

Оглавление

Введение.....	5
I. Понятие и структура операционных систем	7
I.1. История появления и развития операционных систем, виды операционных систем	7
I.1.1. Классификация ОС	10
I.1.2. Виды операционных систем	11
I.1.3. Структура операционных систем.....	12
I.1.4. Подходы к структурному построению ОС.....	13
I.2. Управление ресурсами вычислительных систем.	14
I.2.1. Процессы, потоки, нити	14
I.2.2. Управление памятью	19
I.2.3. Поддержка устройств ввода/вывода	21
I.2.4. Файловые системы.....	24
I.2.5. FAT	27
I.2.6. NTFS	29
II. Операционные системы семейства Microsoft Windows.....	33
II.1. DOS	33
II.2. Windows 3x/9x (DOS-Windows)	34
II.3. Технология NT	35
III. UNIX-подобные операционные системы	38
III.1. Зарождение nix`ов	39
III.2. Многообразие nix`ов.....	43
III.2.1. BSD	43
III.2.2. SunOS.....	45
III.2.3. Solaris	45
III.2.4. MacOS X.....	45
III.2.5. iOS.....	46
III.2.6. HP-UX.....	47
III.2.7. Minix	47

III.2.8. GNU/Linux.....	48
III.2.9. Android.....	49
III.2.10. Файловые системы unix-подобных ОС.....	50
III.3. Linux-системы.....	54
III.3.1. Debian	54
III.3.2. Ubuntu	55
Заключение	57
Рекомендуемая литература.....	58

Введение

Данное методическое пособие разработано преподавателем Волгоградского филиала МГГЭУ А.Б. Вахраневым для студентов 1 курса на базе среднего (полного) общего образования и 2 курса на базе основного общего образования специальности «Прикладная информатика (по отраслям)» по дисциплине «Операционные системы и среды».

В стандарте 09.02.05 по дисциплине ОП.07 «Операционные системы и среды» выделены следующие дидактические единицы:

уметь:

- использовать средства операционных систем и сред для обеспечения работы вычислительной техники;

- работать в конкретной операционной системе;

- работать со стандартными программами операционной системы;

- устанавливать и сопровождать операционные системы;

- поддерживать приложения различных операционных систем;

знать:

- состав и принципы работы операционных систем и сред;

- понятие, основные функции, типы операционных систем;

- машинно-зависимые свойства операционных систем: обработку прерываний, планирование процессов, обслуживание ввода-вывода, управление виртуальной памятью;

- машинно-независимые свойства операционных систем: работу с файлами, планирование заданий, распределение ресурсов;

- принципы построения операционных систем;

- способы организации поддержки устройств, драйверы оборудования, понятие, функции и способы использования программного интерфейса операционной системы, виды пользовательского интерфейса

Так же по стандарту дисциплина должна развивать профессиональные компетенции:

– ПК 1.4. Настраивать и работать с отраслевым оборудованием обработки информационного контента;

– ПК 1.5. Контролировать работу компьютерных, периферийных устройств и телекоммуникационных систем, обеспечивать их правильную эксплуатацию;

– ПК 4.1. Обеспечивать содержание проектных операций;

– ПК 4.4. Определять ресурсы проектных операций.

Пособие состоит (планируется) из трех частей:

1. Теоретические аспекты операционных систем (данный документ)
2. Указания к выполнению практических работ (в разработке)
3. Задания к практическим и контрольным работам (v 1.1 от 2015)

В данном пособии собраны материалы для подготовки к экзамену в теоретической части. Объем пособия, по сравнению с учебниками достаточно мал и содержит основные позиции по теории операционных систем.

I. Понятие и структура операционных систем

Как сказано в свободной Интернет-энциклопедии «Википедия», Операцио́нная систе́ма, сокр. ОС (англ. operating system, OS) – комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между устройствами вычислительной системы и прикладными программами, а с другой стороны – предназначены для управления устройствами, управления вычислительными процессами, эффективного распределения вычислительных ресурсов между вычислительными процессами и организации надёжных вычислений. Это определение применимо к большинству современных операционных систем общего назначения.

Для успешного изучения дисциплины, с целью становления высококвалифицированным специалистом в области операционных систем, обучающемуся необходимо серьезное понимание таких смежных дисциплин, как «Основы теории информации» и «Архитектура электронно-вычислительных машин и вычислительные системы». Работодателям Волгограда и области нужны специалисты, способные собрать рабочие станции, установить и настроить операционные системы, развернуть выделенные серверы. Речь идет о том, что простых умений использования Интернет-приложений, установки драйвера на новую видеокарту и инсталляции компьютерных игр явно не достаточно для того, чтобы считаться компетентным в данной сфере.

I.1. История появления и развития операционных систем, виды операционных систем

Как бы сегодня это ни казалось странным, но было время, когда компьютеры работали без операционных систем. Подробнее рекомендуется изучить поколения ЭВМ по дисциплине «Архитектура электронно-вычислительных машин и вычислительные системы». Предшественником операционных систем следует считать служебные программы (загрузчики и

мониторы), а также библиотеки часто используемых подпрограмм, начавшие разрабатываться с появлением универсальных компьютеров 1-го поколения (конец 1940-х годов). Служебные программы минимизировали физические манипуляции оператора с оборудованием, а библиотеки позволяли избежать многократного программирования одних и тех же действий (осуществления операций ввода-вывода, вычисления математических функций и т. п.). В 1950–1960-х годах сформировались и были реализованы основные идеи, определяющие функциональность ОС: пакетный режим, разделение времени и многозадачность, разделение полномочий, реальный масштаб времени, файловые структуры и файловые системы.

Следующий важный период развития операционных систем относится к 1965-1975 годам. В этот период были реализованы практически все основные механизмы, присущие современным ОС: мультипрограммирование, мультипроцессирование, поддержка многопользовательского режима, виртуальная память, файловые системы, разграничение доступа и сетевая работа. В эти годы начинается расцвет системного программирования. Из направления прикладной математики, представляющего интерес для узкого круга специалистов, системное программирование превращается в отрасль индустрии, оказывающую непосредственное влияние на практическую деятельность миллионов людей. Революционным событием данного этапа явилась промышленная реализация мультипрограммирования. В условиях резко возросших возможностей компьютера по обработке и хранению данных выполнение только одной программы в каждый момент времени оказалось крайне неэффективным. Решением стало мультипрограммирование – способ организации вычислительного процесса, при котором в памяти компьютера находилось одновременно несколько программ, попеременно выполняющихся на одном процессоре. Эти усовершенствования значительно улучшили эффективность вычислительной системы: компьютер теперь мог использоваться почти постоянно, а не менее половины времени работы компьютера, как это было раньше.

Вообще, с 1957 года в Bell Labs была начата работа по созданию операционной системы для собственных нужд. Под руководством Виктора Высотского (русского по происхождению) создавалась система BESYS. Впоследствии он возглавил проект Multics, а затем стал главой информационного подразделения Bell Labs.

В 1964 году появились компьютеры, для которых возможности BESYS уже не подходили. Высотский и его коллеги приняли решение не разрабатывать новую собственную операционную систему, а подключиться к совместному проекту General Electric и Массачусетского технологического института Multics. Телекоммуникационный гигант AT&T, в состав которого входил и Bell Labs, оказал проекту существенную поддержку, но в 1969 году вышел из проекта, поскольку он не приносил никаких финансовых выгод.

ОС разрабатывалась в конце 1960-х годов сотрудниками Bell Labs, в первую очередь Кеном Томпсоном, Деннисом Ритчи и Дугласом Макилроем.

В 1969 году Кен Томпсон, стремясь реализовать идеи, которые были положены в основу MULTICS, но на более скромном аппаратном обеспечении (DEC PDP-7), написал первую версию новой операционной системы, а другой человек – Брайан Керниган, придумал для неё название – UNICS (UNIplicated Information and Computing System, рус. Прimitивная информация и вычислительная служба) – в противовес MULTICS (MULTIplicated Information and Computing Service). Позже это название сократилось до UNIX. Так же можно отметить, что в современном мире существует великое множество именно UNIX-подобных ОС.

