# тема 4.3. UNIX-ПОДОБНЫЕ И ДРУГИЕ posix-СОВМЕСТИМЫЕ операционные системы

В данной теме рассматриваются следующие вопросы:

* Общая характеристика семейства операционных систем UNIX
* Стандарты UNIX.
* История создания UNIX.
* Основные понятия системы UNIX.
* Пользователи системы, атрибуты пользователя.
* Файловая структура ОС.
* Основные свойства UNIX.
* Мобильность операционных систем.
* Методы инсталляции и настройки ОС.
* Особенности процессов.
* Сигналы.
* Обработка сигналов.
* Неименованные каналы.
* Именованные каналы.
* Архитектура ОС UNIX.
* Принципы организации и структура ОС.
* Ядро и основные компоненты ОС.
* iOS.
* ОС Android.

Лекции – 2 часа, лабораторные занятия – 2 часа, самостоятельная работа – 2 часа.

Экзаменационные вопросы по теме:

* Основные понятия системы UNIX. Пользователи системы, атрибуты пользователя. Файловая структура ОС.
* ОС UNIX: особенности процессов, сигналы, обработка сигналов.

### 4.1.1. Общая характеристика семейства операционных систем UNIX

Операционная система UNIX всегда была интерактивной системой, разработанной для одновременной поддержки множества процессов и множества пользователей. Она была разработана программистами и для программистов — чтобы использовать ее в такой среде, в которой большинство пользователей достаточно опытны и занимаются проектами (часто довольно сложными) разработки программного обеспечения. Во многих случаях большое количество программистов активно работает над созданием общей системы, поэтому в операционной системе UNIX есть большое количество средств, позволяющих людям работать вместе и управлять совместным использованием информации. Очевидно, что модель группы опытных программистов, совместно работающих над созданием сложного программного обес печения, существенно отличается от модели одного начинающего пользователя, сидящего за персональным компьютером в текстовом процессоре, и это отличие отражается в операционной системе UNIX от начала до конца. Совершенно естественно, что Linux унаследовал многие из этих установок, даже несмотря на то что первая версия предназначалась для персонального компьютера.

Чего действительно хотят от операционной системы хорошие программисты? Прежде всего, большинство хотело бы, чтобы их система была простой, элегантной и совместимой.

Другие свойства, которые, как правило, опытные программисты желают видеть в операционной системе, — это мощь и гибкость. Это означает, что в системе должно быть небольшое количество базовых элементов, которые можно комбинировать, чтобы приспособить их для конкретного приложения.

Наконец, у большинства программистов есть сильная неприязнь к бесполезной избыточности. Зачем писать **copy**, когда вполне достаточно **cp**, чтобы однозначно выразить желаемое? Это же пустая трата драгоценного времени.

### 4.3.2. Стандарты UNIX

К концу 1980-х широкое распространение получили две различные и в чем-то несовместимые версии системы UNIX : 4.3BSD и System V Release 3. Кроме того, практически каждый производитель добавлял собственные нестандартные усовершенствования. Этот раскол в мире UNIX вместе с тем фактом, что стандарта на формат двоичных программ не было, сильно сдерживал коммерческий успех операционной системы UNIX, поскольку производители программного обеспечения не могли написать пакет программ для системы UNIX так, чтобы он мог работать на любой системе UNIX (как это делалось, например, в системе MS-DOS). Вначале все попытки стандартизации системы UNIX проваливались. Например, корпорация AT&T выпустила стандарт SVID (System V Interface Definition — описание интерфейса UNIX System V), в котором определялись все системные вызовы, форматы файлов и т. д. Этот документ был попыткой построить в одну шеренгу всех производителей System V, но он не оказал никакого влияния на вражеский лагерь (BSD), который просто проигнорировал его [TA].

