов, параметры и аргументы. Использование модулей, импорт, создание.

---

### **12.Основы компьютерной 2D-анимации**

Понятие анимации. Программы для создания 2D-анимации. Кадры, слои и сцены. Ключевой и промежуточный кадры. Изменения положения и формы объекта во времени. Циклическая анимация и многократное использование частей сцены. Публикация анимации.

Проектная деятельность.

---

## **3.2 Второй год обучения**

### **1.Основы работы в режиме командной строки**

История и назначение командной строки, терминалы. Правила ввода, команды и утилиты, работа со страницами руководств. Переменные окружения. Ввод и вывод, перенаправление. Шаблоны и фильтры.

---

### **2.Файловая система Linux**

Работа с файловой системой в режиме командной строки, перемещение по дереву каталогов, просмотр каталогов и файлов, индексный дескриптор, ссылки, виды ссылок, копирование и перемещение файлов; структура файловой системы операционных систем на базе ядра Linux.

---

### **3.Введение в локальные и глобальные сети**

Исторические предпосылки появления компьютерных сетей. Основные особенности локальных сетей, топологии. Глобальная сеть Интернет, история развития и перспектива, структура и принципы работы.

---

### **4.Знакомство с языком гипертекстовой разметки**

Язык гипертекстовой разметки HTML. Структура HTML-документа: понятия тега, контейнера, атрибута и его значения. Разметка текста, гипертекстовые ссылки, добавление графики, работа с таблицами, списки. Фон. Заголовок документа, мета-информация.

---

### **5.Введение в технологии Web**

Основные принципы создания Web-документов. Структура Web-страниц. Понятия меню, контента, сайта. Виды сайтов. Каркас сайта. Особенности Web-дизайна. Цветовые схемы. Проблемы использования различных шрифтов. Каскадные таблицы стилей (CSS): синтаксис, оформление фона, текста, списков и таблиц.

---

### **6.Использование профессиональных инструментов для создания растровой графики**

Знакомство с интерфейсом графического процессора и особенностями создания растровой графики. Выделение и рисование, градиенты и текстуры. Сохранение созданного изображения в различных форматах. Работа со слоями.

Проектная деятельность.

---

### **7.Изменение изображений**

Изменение размера, вращение и масштаб, отображение и др. Изменение цвета. Фильтры. Особенности работы с фотографиями.

Проектная деятельность.

---

### **8.Создание текстовых эффектов**

Особенности работы с текстом в графическом процессоре. Применение фильтров для создания различных текстовых эффектов.

Проектная деятельность.

---

### **9.Парадигмы программирования**

Различные уровни подхода к созданию программы. Особенности и различия структурного, функционального и объектно-ориентированного программирования. Рекурсия.

---

### **10.Основные принципы объектно-ориентированного программирования**

Понятия класса и экземпляра, наследования и полиморфизма. Множественное наследование. Многократное использование кода.

---

### **11.Практика объектно-ориентированного программирования**

Создание классов и их методов. Генерация объектов. Представление о перегрузке операторов. Документирование кода.

Проектная деятельность.

---

### **12.Обработка аудиоинформации**

Форматы файлов для хранения аудиоинформации. Редактирование звуковых файлов.

---

### **13.Обработка видеоинформации**

Форматы видеофайлов. Разрешение. Компьютерная обработка видео. Видеомонтаж. Фильтры для создания спецэффектов. Сжатие.

---

## **3.3 Третий год обучения**

### **1.История развития информационных технологий**

История компьютеров и сетей, операционных систем и прикладного программного обеспечения. Машины Бэббиджа и Тьюринга. Принципы построения компьютеров. Терминалы и суперкомпьютеры, первые программы. Предпосылки появления графического интерфейса пользователя.

---

### **2.Принципы работы в консольных редакторах**

Структура «окна», режимы: ввод текста, командный, редактирование и др. Команды переключения режимов, файловых операций, перемещения по файлу и его редактирования.

---

### **3.Знакомство с расширенными функциями консольных редакторов**

Работа с файловой системой, несколькими файлами. Поиск. Назначение редакторов. Настройка редакторов.