Про появление первых продуктов Microsoft и Apple можно порекомендовать посмотреть художественный фильм «Пираты силиконовой долины».

С понятием «Операционная среда» несколько сложнее. Если термин «Операционная система» определен во многих литературных источниках, то термин «Операционная среда» является несколько спорным. В разных источниках могут давать определения, противоречащие друг-другу. Некоторые

источники намекают, что оба понятия идентичны, это отразилось, к примеру, в «Ответах на mail.ru». Есть такие определения:

Операционная среда — это совокупность инструментов, методов их интеграции и приёмов работы с ними, позволяющая пользователю решать любые задачи в инструментальной области и большинство задач в прикладных областях. (ALT Linux 2.3 Junior. Справочник пользователя. Александр Боковой, Олег Власенко, Алексей Дьяченко и другие – 5 глава).

Операционная среда – это набор соответствующих интерфейсов, необходимых программа и пользователям для обращения к ОС с целью получения определенных сервисов. (msbro.ru Сетевые заметки системного администратора. Понятие операционной среды и операционной системы)

I.1.1. Классификация ОС

Существует множество подходов к вопросу классификации операционных систем с различных точек зрения. С точки зрения поддержки разрядности платформ их можно классифицировать как 32х битные, 64х битные и т.п. Имеется в виду, что компьютер будет работать со словами длиной 32, 64 бита соответственно. С точки зрения поддержки многопроцессорности можно выделить однопроцессорные и многопроцессорные. С точки зрения работы с пользователями – однопользовательские, многопользовательские; с графическими пользовательскими интерфейсами (GUI Graphics Users Interfaces), без них (управляемые через консоли/терминалы). С точки зрения поддержки сетевого взаимодействия – сетевые, автономные. С точки зрения лицензионных политик – закрытые, свободные; и т.п.

Но т.к. все современные операционные системы многопользовательские, многопроцессорные, сетевые (т.е. с реализациями всех современных достижений) гораздо легче все операционные системы разделить на два условных класса:

1. Операционные системы корпорации Microsoft (имеются в виду линейки ОС Windows);

2. Unix-подобные ОС (коих превеликое множество).

Т.е. какой бы ни была ОС, любая современная ОС должна выполнять ряд функций, среди которых можно выделить:

1. Управление процессами
2. Управление памятью
3. Управление пользователями
4. Управление файлами
5. Поддержка устройств ввода/вывода
6. Защитные функции
7. Поддержка универсальности в использовании компьютера
8. Управление сетевым взаимодействием

I.1.2. Виды операционных систем

Операционные системы корпорации Microsoft – Windows – подразделяются на две основные линейки продуктов: 1. Windows 3x/9x – основанные на ядре MS DOS (Windows 1, Windows 2, Windows 3, Windows 95, Windows 98, Windows Millennium) – это «мертвая» линейка продуктов, т.к. Microsoft полностью отказались от ее поддержки; 2. Windows NT – основная на данный момент линейка продуктов, основанных на ядре NT (Windows NT, Windows NT 2, Windows NT 3, Windows NT 4, Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8).

Т.е. существуют различные версии ОС, различные выпуски (релизы). По функциональному назначению принято выделять следующие виды ОС:

1. Desktop (настольные, пользовательские);
2. Server (серверные, обрабатывающие сетевые запросы).

Если описать линейки Windows в принципе достаточно просто, то сделать то же самое с Unix-подобными ОС значительно сложнее. Попытка описания будет сделана во второй теме второго раздела, а пока достаточно будет выделить направления BSD, Linux, и «другие». Ставшая популярной ОС Android на смартфонах принадлежит линии Linux, а на Apple продуктах ОС линии BSD.

I.1.3. Структура операционных систем

Не смотря на то, что операционных систем существует достаточно большое количество, имеются общие подходы к структурному построению. Структура описывает основные архитектурные особенности операционных систем, обеспечивающая их функциональность.



Рис. 1. Классическая структура ОС

Вся вычислительная система состоит из трех основных наборов компонент: 1. Прикладные программы – именно они решают конкретные задачи, т.е. именно ради них и создавались компьютеры; 2. Аппаратные устройства – именно они реализуют функционирование вычислительных процессов на физическом уровне, без устройств программам выполняться негде; 3. Операционная система – всего лишь прослойка между прикладными программами и аппаратными устройствами, которая помогает программам выполняться на аппаратных устройствах.

Т.е. обычные пользователи с операционными системами не работают вовсе – они работают только с программами и вообще не понимают, что происходит на уровнях ОС и устройств.

Программы запускаются и функционируют благодаря интерфейсам системных вызовов (API – Application-Program Interfaces). Т.е. посылают системе запросы в определенном виде, а API на эти запросы отвечает в

соответствии, перенаправив их в ядро ОС. Ядро ОС управляет устройствами через драйверы (управляющие микропрограммы).

Ядро ОС – тот набор микропрограмм, который загружается в ОП при включении компьютера и обеспечивает функционирование всей вычислительной системы.

I.1.4. Подходы к структурному построению ОС

Итак, ОС является, по сути, посредником между исполняемыми на компьютере программами и его аппаратными устройствами. Все современные ОС, как правило, являются многопользовательскими, многозадачными, с поддержкой многоядерности и многопроцессорности и т.п. Но можно выделить некоторые подходы к структурному построению самого ядра ОС. Так ОС можно поделить еще на монолитные и микроядерные (модульные).

Монолитные системы представляют собой набор процедур, каждая из которых может вызывать другие по необходимости. При таком подходе каждая процедура имеет свой четко определенный интерфейс в терминах параметров и результатов. При построении таких систем все процедуры должны быть скомпилированы и связаны в одно целое. Такая организация предполагает, что есть главная программа, которая вызывает требуемые сервисы; набор сервисных процедур, реализующих системные вызовы; и набор системных утилит и прикладных программ, обслуживающих сервисные процедуры. Т.е. для каждого системного вызова имеется своя сервисная процедура.

В многоуровневым монолитным ОС имеется иерархия уровней. Уровни образуются типовыми группами функций (файловые системы, системы управления процессами, устройствами и т.д.). Прикладные программы или части самой ОС передают запросы по уровням вверх и вниз, не перескакивая (не пропуская) ни одного уровня. Внесение изменений (добавление новых функций, удаление ненужных) в такие системы – задача довольно сложная.

Подход к построению ОС на базе технологий клиент-сервер предполагает наличие микроядра (поставщика сервисов) и программных компонент (его клиентов). Взаимодействие сервера и клиента четко стандартизируется, так что

различные клиенты могут быть реализованы различными способами. Модель клиент-сервер это даже не технология – это скорее концептуальный подход к построению не только ОС, но вообще множества информационных систем. Многие приложения имеют свои серверные части, которые взаимодействуют с клиентскими частями через микроядро.

Конечно, существуют и более уточненные подходы, технологии и приемы к построению ОС, такие как объектно-ориентированный подход к программированию, OLE-объекты, множественные программные среды и т.п.

I.2. Управление ресурсами вычислительных систем.

Итак, если ОС управляет ресурсами компьютера (основные – это ресурсы ЦПУ и ОП) и обладает рядом функций (обязанностями по управлению), то стоит рассмотреть основные из них.

I.2.1. Процессы, потоки, нити

Под понятием «процесс» понимается выполнение инструкций компьютерной программы на центральном процессорном устройстве. Так же можно определить процесс как совокупность взаимосвязанных и взаимодействующих действий, преобразующих входящие данные в исходящие.

Т.О. «Процесс» – это некоторая абстракция (упрощенное понимание), описывающая выполняемую программу.

Компьютерная программа сама по себе – это только пассивная совокупность инструкций, в то время как процесс – это непосредственное выполнение этих инструкций.

Часто процессом называют выполняющуюся программу и все её элементы: адресное пространство, глобальные переменные, регистры, стек, открытые файлы и т. д.