Первая серьезная попытка примирить два варианта системы UNIX была предпринята при содействии Совета по стандартам при Институте инженеров по электротехнике и электронике (IEEE Standard Boards), глубокоуважаемой и, что самое важное, нейтральной организации. В этой работе приняли участие сотни людей из области промышленности, академических и правительственных организаций. Коллективное название этого проекта — POSIX . Первые три буквы этого сокращения означали Portable Operating System — переносимая операционная система. Буквы IX в конце слова были добавлены, чтобы имя проекта выглядело юниксоподобно.

После большого количества высказанных аргументов и контраргументов, опровержений и опровергнутых опровержений комитет POSIX выработал стандарт, известный как 1003.1. Этот стандарт определяет набор библиотечных процедур, которые должна обеспечивать каждая соответствующая данному стандарту система UNIX. Большая часть этих процедур делает системный вызов, но некоторые из них могут быть реализованы вне ядра. Типичными процедурами являются open, read, и fork. Идея стандарта POSIX заключается в том, что производитель программного обеспечения, который при написании программы использует только описанные в стандарте 1003.1 процедуры, может быть уверен, что его программа будет работать на любой системе UNIX, соответствующей данному стандарту.

Комитет IEEE взял за основу пересечение множеств функциональных возможностей System V и BSD. То есть дело обстояло примерно так: если какое-либо свойство присутствовало как в System V, так и в BSD, то оно включалось в стандарт, в противном случае это свойство в стандарт не включалось. Документ 1003.1 был написан так, чтобы как разработчики операционной системы, так и создатели программного обеспечения были способны его понять, что также было ново в мире стандартов, хотя в настоящее время уже ведется работа по исправлению этого нестандартного для стандартов свойства.

Несмотря на то что стандарт 1003.1 описывает только системные вызовы, принят также ряд сопутствующих документов, которые стандартизируют потоки, утилиты, сетевое программное обеспечение и многие другие функции системы UNIX. Кроме того, язык C также был стандартизирован Национальным институтом стандартизации США (ANSI) и Международной организацией по стандартизации ISO.

### 4.3.3. История создания UNIX

Одной из операционных систем второго поколения была MULTICS, совместно созданная MIT, Bell Labs и General Electrics. Кен Томпсон (Ken Thompson) написал на ассемблере усеченный вариант системы MULTICS для компьютера PDP-7. Брайан Керниган (Brian Kernighan), как-то в шутку назвал эту систему UNICS (UNiplexed Information and Computing Service — примитивная информационная и вычислительная служба). Несмотря на все каламбуры и шутки, эта кличка прочно пристала к новой системе, хотя написание этого слова позднее превратилось в UNIX [TA].

Вскоре к Кену Томпсону присоединился Деннис Ритчи (Dennis Ritchie), а чуть позднее и весь его отдел. К этому времени относятся два технологических усовершенствования. Во-первых, система UNIX была перенесена с устаревшей машины PDP-7 на гораздо более современный компьютер PDP-11/20, а позднее на PDP-11/45 и PDP-11/70. Второе усовершенствование касалось языка, на котором была написана операционная система UNIX. Ритчи разработал язык, ставший преемником языка B, который, получил название C, и написал для него прекрасный компилятор. Томпсон и Ритчи совместно переписали UNIX на языке C. Язык C оказался как раз тем языком, который и был нужен в то время, и с тех пор он сохраняет лидирующие позиции в области системного программирования.

Операционная система UNIX быстро завоевала популярность в университетах. — и не в последнюю очередь благодаря тому, что система поставлялась с полными исходными кодами, поэтому новые владельцы системы могли без конца подправлять и совершенствовать ее (и они делали это). Операционной системе UNIX было посвящено множество научных симпозиумов, на них докладчики рассказывали о тех неизвестных ошибках в ядре, которые им удалось обнаружить и исправить. В результате всех этих событий новые идеи и усовершенствования системы распространялись с огромной скоростью.