---

#### **4. Основы верстки публикаций**

Публикации и их виды. Инструменты для верстки документов. Работа с текстом, обработка изображений, работа со слоями. Экспорт в различные форматы. Дополнительные возможности программы верстки публикаций.

#### **5. Практика создания публикаций**

Особенности верстки статей, визиток, газет и др.

Проектная деятельность.

#### **6. Виды компьютерного моделирования**

Моделирование, виды моделирования, компьютерное моделирование. Субъект, объект, модель. Абстрагирование. Эксперимент. Преимущество компьютерного моделирования. Графическое компьютерное моделирование: особенности и назначение.

#### **7. Особенности трехмерной графики**

Геометрическая проекция. Понятие сцены. Объекты сцены: геометрия, материалы, источники света. Особенности представления трехмерных объектов.

#### **8. Практика 3D-моделирования**

Камеры и источники света, добавление и изменение объектов, проекции, материалы и текстуры, булевы операции. Формирование изображения.

Проектная деятельность.

#### **9. Инструменты для создания графического интерфейса пользователя**

Понятие графического интерфейса пользователя (GUI), его принципиального отличия от режима командной строки. История GUI. Многообразие инструментов для его создания, их особенности и использование.

#### **10. Знакомство с программированием основных элементов графического интерфейса**

Графическая библиотека к языку программирования. Графические компоненты: главное окно, кнопки, поля, флажки, списки, расположение объектов. Методы обработки действий пользователя.

#### **11. Разработка приложений с графическим интерфейсом**

Диалоговые окна. Работа с изображениями. Программирование меню. Методы редактирования текста.

Проектная деятельность.

#### **12. Введение в базы данных**

Общее определение базы данных, реляционная модель. Назначение баз данных. Типы данных. Поля и записи. Связи. Проектирование баз данных. Сокращение избыточности данных. Уникальность данных.

#### **13. Инструменты для работы с данными**

СУБД — системы управления базами данных: разнообразие, назначение. Создание и заполнение таблиц, установка между ними связей. Создание форм, поиск данных. Формирование запросов и отчетов. Сортировка.

#### **14. Разработка баз данных**

Особенности разработки и создания различных баз данных.

Проектная деятельность.

### 3.4 Четвертый год обучения

#### 1. Правовая охрана программ. Лицензирование ПО

Виды программного обеспечения по правовой принадлежности. Понятия «копирайта» и «копилефта». Коммерческое, свободное программное обеспечение. Вопросы общедоступности исходных кодов.

---

#### 2. Разнообразие операционных систем

Причины разнообразия операционных систем на базе ядра Linux. Понятие дистрибутива, версии и обновления. Другие операционные системы. Особенности операционных систем AltLinux, Debian и Fedora Core. Установка операционной системы.

---

#### 3. Знакомство с языком TeX

История возникновения, основное назначение. Особенности работы: исходный файл, программа-транслятор и др.

---

#### 4. Основы работы с издательской системой LaTeX

Основные понятия: специальные символы, команды, окружения, параметры. Набор и оформление текста и формул. Понятие псевдорисунка, отрезки, окружности, кривые и др. Таблицы. Создание команд. Блоки и клей.

---

#### 5. Практика использования LaTeX

Использование системы LaTeX в различных науках: математика, физика, химия, гуманитарные науки и др.

Проектная деятельность.

---

#### 6. Использование профессиональных инструментов для создания трехмерной графики

Знакомство с профессиональным инструментом для 3D-моделирования. Объекты, меш-объекты и их редактирование. Модификаторы. Булевы операции. Материалы и текстуры. Изображение в качестве фона. Освещение и камеры. Рендеринг.

---

#### 7. Основы 3D-анимации

Движение, вращение и масштабирование в условиях 3D-сцены. Ключевые кадры. Амнирование ламп и камер. Дополнительные возможности создания и редактирования анимации.

---

#### 8. Практика использования трехмерной графики в моделировании

Виртуальность. Особенности моделирование физических явлений и механизмов. Системы частиц и отражение лучей. Моделирование мягких тел. Моделирование в реальном времени.