Понимание процессов лучше начать с их появления в системе. Сперва генерируется системный вызов, который создаёт (порождает) новый процесс. Дочерние процессы представляют собой почти точную копию родительского. Далее дочерний процесс выполняет назначенную ему функцию, а исходный

процесс (родительский) – то, что написано в программе после вызова. Потом отличий может стать больше, так что функционирование процессов способно различаться достаточно сильно. Но при необходимости этому можно воспрепятствовать, задавая флаги, указывающие, что порожденный процесс будет иметь со своим предком общие:

- адресное пространство;
- информацию о файловой системе: корневой и текущий каталоги и т.п.;
- таблицу открытых файлов;
- таблицу обработчиков сигналов;
- родителя – конечно, в этом случае будет порожден не «дочерний», а «сестринский» процесс.

Так как все современные ОС являются многозадачными, процессы могут быть активными и пассивными.

– Активный процесс – в текущий момент выполняется ЦПУ и имеет доступ ко всем необходимым ресурсам.

– Пассивный процесс – в текущий момент ЦПУ не выполняется. На этом принципе и основано мультипрограммирование или псевдопараллельное выполнение программ – один одноядерный ЦПУ не может одновременно выполнять несколько программ, но он так часто переключается с одной задачи на другую, что пользователю кажется, будто множество программ выполняются одновременно.

В различных ОС процессы могут находиться в разных состояниях, но во всех современных многозадачных ОС этих состояний как минимум 3:

- готовность – пассивное состояние процесса, при котором ЦПУ, освободившись от других задач, может исполнять данный процесс;
- выполнение – активное состояние процесса, когда ЦПУ выполняет данный процесс;

– ожидание – пассивное состояние процесса, при котором процесс ожидает либо сигналов от периферийных устройств, либо от других процессов, либо других явлений.

Говоря о процессах, так же нужно ознакомиться с терминами:

– Дескриптор процесса – содержит такую информацию о процессе, которая необходима ядру ОС в течение всего жизненного цикла процесса, независимо от того, находится ли он в активном или пассивном состоянии, находится ли образ процесса в оперативной памяти или выгружен на накопитель.

– Контекст процесса – содержит менее оперативную, но более объемную часть информации о процессе, необходимую для возобновления выполнения процесса с прерванного места.

– Приоритет процесса – некоторое число, указывающее на важность выполнения процесса при выборе его ЦПУ из очереди на выполнение.

Понятия «потoki» и «нити» являются, на самом деле, одним и тем же. Как правило, одной выполняющейся программе ставится в соответствие один или несколько управляющих потоков. Поток выполнения (англ. thread – нить) – наименьшая единица обработки, исполнение которой может быть назначено ядром операционной системы. Реализация потоков выполнения и процессов в разных операционных системах отличается друг от друга, но в большинстве случаев поток выполнения находится внутри процесса. Несколько потоков выполнения могут существовать в рамках одного и того же процесса и совместно использовать ресурсы, такие как память, тогда как процессы не разделяют этих ресурсов. В частности, потоки выполнения разделяют инструкции процесса (его код) и его контекст (значения переменных, которые они имеют в любой момент времени). В качестве аналогии потоки выполнения процесса можно уподобить нескольким вместе работающим поварам. Все они готовят одно блюдо, читают одну и ту же кулинарную книгу с одним и тем же рецептом и следуют его указаниям, причём не обязательно все они читают на одной и той же странице.

На одном процессоре многопоточность обычно происходит путём временного мультиплексирования (объединения нескольких явлений в одно, как и в случае многозадачности): процессор переключается между разными потоками выполнения. Это переключение контекста обычно происходит достаточно часто, чтобы пользователь воспринимал выполнение потоков или задач как одновременное. В многопроцессорных и многоядерных системах потоки или задачи могут реально выполняться одновременно, при этом каждый процессор или ядро обрабатывает отдельный поток или задачу.

Многие современные операционные системы поддерживают как временные нарезки от планировщика процессов, так и многопроцессорные потоки выполнения. Ядро операционной системы позволяет программистам управлять потоками выполнения через интерфейс системных вызовов. Некоторые реализации ядра называют потоком ядра, другие же – лёгким процессом (англ. light-weight process, LWP), представляющим собой особый тип потока выполнения ядра, который совместно использует одни и те же состояния и данные.

Программы могут иметь пользовательское пространство потоков выполнения при создании потоков с помощью таймеров, сигналов или другими методами, позволяющими прервать выполнение и создать временную нарезку для конкретной ситуации.

Нити, т. е. параллельно выполняемые части одной программы, часто реализованы просто как процессы, порожденные с указанием флага адресного пространства, и с точки зрения ядра системы ничем не отличаются от любых других процессов. Однако в некоторых альтернативных реализациях многонитевых библиотек дело обстоит иначе.

Помимо таких процессов бывают еще внутриядерные (иногда называемые «ущербными») для внутренних системных нужд. У них нет параметров командной строки, как правило, они не имеют открытых файлов и т.д. Поскольку, несмотря на свою ущербность, эти процессы все равно фигурируют в списке задач, в литературе иногда различают полноценные

процессы, порожденные из «пространства пользователя» (userspace), и задачи, т.е. все процессы, включая внутренние процессы ядра.

Процессы, выполняющие разные программы, образуются благодаря применению специальных системных функций. Эти функции отличаются форматом вызова, но в конечном итоге делают одну и ту же вещь: замещают внутри текущего процесса исполняемый код на код, содержащийся в указанном файле. Файл может быть не только двоичным исполняемым файлом, но и скриптом командного интерпретатора, и двоичным файлом другого формата (например, классом java, исполняемым файлом DOS и т.п.). В последнем случае способ его обработки определяется настраиваемым модулем ядра.

Таким образом, операция запуска программы, которая в DOS и Windows3x/9x выполняется как единое целое, в Unix-подобных ОС и NT разделена на две: сначала производится запуск, а потом определяется, какая программа будет работать.

Рассмотрев рождение процесса, логично будет обсудить и его завершение. Когда процесс закончит работу (нормально или аварийно), он уничтожается, освобождая все использовавшиеся им ресурсы компьютера.

Если родительский процесс по какой-то причине завершится раньше дочернего, последний становится «сиротой» (orphaned process). «Сироты» автоматически «усыновляются» высшим процессом, который и принимает сигнал об их завершении.

Если же потомок уже завершил работу, а предок не готов принять от системы сигнал об этом событии, то потомок не исчезает полностью, а превращается в «зомби» (zombie). Зомби не занимает процессорного времени, но строка в таблице процессов остается, и соответствующие структуры ядра не освобождаются. После завершения родительского процесса «осиротевший» зомби на короткое время также становится потомком высшего процесса, после чего уже «окончательно умирает».

Наконец, процесс может надолго впасть в «сон», который не удастся прервать: в поле Stat это обозначается буквой D. Процесс, находящийся в таком

состоянии, не реагирует на системные запросы и может быть уничтожен только перезагрузкой системы.

По критериям выбора готовых процессов на исполнение ОС так же делятся на системы разделенного времени (когда какому-либо процессу будет выделено фиксированное количество процессорного времени) и системы реального времени (ОС, в которой успешность работы любой программы зависит не только от её логической правильности, но и от времени, за которое она получила этот результат; если система не может удовлетворить временным ограничениям, должен быть зафиксирован сбой в её работе).

I.2.2. Управление памятью

Следующая задача, с которой имеет дело ОС – управление памятью.

Под памятью (memory) здесь подразумевается оперативная память компьютера. В отличие от памяти жесткого диска, которую называют внешней памятью (storage), оперативной памяти для сохранения информации требуется постоянное электропитание.

Память является важнейшим ресурсом, требующим тщательного управления со стороны операционной системы. Особая роль памяти объясняется тем, что процессор может выполнять инструкции только в том случае, если они находятся в памяти. Память распределяется как между модулями прикладных программ, так и между модулями самой операционной системы.

В ранних ОС управление памятью сводилось просто к загрузке программы и ее данных из некоторого внешнего накопителя (перфоленты, магнитной ленты или магнитного диска) в память. С появлением мультипрограммирования перед ОС были поставлены новые задачи, связанные с распределением имеющейся памяти между несколькими одновременно выполняющимися программами.