Version 7 стала первой переносимой версией операционной системы UNIX (она работала как на машинах PDP-11, так и на Interdata 8/32). Эта версия системы состояла уже из 18 800 строк на языке C и 2100 строк на ассемблере. На Version 7 выросло целое поколение студентов, которые, закончив свои учебные заведения и начав работу в промышленности, содействовали дальнейшему ее распространению. К середине 1980-х годов операционная система UNIX широко применялась на мини-компьютерах и инженерных рабочих станциях самых различных производителей.

Первым коммерческим вариантом системы UNIX была System III. Ее выход на рынок был не очень успешным, поэтому через год она была заменена улучшенной версией, System V. Агрессивная маркетинговая политика AT&T по продвижению System V какое-то время препятствовала распространению новых версий. Но впоследствии она продала свой связанный с UNIX бизнес, а основные компьютерные компании уже имели лицензии. Появилось большое количество несовместимых между собой версий, в связи с чем возникли стандарты POSIX, описанные в предыдущем разделе.

Университет в Беркли разработал и выпустил улучшенную версию операционной системы UNIX для мини-компьютера PDP-11, названную 1BSD. Затем последовали версии 2BSD, 4BSD. Университет в Беркли также добавил в систему UNIX значительное количество утилит, включая новый редактор vi и новую оболочку csh, компиляторы языков Pascal и Lisp и многое другое. Все эти усовершенствования привели к тому, что многие производители компьютеров (Sun Microsystems, DEC и др.) стали основывать свои версии системы UNIX на Berkeley UNIX, а не на «официальной» версии System V компании AT&T. В результате Berkeley UNIX получила широкое распространение в академических и исследовательских кругах, а также в Министерстве обороны.

В 1987 году была выпущена операционная система MINIX. Эта система состояла из 11 800 строк на языке C и 800 строк кода на ассемблере и была одной из первых юниксоподобных систем, основанной на микроядре. Идея микроядра заключается в том, чтобы реализовать в ядре как можно меньше функций и сделать его надежным и эффективным. Соответственно управление памятью и файловая система были перенесены в процессы пользователя. Уже через несколько месяцев после своего появления система MINIX стала чем-то вроде объекта культа — со своей группой новостей comp.os.minix и более чем 40 000 пользователей. Очень многие пользователи стали сами писать команды и пользовательские программы, так что система MINIX быстро стала продуктом коллективного творчества большого количества пользователей по всему Интернету, послужив прототипом для других коллективных проектов, появившихся позднее. В 1997 году была выпущена версия 2.0 системы MINIX. Теперь базовая система включала в себя сетевое программное обеспечение, и ее размер вырос до 62 200 строк.

Финский студент Линус Торвальдс (Linus Torvalds) решил сам написать еще один клон системы UNIX, который он назвал Linux. Это должна была быть полноценная производственная система, со многими изначально отсутствовавшими в системе MINIX функциями. Первая версия 0.01 операционной системы Linux была выпущена в 1991 году. Она была разработана и собрана на компьютере под управлением MINIX и заимствовала из системы MINIX множество идей, начиная со структуры дерева исходных кодов и заканчивая компоновкой файловой системы. Однако в отличие от микроядерной системы MINIX, Linux была монолитной системой, то есть вся операционная система размещалась в ядре. Размер исходного текста составил 9300 строк на языке C и 950 строк на ассемблере, что приблизительно совпадало с версией MINIX как по размеру, так и по функциональности. Фактически это была переделка системы MINIX — единственной системы, исходный код которой имелся у Торвальдса.

Операционная система Linux быстро росла в размерах и впоследствии развилась в полноценный клон UNIX с виртуальной памятью, более сложной файловой системой и многими другими дополнительными функциями. Хотя изначально система Linux работала только на процессоре Intel 386 (и даже имела встроенный ассемблерный код 386-го процессора в процедурах на языке C), она была быстро перенесена на другие платформы и теперь работает на широком спектре машин — так же, как и UNIX. Следует выделить одно отличие системы Linux от UNIX: она использует многие специальные возможности компилятора gcc, поэтому потребуется приложить немало усилий, чтобы откомпилировать ее стандартным ANSI C-компилятором. История операционной системы Linux будет рассмотрена в следующей теме.