---

#### 9. Введение в практическое программирование

Задачи, решаемые с помощью программирования. Основные принципы и этапы создания программного обеспечения. Многофайловые проекты. Тестирование, версии и обновления.

---

#### 10. Практика программирования

Обработка и сохранение данных. Стеки. Сортировка последовательностей. Анализ данных. Обработка текстовых данных, синтаксический анализ, регулярные выражения. Работа с файловой системой.

Проектная деятельность.

---

#### 11. Основы работы с электронными таблицами

Табличные процессоры и области их использования. Основные элементы

---

электронных таблиц (строка, столбец, активная и неактивная ячейки, заголовки). Принципы работы с электронными таблицами. Формулы, абсолютная и относительная адресация. Автоматизация. Функции. Математическое моделирование. Выполнение различных вычислений (суммирование, среднее значение и др.). Графическое представление данных (диаграммы). Поиск, сортировка и анализ данных.

---

## 4 Программное обеспечение

Представленное ниже программное обеспечение является рекомендуемым к программе дополнительного образования «Лаборатория юного линуксоида».

<i>Год обучения</i>	<i>Программное обеспечение</i>	<i>Пример</i>
1	Операционная система на базе ядра Linux	AltLinux
	Среды рабочего стола	KDE и Gnome
	Программы для работы с текстом	KWrite и OpenOffice.org Writer
	Программа для подготовки презентаций	OpenOffice.org Impress
	Графические редакторы	KoulorPaint и Inkscape
	Интерпретатор	Python
	Программа для создания 2D-анимации	Synfig
2	Операционная система на базе ядра Linux	AltLinux
	Среды рабочего стола	KDE
	Web-редактор	Quanta Plus
	Графический процессор	Gimp
	Интерпретатор	Python
	Редакторы аудио- и видеоданных	Audacity и Cinelerra
3	Операционная система на базе ядра Linux	AltLinux
	Среды рабочего стола	KDE
	Консольные редакторы	Vim и Emacs
	Программа для подготовки публикаций	Scribus
	Редактор трехмерной графики	KPovModeler
	Интерпретатор и библиотека компонентов графического интерфейса	Python и TkInter
	Программное обеспечение для разработки баз данных	OpenOffice.org Base
4	Операционные системы на базе ядра Linux и другие	AltLinux, Debian, Fedora и др.
	Среды рабочего стола	KDE
	Системы верстки	Tex и LaTeX
	Программа для создания трехмерной графики, анимации и моделирования	Blender
	Интерпретатор	Python
	Программа для работы с электронными таблицами	OpenOffice.org Calc

## 5 Информационное обеспечение

1. Википедия — свободная энциклопедия (<http://ru.wikipedia.org>)
2. СПО в российских школах (<http://freeschool.altlinux.ru/>)
3. BlenderУкраина. Русскоязычный сайт о Blender (<http://blender3d.org.ua/>)
4. BlenderWiki (<http://wiki.blender.org>)
5. GNU Image Manipulation Program (<http://docs.gimp.org/ru/>)
6. Linux по-русски. Виртуальная энциклопедия (<http://www.rus-linux.net/>)
7. Python Documentation (<http://python.org/doc/>)
8. Synfig animation studio (<http://synfig.org>)
  
9. Керниган Б., Пайк Р. UNIX. Программное окружение — Спб.: Символ-Плюс, 2003. - 416 с.: ил.
10. Лебланк Ди-Анн. Linux для «чайников» — М.: Диалектика, 2005 — 336 с.: ил.
11. Лутц М. Изучаем Python — Спб.: Символ-Плюс, 2003. - 848 с.: ил.
12. Монахов М.Ю и др. Учимся проектировать на компьютере. Практикум — М.: БИНОМ, 2005 — 172 с.: ил.
13. Семакин И.Г, Хеннер Е.К. Информационные системы и модели. Учебное пособие — М.: БИНОМ, 2005 — 303 с.: ил.
14. Шредер К. Linux. Сборник рецептов — Спб.: Питер, 2006. - 432 с.: ил.