Функциями ОС по управлению памятью являются:

- отслеживание свободной и занятой памяти;

- выделение памяти процессам и освобождение памяти по завершении процессов;

- вытеснение кодов и данных процессов из оперативной памяти на диск (полное или частичное), когда размеры основной памяти не достаточны для размещения в ней всех процессов, и возвращение их в оперативную память, когда в ней освобождается место;

- настройка адресов программы на конкретную область физической памяти.

Так все данные, все процессы и потоки, с которыми может работать компьютер, находятся в оперативной памяти, для программирования вообще и для системного программирования в частности, оперативной памяти всегда мало. Т.е. обязательно найдется случай, когда оперативной памяти не хватит. Так системное программирование пришло к следующему. Все современные операционные системы работают со swp (своп)-файлом. В этот файл можно помещать те наборы нулей и единиц (процессы, данные), которые не находятся в активных состояниях, не нужны для работы «прямо сейчас». Т.е. с помощью своп-файла можно освободить часть оперативной памяти для использования в более актуальных приложениях.

Изначально под свопингом понималась выгрузка процесса из оперативной памяти целиком, в результате чего неактивные процессы могли полностью отсутствовать в ОЗУ. При наступлении условий активизации процесса диспетчер памяти загружал образ процесса обратно.

Смысл термина изменился в 60-х годах, когда в операционных системах появилась поддержка виртуальной памяти: под свопингом стали понимать загрузку и выгрузку отдельных страниц.

Виртуальная память (англ. Virtual memory) – технология управления памятью ЭВМ, разработанная для многозадачных операционных систем. При использовании данной технологии для каждой программы используются независимые схемы адресации памяти, отображающиеся тем или иным способом на физические адреса в памяти ЭВМ. Позволяет увеличить

эффективность использования памяти несколькими одновременно работающими программами, организовав множество независимых адресных пространств и обеспечить защиту памяти между различными приложениями. Также позволяет программисту использовать больше памяти, чем установлено в компьютере, за счет отправки неиспользуемых страниц на вторичное хранилище.

При использовании виртуальной памяти упрощается программирование, так как программисту больше не нужно учитывать ограниченность памяти, или согласовывать использование памяти с другими приложениями. Для программы выглядит доступным и непрерывным все допустимое адресное пространство, вне зависимости от наличия в ЭВМ соответствующего объема ОЗУ.

Применение механизма виртуальной памяти позволяет:

- упростить адресацию памяти клиентским программным обеспечением;
- рационально управлять оперативной памятью компьютера (хранить в ней только активно используемые области памяти);
- изолировать процессы друг от друга (процесс полагает, что монопольно владеет всей памятью).

В настоящее время эта технология имеет аппаратную поддержку на всех современных бытовых процессорах. В то же время во встраиваемых системах и в системах специального назначения, где требуется либо очень быстрая работа, либо есть ограничения на длительность отклика (системы реального времени), виртуальная память используется относительно редко. Также в таких системах реже встречается многозадачность и сложные иерархии памяти.

Таким образом, для простоты понимания можно сказать, что виртуальная память – это суммарная память ОЗУ+накопитель.

Реализуется виртуальная память, как правило, при помощи файла подкачки (swap). Хороший администратор обязательно держит под контролем его объем.

I.2.3. Поддержка устройств ввода/вывода

ОС управляет устройствами через драйверы – микропрограммы управления. Устройства компьютера состоят не только из механизмов, ими управляют микроконтроллеры – электронные блоки управления. Драйвер устройства служит некоторым интерфейсом взаимодействия такого управляющего электронного блока и ядра ОС. Можно представить такую работу (очень примерно) в виде цепи передачи: устройство – контроллер – драйвер – ядро ОС. Если какой-либо программе вдруг понадобятся ресурсы какого-то устройства, она будет обращаться не к нему напрямую, а через системные программные интерфейсы к ядру ОС, а уж ядро ОС проделает всю остальную работу с помощью драйверов.

Используя термин «драйвер» (водитель), нужно помнить, что это – микропрограмма или набор микропрограмм, служащих для связи различных частей вычислительной системы. Т.е. имеются аппаратные драйверы, программные драйверы.

Аппаратный драйвер – компьютерное программное обеспечение, с помощью которого другое программное обеспечение (операционная система) получает доступ к аппаратному обеспечению некоторого устройства. Обычно с операционными системами поставляются драйверы для ключевых компонентов аппаратного обеспечения, без которых система не сможет работать. Однако для некоторых устройств (таких, как видеокарта или принтер) могут потребоваться специальные драйверы, обычно предоставляемые производителем устройства.

В общем случае драйвер не обязан взаимодействовать с аппаратными устройствами, он может их только имитировать (например, драйвер принтера, который записывает вывод из программ в файл), предоставлять программные сервисы, не связанные с управлением устройствами (например, `/dev/zero` в Unix, который только выдаёт нулевые байты), либо не делать ничего (например, `/dev/null` в Unix и `NUL` в DOS/Windows).

Теперь нужно разобраться с основами системы прерываний.

Как сказано в той-же Википедии, «Прерывание» (англ. interrupt) – сигнал, сообщающий процессору о наступлении какого-либо события. При этом выполнение текущей последовательности команд приостанавливается, и управление передаётся обработчику прерывания, который реагирует на событие и обслуживает его, после чего возвращает управление в прерванный код.

В зависимости от источника возникновения сигнала, прерывания делятся на:

- асинхронные, или внешние (аппаратные) – события, которые исходят от внешних ЦПУ источников (например, периферийных устройств) и могут произойти в любой произвольный момент: сигнал от таймера, сетевой карты или дискового накопителя, нажатие клавиш клавиатуры, движение мыши. Факт возникновения в системе такого прерывания трактуется как запрос на прерывание (англ. Interrupt request, IRQ);

- синхронные, или внутренние – события в самом процессоре как результат нарушения каких-то условий при исполнении машинного кода: деление на ноль или переполнение стека, обращение к недопустимым адресам памяти или недопустимый код операции;

- программные (частный случай внутреннего прерывания) – инициируются исполнением специальной инструкции в коде программы. Программные прерывания, как правило, используются для обращения к функциям встроенного программного обеспечения (firmware), драйверов и операционной системы.

Термин «ловушка» (англ. trap) иногда используется как синоним термина «прерывание» или «внутреннее прерывание». Как правило, словоупотребление устанавливается в документации производителя конкретной архитектуры процессора.

К примеру, пользователь нажал кнопку «Й» на клавиатуре, при открыто программе «Word». Блок управления клавиатуры передал сигнал (между прочим 2 байта) на аппаратный интерфейс (PS/2 или USB, в зависимости от

порта), сформировалось асинхронное прерывание и обработчик прерываний с помощью драйвера клавиатуры, используя программное прерывание «отчитался» перед ядром ОС о произошедшем событии. Ядро ОС через системный программный интерфейс передало соответствующие данные активной программе, и на экране отразилась буква «й». Пример слишком грубый и частично не точный, но для понимания основ, может быть сгодится.

Главное тут для начала понять: для того, чтобы ВС работала, как единое целое, среди прочего нужна система прерываний. ОС активно использует программные прерывания для управления периферийными устройствами, в том числе, устройствами ввода/вывода.

I.2.4. Файловые системы

Для того, чтобы понять принципы функционирования файловых систем, нужно понимать, что такое «файл». И как у многих терминов в информационных технологиях, у термина «файл» существует множество определений, часто противоречащих друг другу.

В Википедии сказано, что «Файл» (англ. file) — именованная область данных на носителе информации. В принципе, это — одно из лучших определений.

Файл — это информация, хранящаяся в долговременной памяти как единое целое и обозначенная именем. Так сказано в учебнике по информатике за 6 класс, автор — Босова Л.Л. Не плохое определение, если не придирается к понятию «информация».

У одного из наиболее уважаемых специалистов в области операционных систем, профессора Эндрю Стюарта Таненбаума, в книге «Современные операционные системы» сказано, что файлы являются логическими информационными блоками, создаваемыми процессами, и, в принципе, близко к истине рассматривать каждый файл как некую разновидность адресного пространства.