Когда мода на Linux начала набирать обороты, она получила поддержку с неожиданной стороны — от корпорации AT&T. В 1992 году университет в Беркли, лишившись финансирования, решил прекратить разработку BSD UNIX на последней версии 4.4BSD (которая впоследствии послужила основой для FreeBSD). Поскольку эта версия по существу не содержала кода AT&T, университет в Беркли выпустил это программное обеспечение с лицензией открытого исходного кода, которая позволяла всем делать все, что угодно, кроме одной вещи — подавать в суд на университет Калифорнии. Контролировавшее систему UNIX подразделение корпорации AT&T отреагировало немедленно, подав в суд на университет Калифорнии и компанию BSDI. Хотя этот спор в конечном итоге удалось урегулировать в досудебном порядке, он не позволял выпустить на рынок FreeBSD в течение долгого периода — достаточного для того, чтобы система Linux успела упрочить свои позиции.

Если бы судебного иска не было, то уже примерно в 1993 году началась бы серьезная борьба между двумя бесплатными версиями системы UNIX, распространяющимися с исходными кодами: царствующим чемпионом — системой BSD (зрелой и устойчивой системой с многочисленными приверженцами в академической среде еще с 1977 года) и энергичным молодым претендентом — системой Linux всего лишь двух лет от роду, но с уже растущим числом последователей среди индивидуальных пользователей. Кто знает, чем обернулась бы эта схватка двух бесплатных версий системы UNIX.

### 4.3.4. Основные понятия системы UNIX

Операционную систему UNIX можно рассматривать как пирамиду (рис. 4.3.d). У основания пирамиды располагается аппаратное обеспечение, состоящее из центрального процессора, памяти, дисков, монитора и клавиатуры, а также других устройств. Операционная система работает на «голом железе». Ее функция заключается в управлении аппаратным обеспечением и предоставлении всем программам интерфейса системных вызовов. Эти системные вызовы позволяют программам пользователя создавать процессы, файлы и прочие ресурсы, а также управлять ими.

Программы делают системные вызовы, помещая аргументы в регистры (или иногда в стек) и выполняя команду эмулированного прерывания для переключения из пользовательского режима в режим ядра. Поскольку на языке C невозможно написать команду эмулированного прерывания, то этим занимается библиотека, в которой есть по одной процедуре на системный вызов. Эти процедуры написаны на ассемблере, но они могут вызываться из языка C. Каждая такая процедура сначала помещает аргументы в нужное место, а затем выполняет команду эмулированного прерывания. Таким образом, чтобы обратиться к системному вызову read, программа на языке C должна вызвать библиотечную процедуру read. Кстати, в стандарте POSIX определен именно интерфейс библиотечных функций, а не интерфейс системных вызовов. Иначе говоря, стандарт POSIX определяет, какие библиотечные процедуры должна предоставлять соответствующая его требованиям система, каковы их параметры, что они должны делать и какие результаты возвращать. В стандарте даже не упоминаются реальные системные вызовы.



Рис. 4.3.d. Уровни операционной системы Unix

В операционной системе содержится большое количество стандартных программ, некоторые из них указаны в стандарте POSIX 1003.2, тогда как другие могут различаться от версии к версии. К этим программам относятся командный процессор (оболочка), компиляторы, редакторы, программы обработки текста и утилиты для работы с файлами. Именно эти программы и запускает пользователь с клавиатуры.