## 6 Методическое обеспечение

### 6.1 Информационно-методическая поддержка

Информационно-методическую поддержку программы дополнительного образования детей «Лаборатория юного линуксоида» можно найти на сайте <http://www.younglinux.info>. Здесь опубликован текст самой программы, размещаются методические разработки занятий, презентации, проверочные тесты и др.

### 6.2 Тест по теме «Обработка текстовой информации»

#### 1. Чем отличается текстовый процессор от текстового редактора?

- ◆ Текстовый процессор обычно является коммерческим продуктом
- ◆ Текстовый процессор имеет расширенные функции для оформления текста
- ◆ Эти понятия обозначают одно и то же
- ◆ Текстовый процессор обычно содержит меньшее количество инструментов, чем редактор

#### 1. Какие существуют способы выравнивания текста по горизонтали?

- ◆ По левому, правому, верхнему и нижнему краям
- ◆ По центру, диагонали, левому краю
- ◆ По центру, левому и правому краю
- ◆ По центру, ширине, левому и правому краям

#### 1. Как называется текстовый процессор, входящий в состав пакета OpenOffice.org?

- ◆ Write
- ◆ Word
- ◆ Letters
- ◆ Writer

#### 1. Какой пункт строки меню отсутствует в Writer?

- ◆ Окно
- ◆ Формат
- ◆ Текст
- ◆ Справка

#### 1. Какие существуют ориентации листа?

- ◆ Книжная и альбомная
- ◆ Горизонтальная и вертикальная
- ◆ Широкая и узкая
- ◆ Плоская и объемная

#### 1. С помощью какой функциональной клавиши вызывается окно "Стили и форматирование"?

- ◆ F5

- ◆ F7
- ◆ F9
- ◆ F11

**1.Каких списков не бывает?**

- ◆ Многоуровневых
- ◆ Нумерованных
- ◆ Маркированных
- ◆ Иерархических

**1.Нажатие какой клавиши приводит к созданию нового абзаца?**

- ◆ Space
- ◆ Enter
- ◆ Tab
- ◆ Insert

**1.С помощью каких "горячих клавиш" можно выполнить копирование текста?**

- ◆ ALT+X
- ◆ CTRL+X
- ◆ CTRL+C
- ◆ SHIFT+C

**1. Что обозначает кнопка на панели инструментов с изображением ножниц?**

- ◆ Вырезать
- ◆ Сократить
- ◆ Обрезать
- ◆ Разделить

**1. Какого способа объединить несколько ячеек таблицы не существует?**

- ◆ Выделить ячейки и в меню выбрать Таблица -> Объединить ячейки
- ◆ Выделить ячейки и нажать клавишу Backspace
- ◆ Выделить ячейки, вызвать контекстное меню, в нем выбрать Ячейки -> Объединить
- ◆ Выделить ячейки и на панели инструментов "Таблица" нажать кнопку "Объединить ячейки"

**1. Для чего предназначены стили?**

- ◆ Только стили позволяют красочно оформить документ
- ◆ Применение стилей позволяет придать документу логическую структуру
- ◆ Стиль документа указывается при его сохранении, чтобы любой пользователь мог понять, в какой программе документ был создан.
- ◆ В Writer нет такого понятия как "стиль"

**1. Перед тем как добавить нумерацию страниц в Writer, необходимо вставить ...**

- ◆ Примечания

- ◆ Сноски
- ◆ Колонтитулы
- ◆ Поля

#### 1. Какой характеристики нет у абзаца?

- ◆ Фон
- ◆ Выравнивание
- ◆ Междустрочный интервал
- ◆ Ширина

#### 1. Какие две панели инструментов обычно по-умолчанию всегда отображены в Writer?

- ◆ Стандартная и Форматирование
- ◆ Стандартная и Рисование
- ◆ Форматирование и Таблица
- ◆ Выравнивание и Маркеры и нумерация

### 6.3 Разработка занятия к теме «Принципы работы в графических средах»

Тема занятия:

#### Файловая структура

Продолжительность: 45 мин

План занятия:

1. Новый материал
  - Что такое файл и каталог?
  - Получаем дерево?!
  - Каждому файлу - адрес!
  - Файловый менеджер – ваш помощник.
  - Навигация и создание файлов
2. Практическая работа. “Создание каталогов и файлов”
3. Вопросы
4. Выводы

Ход занятия:

#### Новый материал

##### Что такое файл и каталог?

Одно из назначений компьютера – это долговременное хранение данных, которые чаще всего сохраняются на жестком диске. Выясним, каким образом организовано их хранение.