Совсем плохие определения содержат термины «жесткий диск», «дискета» (т.е. идет привязка к конкретному типу накопителя) или другие термины, которые не вносят определенность, а лишь добавляют путаницы.

Но, так как «файл» в разных операционных системах обладает различным набором атрибутов, свойств и методов доступа, универсального определения, которое бы учитывало все особенности, сформулировано не было.

Главное – это понимать, что во всех современных ОС есть как-бы только два типа файла (на самом общем уровне):

- текстовые файлы – обычные файлы, контентом которых являются символы, преобразуемые прикладным ПО – это не только «вордовские» документы – и музыкальные, и графические, и видеофайлы – всё это текстовые файлы;

- исполняемые файлы – двоичные, объектные, которые исполняются, так или иначе, ядром ОС.

Папка, или каталог – всего лишь текстовый файл, в который можно записать внутренние ссылки.

Далее можно попытаться разобраться, что такое файловая система. И у этого термина множество определений. На той же Википедии сказано, что «Фáйловая систéма» (англ. file system) – порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах, а также в другом электронном оборудовании: цифровых фотоаппаратах, мобильных телефонах и т. п. Файловая система определяет формат содержимого и способ физического хранения информации, которую принято группировать в виде файлов. Конкретная файловая система определяет размер имен файлов и (каталогов), максимальный возможный размер файла и раздела, набор атрибутов файла. Некоторые файловые системы предоставляют сервисные возможности, например, разграничение доступа или шифрование файлов.

Основные функции любой файловой системы нацелены на решение следующих задач:

- именование файлов;
- программный интерфейс работы с файлами для приложений;
- отображения логической модели файловой системы на физическую организацию хранилища данных;
- организация устойчивости файловой системы к сбоям питания, ошибкам аппаратных и программных средств;
- содержание параметров файла, необходимых для правильного его взаимодействия с другими объектами системы (ядро, приложения и пр.).

В многопользовательских системах появляется ещё одна задача: защита файлов одного пользователя от несанкционированного доступа другого пользователя, а также обеспечение совместной работы с файлами, к примеру, при открытии файла одним из пользователей, для других этот же файл временно будет доступен в режиме «только чтение».

В качестве примера одной из первых файловых систем (еще не для дисков, а для магнитных лент) можно привести систему TAR (Tape Archive – архив на магнитной ленте). Современные технологии унаследовали от этой системы архивные файлы *.tar.

В этой системе все файлы просто записывались один за другим. Файл начинался и заканчивался служебными данными, содержащими имя файла и длину поля данных. Всё.

С появлением других накопителей, как диски (гибкие, жесткие, flash и т.д.) ФС тоже стали более сложными. Для начала нужно понять, что каждый физический накопитель (будь то диск или flash-накопитель) может быть «разбит» на несколько «логических дисков». Т.е. логический диск – виртуальный накопитель.

Еще одно понятие, которое следует рассмотреть до конкретных примеров современных ФС – это MBR.

Главная загрузочная запись (англ. master boot record, MBR) – код и данные, необходимые для последующей загрузки операционной системы и

расположенные в первых физических секторах (чаще всего в самом первом) на жёстком диске или другом устройстве хранения информации.

MBR содержит небольшой фрагмент исполняемого кода, таблицу разделов (partition table) и специальную сигнатуру.

Функция MBR – «переход» в тот раздел жёсткого диска, с которого следует исполнять «дальнейший код» (обычно – загружать ОС). На «стадии MBR» происходит выбор раздела диска, загрузка кода ОС происходит на более поздних этапах алгоритма.

В процессе запуска компьютера, после окончания начального теста (Power-on self-test – POST), Базовая система ввода-вывода (BIOS) загружает «код MBR» в оперативную память (в IBM PC обычно с адреса 0000:7c00) и передаёт управление находящемуся в MBR загрузочному коду.

Кла́стер (англ. cluster – скопление, кисть, рой) – объединение нескольких однородных элементов, которое может рассматриваться как самостоятельная единица, обладающая определёнными свойствами.

В некоторых типах файловых систем кластер является логической единицей хранения данных в таблице размещения файлов, объединяющей группу секторов. Например, на дисках с размером секторов в 256 байт, 256-байтный кластер содержит один сектор, тогда как 2-килобайтный кластер содержит восемь секторов.

Как правило, это наименьшее место на диске, которое может быть выделено для хранения файла.

Понятие кластер используется в файловых системах FAT, NTFS, а так же HFS Plus. Другие файловые системы оперируют схожими понятиями (зоны в Minix, блоки в Unix).

Файловые системы UNIX-подобных ОС рассмотрим позже, а тут приведем примеры самых распространённых ФС для Windows.

1.2.5. FAT

FAT (англ. File Allocation Table – «таблица размещения файлов») – классическая архитектура файловой системы, которая из-за своей простоты всё

ещё широко используется для флеш-накопителей. Используется в дискетах и некоторых других носителях информации. Ранее использовалась и на жестких дисках.

Существует четыре версии FAT – FAT8, FAT12, FAT16 и FAT32. Они отличаются разрядностью записей в дисковой структуре, то есть количеством бит, отведённых для хранения номера кластера. FAT12 применяется (применялся) в основном для дискет, FAT16 – для дисков малого объёма (до 2Гб). На основе FAT была разработана новая файловая система exFAT (extended FAT), используемая преимущественно для флеш-накопителей.

Изначально FAT не поддерживала иерархическую систему каталогов. Все файлы располагались в корневом каталоге. Это оказалось неудобно и к тому же малый размер корневого каталога ограничивал количество файлов на диске. Каталоги были введены с выходом MS-DOS 2.0.

Таким образом, схематично FAT можно представить в виде:



Рис. 2. Структура FAT

MBR – этот сектор несет информацию о разделах, существующих на диске, с каких цилиндров, головок и секторов они начинаются (partition table). Есть также указание для BIOS, где надо искать основные файлы операционной системы.

Partition Table – таблица разделов. Этот сектор содержит информацию о дорожках, секторах, головках HDD, и, конечно же, содержит информацию о типе самой ФС.

Volume Boot Sector – загрузочный сектор тома. Этот сектор отвечает за данные о размере кластера, тип и размер используемой таблицы расположения файлов.

File Allocation Table – таблица расположения (размещения) файлов. Этот раздел содержит данные о расположении файлов на накопителе. На HDD она содержится в двух экземплярах.

Root Directory – корневой каталог. Этот раздел несет информацию о типе файлов, их названиях, размерах, дате создания. Кроме этого, для каждого файла в корневом каталоге есть номер кластера, с которого начинается файл.

Стоит заметить, что если HDD разбит на разделы, то FAT и RD будут записаны в начале каждого раздела. То есть, для каждого раздела HDD используются свои таблицы и корневые каталоги.

Таким образом, форматирование накопителя в файловую систему FAT может выглядеть следующим образом: Для начала накопитель делится (или не делится) на разделы, а потом создаются FAT и RD секторы. При записи файла на накопитель, данные о нем записываются в Root Directory. В поле «указатель» записывается номер начального кластера. По этому номеру, система обращается в ячейку таблицы, с таким же номером, где будет записан номер следующего кластера. И так далее, до тех пор, пока не будет описано расположение всего файла на диске. Запись закончится командой «стоп», то есть на этом кластере файл заканчивается.

Считывание происходит так же. Сначала считываются данные о файле, затем по указателю система переходит в таблицу и там, считывает остальные номера кластеров, которые занимает файл.

I.2.6. NTFS

Эту файловую систему используют операционные системы корпорации Microsoft линейки NT – Windows NT/2000/XP/Vista/7/8/10 и т.д.

При установке NTFS (NT File System), диск разделяется на две неравные части: первая отводится под MFT (Master File Table – общая таблица файлов), называется MFT (так же) – зоной и занимает порядка 12% от общего размера диска, вторую часть занимают собственно записанные данные.

Есть еще и третья зона, но о ней ниже. Главное понятие – MFT – это основа NTFS. Он лежит, как было сказано, в MFT-зоне т. е. в начале диска.