Графические интерфейсы пользователя поддерживает оконная система X Windowing System, которую обычно называют Х11 (или просто Х). Она определяет обмен и протоколы отображения для управления окнами на растровых дисплеях UNIX-подобных систем. Х-сервер является главным компонентом, который управляет такими устройствами, как клавиатура, мышь и экран, и отвечает за перенаправление ввода или прием вывода от клиентских программ. Реальная среда графического интерфейса пользователя обычно построена поверх библиотеки низкого уровня (xlib), которая содержит функциональность для взаимодействия с Х-сервером. Графический интерфейс расширяет базовую функциональность Х11, улучшая вид окон, предоставляя кнопки, меню, значки и пр. Х-сервер можно запустить вручную из командной строки, но обычно он запускается во время загрузки диспетчером окон, который отображает графический экран входа в систему.

### 4.3.5. Пользователи системы, атрибуты пользователя

Абзац 1.

Абзац 2.

### 4.3.6. Файловая структура ОС

Файловую структуру UNIX рассмотрим на примере первых файловых систем Linux (в теме 3.3 уже были рассмотрены некоторые файловые системы).

Первоначально файловой системой в Linux была файловая система MINIX 1. Однако из-за того обстоятельства, что имена файлов были ограничены в ней 14 символами (чтобы поддерживать совместимость с UNIX Version 7), а максимальный размер файла составлял 64 Мбайт (что было даже слишком много для жестких дисков того времени, размер которых составлял 10 Мбайт), интерес к более совершенным файловым системам появился сразу же после начала разработки системы Linux (которая началась примерно через 5 лет после выпуска MINIX 1). Первым улучшением стала файловая система ext, которая позволяла использовать имена файлов длиной 255 символов и размер файлов 2 Гбайт (однако она была медленнее, чем файловая система MINIX 1, так что поиски продолжались еще некоторое время). В итоге была изобретена файло-10.6. Файловая система ext2 (с длинными именами файлов, большими файлами и более высокой производительностью), которая и стала основной файловой системой. Однако Linux поддерживает несколько десятков файловых систем при помощи уровня виртуальной файловой системы Virtual File System (VFS), описанного в следующем разделе. Когда система Linux собирается, вам предлагается сделать выбор тех файловых систем, которые будут встроены в ядро. Другие можно загружать динамически (как модули) во время выполнения (если будет такая необходимость).

Файл в системе Linux — это последовательность байтов произвольной длины (от 0 до некоторого максимума), содержащая произвольную информацию. Не делается различия между текстовыми (ASCII) файлами, двоичными файлами и любыми другими типами файлов. Значение битов в файле целиком определяется владельцем файла. Системе это безразлично. Имена файлов ограничены 255 символами. В именах файлов разрешается использовать все ASCII-символы, кроме символа NUL, поэтому допустимо даже состоящее из трех символов возврата каретки имя файла (хотя такое имя и не слишком удобно в использовании).

По соглашению многие программы ожидают, что имена файлов будут состоять из основного имени и расширения, разделенных точкой (которая также считается символом). Так, prog.c — это обычно программа на языке C, prog.py — это обычно программа на языке Python, а prog.o — чаще всего объектный файл (выходные данные компилятора). Эти соглашения никак не регламентируются операционной системой, но некоторые компиляторы и другие программы ожидают файлов именно с такими расширениями. Расширения имеют произвольную длину, причем файлы могут иметь по несколько расширений, например prog.java.Z, что, скорее всего, представляет собой сжатую программу на языке Java.

Для удобства файлы могут группироваться в каталоги. Каталоги хранятся на диске в виде файлов, и с ними можно работать практически так же, как с файлами. Каталоги могут содержать подкаталоги, что приводит к иерархической файловой системе. Корневой каталог называется / и всегда содержит несколько подкаталогов. Символ / используется также для разделения имен каталогов, поэтому имя /usr/ast/x обозначает файл x, расположенный в каталоге ast, который в свою очередь находится в каталоге usr. Некоторые основные каталоги (находящиеся у вершины дерева каталогов) показаны в табл. 4.3.ы.