Данные – это любая информация. Данными можно назвать текст, рисунок, таблицу.

Сохраняя определенные данные на диске, мы предполагаем, что когда-нибудь они нам понадобятся снова. Но как их потом найти? Необходимо знать, где были сохранены эти данные, а также имя, по которому к ним обращаться. Для сохранения адреса и имени области диска, где хранятся конкретные данные, было введено понятие файла. Итак, **файл – это именованное место на диске для хранения данных определенного типа.**

Файлов может быть слишком много и они могут находиться в разных местах жесткого диска. Однако человеку ориентироваться в беспорядочном размещении файлов вряд ли будет удобно. Для того, чтобы упорядочить и систематизировать файлы, был придуман **особый тип файла – каталог**, который представляет собой список ссылок на файлы с данными или другие каталоги. Если в операционной системе открыть каталог, то можно увидеть файлы и другие каталоги, которые как бы в нем хранятся. На самом деле эти вложенные файлы могут находиться в разных местах жесткого диска, но операционная система покажет их вместе.

Часто каталоги называют также директориями или папками. Все три термина обозначают одно и то же.

### **Получаем дерево?!**

Итак, каждый файл в операционной системе должен находиться в каталоге, а также, каждый каталог (за одним исключением) должен находиться в другом каталоге. Исключением является так называемый “корневой каталог”, с которого все начинается. Кроме того, не может быть ситуации, когда один каталог ссылается на второй, а тот, обратно, на первый.

Описанное выше можно представить в виде дерева (рис. 1), у которого ствол и ветви являются каталогами, а листья – файлами. От любой ветви может отрастать ветка следующего порядка и/или листья, т.е. любой каталог может содержать вложенные каталоги и/или файлы с данными.

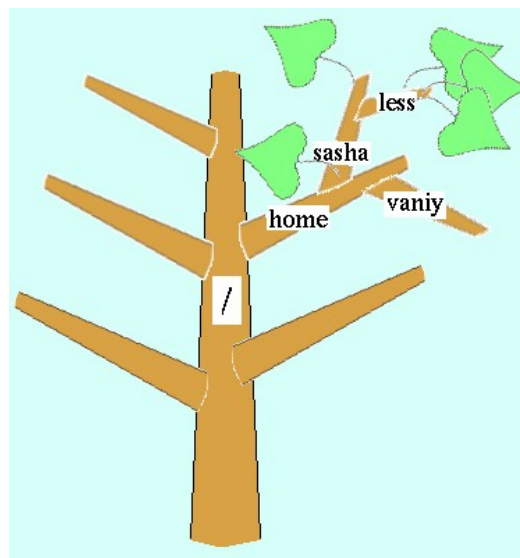


Рис. 1. Древоподобная структура файловой системы

Можно сделать следующий вывод: **файловая система в операционной системе – это логическая структура файлов и каталогов.** Структуру, описанную выше, часто называют иерархической или древоподобной. Она характерна для большинства операционных

систем. Однако существуют и некоторые отличия. Так в операционных системах, основанных на ядре Linux, существует лишь один корневой каталог. Обозначается он символом слеша (/) и содержит больше десятка вложенных каталогов. В ОС семейства Windows корневых каталогов столько, сколько разделов жесткого диска доступно системе.

### ***Каждому файлу – адрес!***

Из вышеописанной структуры можно заключить, что у каждого файла и каталога должен быть свой адрес.