Каждая запись в MFT соответствует какому-либо файлу и занимает около 1 Кб. По своей сути это каталог всех файлов, находящихся на логическом диске. Надо заметить, что любой элемент данных в NTFS рассматривается как файл, даже MFT. Первые 16 файлов (метафайлы) в MFT-зоне являются особым набором данных. В них содержатся служебные данные, они имеют фиксированное положение и они недоступны даже операционной системе. Первым из этих 16 файлов является сам MFT-файл. Существует копия первых трех записей. Выше говорилось о третьей зоне, так вот там она и лежит, и своим положением, если можно так выразиться, делит диск на две части.

Это сделано для надежности, в случае утери данных в MFT-файле, всегда можно попытаться восстановить данные. Все остальные файлы в MFT-зоне могут располагаться произвольно. Надо заметить, что в MFT-зоне теоретически, кроме служебных файлов, ничего не находится. Но бывают случаи, когда места на той части диска, что отведена для пользователя не остается, и тогда MFT-зона уменьшается.

Соответственно, появляется место во второй половине диска для записи данных. Когда же в этой зоне освобождается достаточное количество свободного места, MFT-зона опять расширяется. И вот тут появляется проблема. В MFT-зону попадают обычные файлы и она начинает фрагментироваться. Это не критично, но и приятного тут мало.

Каждый из остальных метафайлов отвечает за какую-либо область работы. Начинаются они с символа имени \$ (тем, кто занимается программированием значок известен). Вот некоторые из них:

\$MFT – не что иное, как сам MFT

\$MFTmirr – та самая копия, что располагается «посередине» накопителя

\$LogFile – это файл журналирования (log – хронологические записи происшествий/действий)

\$Boot – загрузочный сектор

\$Bitmap – битовая карта свободного места раздела

Ну и так далее. Данные о метафайлах находятся в MFT-файле.

Всё это разработано для увеличения надежности NTFS и себя оправдывает. NTFS практически не имеет ограничения на размеры диска (во всяком случае, при нынешних технологиях производства накопителей). Размер кластера может варьироваться от 512 b до 64 Kb, хотя обычный его размер равен 4 Kb.

Далее о каталоге. Это метафайл с обозначением \$.. Он разделен на части, в каждой из которых содержится имя файла, его атрибуты и ссылка на MFT-файл. А там уже есть все остальные данные. Каталог представляет собой бинарное дерево. В каталоге «информация» о данных на диске расположена таким образом, что при поиске какого-либо файла каталог разбивался на две части и ответ заключался в том, в какой именно части находится искомое. Затем та же самая операция повторяется в выбранной половине. И так до тех пор, пока не будет найден нужный файл.

А теперь о файлах. Их как таковых нет. Есть так называемые стримы, или говоря нормальным русским языком – потоки. То есть, любая единица информации представляет собой несколько потоков. Один поток – это сами данные, он является основным. Другие потоки – атрибуты файла. К любому файлу можно прикрепить любой другой файл. Проще говоря, к потокам одних данных можно прикрепить совершенно новый поток и записать туда новые данные. Вот только данные по объему файла берутся по объему основного потока.

Пустые или малоразмерные файлы на диске отображены только в метафайлах. Сделано это в целях экономии дискового пространства. Вообще надо отметить, что понятие «файл» намного глубже и шире и все свойства описать довольно сложно. Стоит отметить, что максимальная длина имени файла может достигать 255 символов.

Ко всему прочему, файлы NTFS имеют такой замечательный атрибут, как «сжатый». Любой файл или даже каталог может быть сжат. Сама операция сжатия происходит незаметно, так как скорость ее довольно высока. Плюс,

используется так называемое «виртуальное сжатие» т. е. одна часть файла может быть сжата, а другая нет.

Сжатие осуществляется блоками. Каждый блок равен 16 кластерам.

В NTFS используется шифрование данных. Таким образом, если ОС была удалена, а затем установлена по новой, зашифрованные файлы без соответствующей санкции прочитать нельзя.

Теперь о журналировании. Тут для начала придется рассмотреть понятие «транзакция». Транзакция – это набор действий, которые должны быть выполнены целиком и полностью (корректно), как единое, неделимое действие. В противном случае, оно вообще не будет выполнено. Так вот, на основе этого подхода, при сбое во время записи данных на диск, пометок о новом файле в метафайлах сделано не будет.

А место, куда была начата запись будет считать чистым. Это необходимо для предохранения от различного рода проблем. В общем, выполнилось действие до конца – осуществилась запись. Не удалось – и записывать об этом незачем. Но следует заметить, что функция журналирования сохраняет работоспособность файловой системы, а не пользовательских данных.

И, наконец, в NTFS есть еще две такие функции, как Symbolic Links – возможность создания виртуальных каталогов, и Hard Links – поддержка нескольких имен для одного и того же файла.

II. Операционные системы семейства Microsoft Windows

Корпорация Microsoft занимает серьезные (если не сказать «подавляющие») позиции на рынке операционных систем. Для формирования первоначального представления об образовании и развитии такой ситуации можно порекомендовать посмотреть художественный фильм «Пираты силиконовой долины». Хотя фильм и художественный, в нем довольно доходчиво показаны явления возникновения таких корпораций, как Apple, Microsoft. Так же повествуется, как акулы бизнеса, беспринципные и жесткие, идут на всё ради развития бизнеса и прибыли, в том числе на «кражу» идей и программного кода.

Ограничиваться просмотром фильмов, тем не менее не стоит – нужно читать дополнительную литературу, научные статьи, и изучать соответствующие вопросы самостоятельно для того, чтобы сформировать действительно свою точку зрения.

II.1. DOS

DOS (англ. Disk Operating System – дисковая операционная система, ДОС) – семейство операционных систем для персональных компьютеров, ориентированных на использование дисковых накопителей, таких как «жёсткий диск» и «дискета».

Существовало достаточно большое количество отдельных реализаций DOS. Одну из них приобрела компания Microsoft в 1981 году у Seattle Computer Products, предварительно заключив контракт с самой крупной в то время IT-компанией IBM на лицензионное сопровождение и оснащение компьютеров IBM системой MS-DOS.

Благодаря тому, что IBM была в то время самой крупной фирмой-производителем компьютеров, MS-DOS распространился вместе с этими компьютерами, практически, повсеместно. Таким образом, операционная система MS-DOS не являлась, строго говоря, разработкой фирмы Microsoft; популярность этой системе обеспечили компьютеры IBM; DOS была

однозадачной, 16-разрядной, автономной системой без графического пользовательского интерфейса, без подсистемы управления пользователями и т.п.

II.2. Windows 3x/9x (DOS-Windows)

Windows 3.x – общее название поколения операционных систем Windows от компании Microsoft, выпущенных с 1990 по 1994 год. Первой широко распространённой версией Microsoft Windows стала 3.0, позволив Майкрософт соревноваться с Macintosh и Commodore Amiga в области операционных систем с графическим интерфейсом.

На самом деле Windows 3.x не является полноценной самостоятельной операционной системой, потому что она фактически является оболочкой-надстройкой над DOS, так как использует её в качестве базовой рабочей платформы.

Windows 9x – часто используемое общее название для операционных систем Windows версий 4.x: Windows 95, Windows 98/98SE и Windows ME от корпорации Microsoft. Поскольку архитектура этих систем весьма схожа, термин Windows 9x зачастую используется для обозначения их всех (например, при сравнении этих систем с системами линии Windows NT).

DOS-Windows – общее обозначение всех ОС корпорации Microsoft, основанных на ядре MS-DOS.

Всю линейку продуктов DOS-Windows можно представить следующим образом:

1. Microsoft Windows 1 – графический интерфейс пользователя компании Microsoft для операционной системы MS-DOS, использующий принцип фреймового менеджера окон, созданный для облегчения диалога с последней, унификации внешнего вида приложений и оптимизации работы с периферийными устройствами (например, с принтером). 1983 г.

2. Windows 2 – так же графическая оболочка для MS-DOS, под более новые компьютеры. 1987 г.

3. Windows 3 – обновленный пользовательский интерфейс для MS-DOS, для более новых компьютеров. 1990 г.

4. Windows 95 – это, по сути, Windows 4, но с более усовершенствованным графическим интерфейсом пользователя (в т.ч. появилась всеми любимая кнопка «Пуск»), к тому же поддерживающая использование 32х разрядных кодов и многозадачная (спорный момент). 1995 г.