**Таблица 4.3.ы**. Некоторые важные каталоги, существующие в большинстве систем Linux

|  |  |
| --- | --- |
| Каталог | Содержание |
| bin | Двоичные (исполняемые) программы |
| dev | Специальные файлы для устройств ввода-вывода |
| etc | Разные системные файлы |
| lib | Библиотеки |
| usr | Каталоги пользователей |

848

Существует два способа задания имени файла в системе Linux (как в оболочке, так и при открытии файла из программы). Первый способ заключается в использовании **абсолютного пути** (absolute path), указывающего, как найти файл от корневого каталога. Также имена путей могут указываться относительно рабочего каталога. Путь, заданный относительно рабочего каталога, называется **относительным путем** (relative path).

Кроме обычных файлов Linux поддерживает также символьные специальные файлы и блочные специальные файлы. Символьные специальные файлы используются для моделирования последовательных устройств ввода-вывода, таких как клавиатуры и принтеры. Блочные специальные файлы (обычно с такими именами, как /dev/hd1) могут использоваться для чтения и записи необработанных дисковых разделов, минуя файловую систему. При этом поиск байта номер k, за которым последует чтение, приведет к чтению k-го байта из соответствующего дискового раздела, игнорируя i-узел и файловую структуру.

Для безопасного совместного доступа к файлам стандарт POSIX предоставляет гибкий и детальный механизм, позволяющий процессам за одну неделимую операцию блокировать даже единственный байт файла (или целый файл). Механизм блокировки требует от вызывающей стороны указать блокируемый файл, начальный байт и количество байтов. Если операция завершается успешно, то система создает запись в таблице, в которой указывается, что определенные байты файла (например, запись базы данных) заблокированы.

Стандартом определены два типа блокировки: блокировка с монополизацией (exclusive locks) и блокировка без монополизации (shared locks). Если часть файла уже имеет блокировку без монополизации, то повторная попытка установки блокировки без монополизации на это место файла разрешается, но попытка установить блокировку с монополизацией будет отвергнута. Если же какая-либо область файла содержит блокировку с монополизацией, то любые попытки заблокировать любую часть этой области файла будут отвергаться, пока не будет снята блокировка. Для успешной установки блокировки необходимо, чтобы каждый байт в блокируемой области был доступен.

### 4.3.7. Основные свойства UNIX

Абзац 1.

Абзац 2.

### 4.3.8. Мобильность операционных систем

Абзац 1.

Абзац 2.

### 4.3.9. Методы инсталляции и настройки ОС

Абзац 1.

Абзац 2.

### 4.3.10. Особенности процессов

Абзац 1.

Абзац 2.

### 4.3.11. Сигналы

Абзац 1.

Абзац 2.

### 4.3.12. Обработка сигналов

Абзац 1.

Абзац 2.

### 4.3.13. Неименованные каналы

Абзац 1.

Абзац 2.

### 4.3.14. Именованные каналы

Абзац 1.

Абзац 2.

### 4.3.15. Архитектура ОС UNIX

Абзац 1.

Абзац 2.

### 4.3.16. Принципы организации и структура ОС

Абзац 1.

Абзац 2.

### 4.3.17. Ядро и основные компоненты ОС

Абзац 1.

Абзац 2.

### 4.3.18. MacOS

Mac OS (Macintosh Operating System) — семейство проприетарных операционных систем с графическим интерфейсом. Apple хотела, чтобы Макинтош представлялся как компьютер «для всех». Сам термин «Mac OS» в действительности не существовал до тех пор, пока не был официально использован в середине 1990-х годов.

Компания Apple была так же первая, кто придумал и использовал компьютерную мышь. Которая стала очень популярным устройством.