На рисунке 1 можно видеть, что каталог less находится в каталоге sasha. Тот в свою очередь в каталоге home, который находится непосредственно в корневом каталоге. Адрес записывается следующим образом: сначала пишется корневой каталог «/», и далее указываются каталоги, начиная с самого верхнего (родительского) и заканчивая самым нижним (дочерним), разделяемые между собой все тем же «/». В нашем примере адрес каталога less, будет выглядеть так: /home/sasha/less. **Адрес и имя файла, формируют его полное имя.** При этом следует запомнить правило: в операционной системе не может быть двух файлов (или каталогов) с одинаковыми полными именами.

**Адресация, описанная выше, является абсолютной** (т.е. адрес записывается, начиная с корневого каталога). Помимо абсолютной нередко используют **относительную адресацию**, где запись адреса начинается не с корневого каталога, а относительно текущего (того, который открыт в данный момент). Например, если мы, находясь в каталоге sasha (рис. 1), захотим перейти в каталог vaniy, то следует написать такой адресный путь: ../vaniy. Здесь две точки обозначают переход на уровень выше (в данном случае каталог /home). Если из каталога sasha потребуется перейти в каталог less, то достаточно будет указать просто имя этого каталога less; хотя более правильным будет такой вариант: ./less. Одна точка обозначает текущий каталог.

### ***Файловый менеджер – ваш помощник***

Как перемещаться по дереву каталогов, просматривать их содержимое и управлять им? Для этих целей используется так называемые файловые менеджеры (современные менеджеры графических оболочек называют также браузерами, т.к. они позволяют просматривать некоторые типы файлов).

В графической оболочке GNOME по умолчанию таким менеджером/браузером является **Nautilus**. Запустить его можно через Главное меню (Приложения -> Система -> Обозреватель файлов) или другими способами.

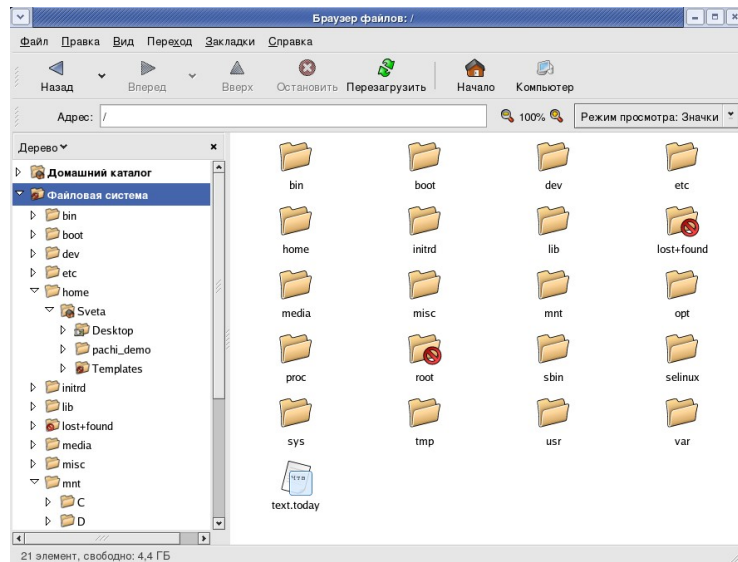


Рис. 2. Браузер Nautilus (Наутилус)

Опишем структуру окна данного браузера и основные принципы навигации в нем.

В окне Наутилуса можно выделить пять частей. Первая, – строка меню (содержит пункты Файл, Правка, Вид и т.д.) - где осуществляется доступ к командам браузера. Вторая – панель инструментов (кнопки Назад, Вперед, Вверх и т.д.) - содержит наиболее используемые команды меню, реализованные в виде кнопок. Третья – адресная строка – содержит поле, отображающее адрес текущего каталога. Четвертая (слева) – это боковая панель – в данном случае отображающее дерево каталогов. И пятая (самая большая часть) – это область просмотра – отображает содержимое текущего каталога.