5. Windows 98 – по сути, это обновлённая версия Windows 95, по-прежнему являющаяся гибридным 16/32-разрядным продуктом, основанным на MS-DOS. Улучшениям подверглась поддержка развивающейся аппаратной части, сетевых возможностей, а в интерфейс системы (Explorer.exe) интегрирован Internet Explorer 4 (функция Active Desktop). 1998 г.

6. Windows ME – Millenium Edition (в честь нового тысячелетия) – слегка обновленный продукт. 2000 г.

DOS-Windows сильно критиковались пользователями из-за их нестабильности и ненадёжности, частых зависаний и аварийных завершений работы. К примеру, в одной из статей в журнале PC World аббревиатуру ME расшифровали как «Mistake Edition» (ошибочное издание) и поместили на 4-е место в списке «Худших продуктов всех времён».

В конце концов, Microsoft полностью отказались от этой линейки продуктов, и разрешили делать с этими ОС «всё, что угодно» (а что в современных условиях с ними можно делать?), полностью перейдя на «нормальную» ветку продуктов – NT.

II.3. Технология NT

Аббревиатуру NT расшифровывают по-разному, к примеру:

– New Technology – новые технологии – наиболее часто встречающееся, официальное объяснение;

– Net Technology – сетевые технологии – т.к. все продукты в линейке Windows NT разрабатывались изначально с поддержкой сетевых технологий – такое объяснение встречается гораздо реже;

– Number Ten – номер десять – т.к. изначально система разрабатывалась под процессор фирмы Intel с внутренней маркировкой N10 – скорее всего, именно так оно и было.

Смысл в том, что системы DOS-Windows были действительно ущербные, и корпорация IBM была не слишком «довольна» распространением своих компьютеров, загруженным такими продуктами. Все чаще и чаще фирмы-потребители, да и увеличивающийся сектор домашних пользователей роптал, узнавал о «более удачных» решениях, а терять клиентов – не в интересах бизнеса.

И вот, в 1988 году Microsoft и IBM начали совместную разработку системы «по науке», т.е. по UNIX-философии, переманив специалистов из компании DEC. По этому поводу в последствии были скандалы, судебные разбирательства, которые, впрочем были «замяты» довольно крупными денежными выплатами.

Разрабатываемый проект назывался OS/2 (ОС пополам, с 1984 г.), изначально прорабатываемый силами только специалистов IBM. Система вышла в 1992 г, но с 1990 года Microsoft внедрили в OS/2 свой WinAPI, что вызвало недовольство со стороны IBM. В итоге, два производителя разорвали совместные отношения и продолжили развитие своими путями.

Всю линейку продуктов Windows NT можно представить следующим образом:

1. Windows NT 3.1 – первая операционная система линейки Microsoft Windows NT для серверов и корпоративных рабочих станций. Номер версии был выбран таким образом, чтобы соответствовать последней версии оболочки Windows 3.1, имевшей похожий интерфейс пользователя. Было доступно две редакции NT 3.1: Windows NT 3.1 и Windows NT Advanced Server. Далее практически все продукты имели настольный и серверный варианты. 1993 г.

2. Windows NT 3.5 – вторая операционная система линейки Microsoft Windows NT. Одной из главных целей разработки Windows NT 3.5 было увеличение скорости работы операционной системы. 1994 г.

3. Windows NT 4.0 – последняя версия семейства сетевых операционных систем Microsoft Windows NT, вышедшая под этим названием. Windows NT 4.0 имела пользовательский интерфейс в стиле Windows 95. 1996 г.

4. Windows 2000 (также называемая Win2k, W2k или Windows NT 5.0) — операционная система семейства Windows NT компании Microsoft, предназначенная для работы на компьютерах с 32-битными процессорами. 2000 г. (внезапно).

5. Windows XP (Windows NT 5.1, XP – eXPerience – опыт) – очередная ОС, которая, впрочем, получила наиболее широкое распространение, поддерживалась наиболее долгий срок. 2001 г.

6. Windows Vista (WinVI, Windows NT 6.0) – следующая ОС, основные изменения которой коснулись работы с памятью, управления вводом-выводом, управление питанием. Замыслы были довольно дельные, но реализацию рынок не оценил – система получилась на столько «не очень», что пользователи предпочитали продолжать пользоваться XP. 2006 г.

7. Windows 7 (WinVII, Windows NT 6.1 (опять не состыковка)) – система вобрала в себя исправления проблем WinVI, а так же поддержку новых технологий, таких, как мультитач-управление, поддержка большего количества языков за счет Unicode 5.1 и т.д. 2009 г.

8. Windows 8 (Windows NT 6.2) – система с новым графическим пользовательским интерфейсом Metro, в котором, изначально, убрали кнопку «Пуск», что вызвало шквал недовольства от пользователей, поэтому быстро выпустили обновление 8.1, в котором вернули кнопку «Пуск». Так же присутствует масса нововведений, делающих Windows похожим всё больше на Unix-системы. 2012 г.

9. Windows 10 – основана на 8.1, скорее всего та же Windows NT 6.2 ядром, но еще более похожая на Unix-системы, т.к. едина для ПК, планшетов/смартфонов, игровых приставок и т.п. Собирает и отправляет «хозяевам» всю возможную информацию об использовании – местоположение, запускаемое ПО, посещаемые ресурсы, и т.д. Шквал крити не заставил себя

ждать. Поэтому пользователи предпочитают до сих пор пользоваться Win7. 2014 г.

Сторонники теории мирового масонского заговора предпочитают не использовать системы корпорации Microsoft.

III. UNIX-подобные операционные системы

Вообще, термин UNIX был введен неким Брайаном Уилсоном Керниганом (канадский IT-ученый, соавтор языков Си, AWK), как попытку дать имя разрабатываемой системе (подразделением Bell Labs компании AT&T), означающую Uniplexed Information and Computing Service (Односложная информационная и вычислительная система). Так же многие понимают под термином UNIX – Universal Computers System – универсальную компьютерную систему. Но, такой подход к трактовке термина официальным не является.

The Open Group (промышленный консорциум, созданный для установки нейтральных открытых технологических стандартов для вычислительной инфраструктуры) обладает торговой маркой UNIX и управляет разработкой стандарта Single UNIX Specification, где слово «UNIX» используется как знак соответствия. Они не приветствуют употребление термина «UNIX-подобный» и считают, что это злоупотребление их товарным знаком. Руководства, изданные группой, требуют использования заглавных букв в названии UNIX либо выделение другим способом от остального текста, одобряют использование слова «UNIX» как прилагательного в сочетании с такими словами, как «система», и не одобряют написание через дефис (относится к английским текстам). Наиболее близкий термин, который они сочли бы корректным, был бы UNIX system-like.

В конце концов, для обозначения всего многообразия таких систем, часто используют термин «nix».

III.1. Зарождение nix`ов

В 1957 году компанией Bell Labs для собственных нужд её вычислительного центра была разработана система BESYS (Bell Operating System). Она должна была эффективно выполнять большое количество динамически загружаемых в неё коротких задач, используя перфокарты. Таким образом, она послужила некоторым прообразом для операционных систем с

разделяемым временем исполнения задач. При этом BESYS никогда не выпускалась компанией Bell Labs как законченный продукт. Хотя операционная система и использовалась в дальнейшем в различных департаментах компании, она никогда не имела должной технической поддержки. Руководителем проекта по созданию BESYS был назначен Виктор А. Высотский. Материал и идеи, собранные во время проекта BESYS, послужили отправной точкой для проекта Multics (Multiplexed Information and Computing Service, Многосложная информационная и вычислительная система), начатого в середине 60-х годов.

Multics была одной из первых операционных систем, реализовавшей плоскую модель хранения данных, чётко разделявшая концепции файлов (называемых в системе Multics сегментами) и памяти вычислительных процессов. Память вычислительных процессов состояла из сегментов, каждый из которых обладал своим адресным пространством. Для чтения или записи в сегменты вычислительный процесс использовал инструкции центрального процессора системы, а операционная система брала на себя всю заботу по сохранению изменённых данных на устройстве внешней памяти компьютера.