По данным компании Net Applications, в июле 2022 года рыночная доля составляла: 4.86%

**История ОС Mac OS**

* Mac OS вышла в свет в 1984 году вместе с первым персональным компьютером Macintosh от компании Apple. Идеи, воплощенные в первой версии системы Mac OS, ее авторы почерпнули у фирмы Xerox.
* Пользователи управляли своим компьютером не только вводимыми с клавиатуры командами и инструкциями, но и с помощью нового в те времена устройства, названного мышью.
* В исследовательском центре Xerox PARC в то время уже существовал компьютер с графической операционной системой, что тогда было настоящим прорывом в эволюционном развитии операционных систем. Но они использовали ее только для собственных нужд и не планировали коммерческого применения.
* Соединив уже имеющиеся наработки и собственные идеи, программисты компании Apple создали Mac OS, первую доступную для всех графическую операционную систему. В ней уже тогда был использован всем нам привычный оконный интерфейс, папки с файлами, и впервые был применен манипулятор, названный компьютерной мышью, способный передвигать курсор по всей области экрана. Такая концепция вполне соответствовала главной идее самой компании Apple, предлагавшей создать компьютер доступный для всех, как по цене, так и в техническом плане.

**Первая версия ОС Mac OS X**

* Первая версия Mac OS занимала всего 216 кб дискового пространства и работала даже при обычном копировании с одного компьютера на другой. Но такой продукт был совершенно не защищен от подделки, поэтому для того, чтоб сохранить свои доходы разработчики все дальнейшее время посвятили не только ее техническому усовершенствованию, расширению функциональности и стабильности, но и защите.

**MacOS X**

* Теперь в Mac OS X используется ядро Mach, стандартные сервисы BSD и все основные возможности операционной системы Unix. Это дало возможность в много раз повысить ее функциональность, защищенность и стабильность. Вытесняющая многозадачность, которая используется в Mac OS X, позволяет работать нескольким процессам сразу, но при этом не мешать друг другу, а при сбое в работы одного из них не допускать сбоя всей системы и прерывания работы других процессов.
* После выхода версии Mac OS X 10.0 было выпущено еще шесть ее модификаций, каждая из которых носит название животного из семейства кошачьих.(**Cheetah (Гепард),Puma, Jaguar, Tiger, Leopard, Snow Leopard))**
* На данный момент Mac OS X имеет собственный красивый, не перегруженный спецэффектами и приятный для глаз интерфейс Aqua. Она проста в использовании и дружелюбна. В ней используется среда программирования Core Foundation. К тому же Mac OS X позволяет использовать программное обеспечение, написанное на таких языках программирования, как Си, C++, Objective-C, Ruby и Java. Немаловажным достоинством Mac OS X является ее безопасность при работе в интернете, она неплохо защищена от интернет-атак, да и количество вирусов способных ее поразить на сегодняшний день ничтожно мало.

**Особенности для программиста**

**1. Unix-подобная ОС:**

macOS основана на ядре Unix, что делает ее близкой к другим Unix-подобным системам, таким как Linux. Это обеспечивает программистам доступ к широкому спектру инструментов командной строки, а также возможность разработки с использованием стандартных для Unix технологий.

* macOS основана на Unix, что обеспечивает программистам доступ к Unix-подобным инструментам. В то время как Windows является преимущественно осью, ориентированной на предприятия.
* macOS поставляется с Xcode и поддерживает язык программирования Swift. Windows, с другой стороны, обычно используется для разработки на C#, C++, и Java, с использованием инструментов, таких как Visual Studio.

**2. Xcode и Swift:**

* Xcode — это интегрированная среда разработки (IDE) от Apple, предназначенная для создания приложений под macOS, iOS, watchOS и tvOS. Она включает в себя множество инструментов для разработки, отладки и профилирования приложений. Swift, современный язык программирования, был разработан Apple и является основным языком для создания приложений под macOS.

Visual Studio - популярная IDE, широко используемая для разработки под Windows. Однако, существуют и другие IDE для обеих платформ.