Остановимся на боковой панели, отображающей дерево каталогов (помимо этого панель может быть настроена на отображения сведений, заметок и истории текущего каталога). Знак перевернутого треугольника, расположенный у каталога обозначает, что в нем находятся другие каталоги и/или файлы. Если нажать на этот знак, то отобразятся вложенные в него каталоги. Повторное нажатие закроет структуру каталога. Если щелкнуть левой кнопкой мыши непосредственно по каталогу, то его содержимое отобразится в области просмотра.

Кнопка Назад на Панели инструментов возвращает нас в каталог, который был текущим до этого. Кнопка Вверх позволяет перейти в родительский каталог по отношению к текущему. Кнопка Перезагрузить обновляет вид содержимого текущего каталога (полезна в случаях, когда известно, что в каталоге произошли изменения, но браузер их еще не отобразил). Кнопка Домой делает текущим домашний каталог пользователя (который находится по адресу /home/имя\_пользователя).

Помимо Наутилуса существует много других файловых менеджеров. Например, в графической оболочке KDE по умолчанию им является Konqueror.

### **Навигация и создание файлов**

Файлы и папки можно создавать несколькими способами. Одним из них является создание с помощью контекстного меню. Для этого необходимо щелкнуть в пустое место текущего каталога правой кнопкой мыши и в контекстном меню выбрать пункт **Создать папку** или **Создать документ** (в последнем случае появляется подменю, где выбирается тип документа). После этого вводится имя объекта.

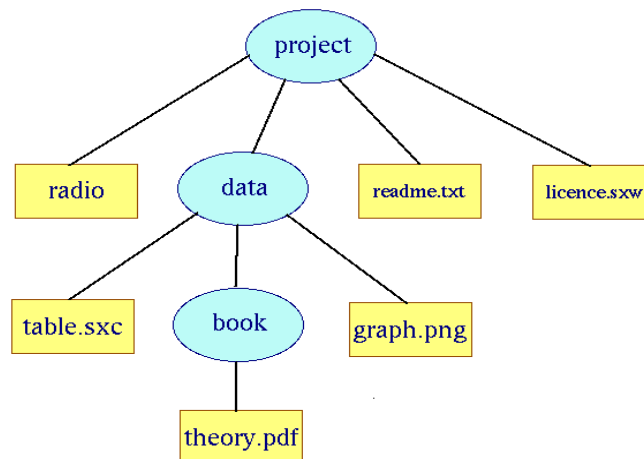
Если понадобится переименовать объект, то это можно сделать следующими способами:

- выделить папку/файл, затем щелкнуть по нему левой кнопкой мыши (не путать с двойным кликом, когда щелчки происходят быстро друг за другом);
- выделить папку/файл, затем нажать на клавиатуре клавишу F2;
- кликнуть папку/файл правой кнопкой мыши и в появившемся контекстном меню выбрать пункт **Переименовать**.

После любого из этих трех действий следует ввести новое имя файла и нажать Enter.

## Практическая работа. Создание файлов и каталогов

Задание. Создайте в своем домашнем каталоге следующую структуру.



Обозначения: эллипс – каталог, прямоугольник – файл.

## Вопросы

1. Почему файловую систему называют древовидной и иерархической?
2. Что такое файл?
3. Как обозначается корневой каталог в Linux? В каком каталоге по умолчанию содержатся каталоги пользователей?
4. Что такое абсолютный адрес? относительный адрес? В каких случаях более уместно употреблять относительный адрес?

## Выводы

- Файлы представляют собой именованные места на диске для хранения данных. Структура данных может быть различна.
- Структура каталогов операционных систем устроена по иерархическому принципу и имеет древовидную структуру.
- Каждый объект файловой структуры имеет свой уникальный адрес, запись которого начинается от корневого каталога и заканчивается искомым объектом (файлом или каталогом). Переходить по структуре каталогов можно как с помощью указания абсолютных адресов, так и относительных.
- В операционных системах предусмотрены различные файловые менеджеры для просмотра и управления структурой файлов и каталогов.
- Помимо системных файлов и каталогов человек может создавать и собственные. Создавать папки и файлы можно как в графическом режиме, так и с помощью командной оболочки в текстовом режиме.