**3. App Store:**

* macOS имеет свой собственный App Store, который упрощает установку и обновление приложений. Это может быть удобно для конечных пользователей, но для программистов также существует возможность распространения приложений вне App Store.
* Homebrew — это популярная система управления пакетами для macOS, позволяющая устанавливать, обновлять и управлять программными пакетами из командной строки.
* Windows предоставляет Microsoft Store, а также установку программ из сторонних источников. Обновление системы и программ происходит с помощью Windows Update.

**4. Ограниченная совместимость с аппаратным обеспечением:**

* macOS ограничена аппаратным обеспечением, производимым Apple. Это означает, что разработчики могут ориентироваться на конкретные конфигурации Mac, что облегчает тестирование и оптимизацию приложений.
* Windows существует на широком спектре устройств от различных производителей, что означает более широкую совместимость с разными конфигурациями.

**5. Docker и Виртуализация:**

* macOS поддерживает использование Docker и виртуализации с помощью инструментов, таких как Parallels или VirtualBox, что облегчает тестирование и развертывание приложений в различных средах.

**Плюсы и Минусы**

* Главной причиной небольшого процента людей, которые выбрали для работы компьютеры от компании Apple, является в первую очередь цена, а во вторую очередь закрытость MacOS. Политика руководства компании направлена на то, чтоб операционная система MacOS могла устанавливаться только на компьютеры их собственного производства, поэтому те, кто решил насладиться всеми преимуществами Mac OS, просто обязаны купить себе Macintosh.
* Модельный ряд компьютеров Macintosh предложенный в магазина довольно бедный, каждое направление представлено лишь двумя тремя экземплярами. При том все компьютеры выпускаются только в готом виде, и тем, кто привык сам собирать себе компьютер этот вариант совсем не подойдет. Но с другой стороны придя в магазин, вам не придется долго думать какой из Macintosh стоит выбрать, при этом качество каждого из них будет на самом высоком уровне
* При этом ОС Mac OS была создана именно для компьютеров Macintosh, что позволяет вам использовать возможности железа на 100 процентов, а не переплачивать деньги за новинки.
* К тому же в комплект с Mac OS входит набор действительно полезных программ, позволяющих организовать весь рабочий процесс современного человека. Также плюсом является бесплатное сервисное обслуживание в любом фирменном магазине компании Apple.
* Еще одной неприятной проблемой является закрытость ОС Mac OS, что в первую очередь сказывается на недостатке программного обеспечения для нее от сторонних разработчиков. До сих пор еще не существует некоторых важных программных продуктов написанных под Macintosh, да и игроманам разгуляться не получится, поскольку игры разрабатываются в первую очередь для Windows, а потом уже для Mac OS, к тому некоторых игрушек вы вообще не найдете.
* Но время не стоит на месте, и появляются организации, которые занимаются разработкой программных продуктов под Mac OS, а известные разработчики программного обеспечения заинтересованы в том, чтоб их продукт работал на компьютерах Macintosh.
* Споры, что лучше можно продолжать до бесконечности, но если вы спросите у тех, кто решился и приобрел себе компьютер Macintosh, согласен ли он его поменять на другой, скорей всего вы получите отрицательный ответ. Те, кто работает на Macintosh – любят свои компьютеры. Объяснить это можно тем, что руководство компании Apple создает свои продукты в первую очередь для людей. Главной их стратегией является красота и удобство. К тому же все их разработки идут в ногу со временем, и даже немного его опережают. Покупая компьютер Macintosh с ОС Mac OS можно быть уверенным, что он не устареет через полгода, а будет актуален еще долгое время.

### 4.3.19. ОС Android

Абзац 4

876

Абзпац 5

### Список использованных источников

3. [TA] Таненбаум, Э. Современные операционные системы. / Э. Таненбаум, Х. Бос. — 4-е изд. — СПб.: Питер, 2015. — 1120 с.