Таким образом, в Multics была спроектирована и реализована полностью централизованная файловая система, в которой файлы, физически располагающиеся на разных физических устройствах внешней памяти, логически объединяются в один централизованный архив или древовидную иерархическую структуру, промежуточными узлами которой являются именованные директории (или каталоги), а в листьях содержатся файлы.

Разработка операционной системы Multics была начата в 1964 году. Изначально в этом проекте были заняты Массачусетский Технологический Институт (MIT), а также компании General Electric (GE) и Bell Labs. Компания Bell Labs вышла из проекта в 1969 году, а в 1970 году компьютерный бизнес компании General Electric (вместе с Multics) отошёл к компании Honeywell, которая продала его BULL (Groupe Bull). Multics был задуман компанией General Electric как коммерческий продукт, и стал таковым уже под

руководством компании Honeywell, хотя так никогда и не нашёл успеха на компьютерном рынке.

Первоначально разработки в конце 1960-х годов велись сотрудниками Bell Labs, в первую очередь Кеном Томпсоном, Деннисом Ритчи и Дугласом Макилроем.

В 1969 году Кен Томпсон, стремясь реализовать идеи, которые были положены в основу MULTICS, но на более скромном аппаратном обеспечении (DEC PDP-7), написал первую версию новой операционной системы для миникомпьютеров 1970-х (в СССР его аналоги, выпускавшиеся Министерством электронной промышленности были известны как СМ ЭВМ и «Электроника», позже ДБК, производились в Киеве, Воронеже, Зеленограде). Эта версия получила название «первая редакция» (Edition 1) и была первой официальной версией. Системное время все реализации UNIX отсчитывают с 1 января 1970 года.

Первые версии UNIX были написаны на ассемблере и не имели встроенного компилятора с языком высокого уровня. Примерно в 1969 году Кен Томпсон при содействии Денниса Ритчи разработал и реализовал язык Би (B), представлявший собой упрощённый (для реализации на миникомпьютерах) вариант разработанного в 1966 языка BCPL. Би, как и BCPL, был интерпретируемым языком. В 1972 году была выпущена вторая редакция UNIX, переписанная на языке Би. В 1969–1973 годах на основе Би был разработан компилируемый язык, получивший название Си (C).

В 1973 году вышла третья редакция UNIX, со встроенным компилятором языка Си. 15 октября того же года появилась четвёртая редакция, с переписанным на Си системным ядром (в духе системы Multics, также написанной на языке высокого уровня ПЛ/1), а в 1975 – пятая редакция, полностью переписанная на Си.

С 1974 года UNIX стал распространяться среди университетов и академических учреждений. С 1975 года началось появление новых версий, разработанных за пределами Bell Labs, и рост популярности системы. В том же

1975 году Bell Labs выпустила шестую редакцию, известную по широко разошедшимся комментариям Джона Лайонса.

К 1978 году система была установлена более чем на 600 машинах, прежде всего, в университетах. Седьмая редакция была последней единой версией UNIX. Именно в ней появился близкий к современному интерпретатор командной строки Bourne shell.

Деннис Ритчи, один из создателей UNIX, выразил своё мнение, что UNIX-подобные системы, такие, как Linux, являются де-факто UNIX-системами. Эрик Рэймонд предложил разделить UNIX-подобные системы на 3 типа:

- Генетический UNIX: Системы, исторически связанные с кодовой базой AT&T. Большинство, но не все коммерческие дистрибутивы UNIX-систем попадают под эту категорию. Так же, как и BSD-системы, которые являются результатами работы университета Беркли в поздних 1970-х и ранних 1980-х. В некоторых из этих систем отсутствует код AT&T, но до сих пор прослеживается происхождение от разработки AT&T.

- UNIX по товарному знаку, или бренду: эти системы, в основном коммерческого характера, были определены The Open Group как соответствующие Единой спецификации UNIX, и им разрешено носить имя UNIX. Большинство этих систем – коммерческие производные кодовой базы UNIX System V в той или иной форме (например, Amiga UNIX), хотя некоторые (например, z/OS компании IBM) заслужили торговую марку через слой совместимости с POSIX, не являясь, по сути, UNIX. Многие старые UNIX-системы не подходят под это определение.

- UNIX по функциональности: В целом, любая система, поведение которой примерно соответствует спецификации UNIX. К таким системам можно отнести Linux и Minix, которые ведут себя подобно UNIX-системе, но не имеют генетических связей с кодовой базой AT&T. Большинство свободных/открытых реализаций UNIX, являясь генетическим UNIX или нет, подпадают под ограниченное определение этой категории в связи с

дороговизной сертификации The Open Group, которая стоит несколько тысяч долларов.

Cygwin, не являясь операционной системой, предоставляет UNIX-подобную среду в Microsoft Windows; также существуют сервисы Microsoft Windows для UNIX.

III.2. Многообразие nix`ов

Итак, существует великое множество систем, корень разработки которых уходит к Unix. Среди всего множества направлений развития получили особое распространение:

- BSD;
- SunOS;
- Solaris;
- MacOS X;
- iOS;
- HP-UX;
- Minix;
- GNU/Linux;
- Android.

Стоит попробовать рассмотреть их, хотя бы вкратце.

III.2.1. BSD

BSD (Berkeley Software Distribution) – система распространения программного обеспечения в исходных кодах, созданная для обмена опытом между учебными заведениями. Особенностью пакетов ПО BSD была специальная лицензия BSD, которую кратко можно охарактеризовать так: весь исходный код – собственность BSD, все правки – собственность их авторов.

В данный момент термин BSD чаще всего употребляется как синоним BSD-UNIX – общего названия вариантов UNIX, восходящих к дистрибутивам университета Беркли.

К семейству BSD относятся: NetBSD, FreeBSD, OpenBSD, ClosedBSD, MirBSD, DragonFly BSD, PC-BSD, DesktopBSD, SunOS, TrueBSD, Frenzy, Ultrix и частично Darwin (ядро Mac OS X).

В начале 1978 года аспирант Университета Калифорнии в Беркли Билл Джой начал создание дистрибутива программ. Первая магнитная лента содержала систему Pascal для UNIX и редактор ex; распространено около 30 копий. До конца 1978 года распространялся пакет 2BSD, продано около 75 копий. Приблизительно тогда же компания Interactive Systems выпускает первую коммерческую версию UNIX, а компания Whitesmiths – первый клон системы UNIX – Idris.

Начиная с версии 2.79BSD, программные дистрибутивы университета Беркли перестали содержать код AT&T UNIX. Версия UNIX 7 была первой переносимой операционной системой. Однако её производительность была хуже, чем в версии шесть. После многих доработок в январе 1982 года вышла в свет модель 2.8.1BSD.

Впоследствии вышли версии:

4BSD – октябрь 1980 года

4.1BSD – июнь 1981 года

4.1a, 4.1b и 4.1c (1982–1983)

4.2 – сентябрь 1983 года – сильно переработанная система, включавшая поддержку сетей TCP/IP, новую файловую систему и возможность использования сигналов

Последней университетской версией была 4.4BSD (июнь 1993 года). Из нововведений эта версия включала виртуальную файловую систему, поддержку сетевой файловой системы NFS, отладчик ядра и мощную реализацию стека протоколов TCP/IP.

После суда права на BSD были переданы компании BSDI (Berkeley Software Design, Inc).

На базе BSD-UNIX создано множество операционных систем, преимущественно также с открытыми исходными текстами.

III.2.2. SunOS

SunOS – версия операционной системы UNIX, разработанная компанией Sun Microsystems для рабочих станций и серверов собственного производства. Название SunOS обычно используется для обозначения версий с 1.0 по 4.1.4. Эти версии были основаны на BSD варианте UNIX, тогда как SunOS версии 5.0 и далее основана на UNIX System V четвертого выпуска (SVR4) и фигурирует под названием Solaris. Одним из основателей и руководителем исследовательского отдела компании Sun Microsystems был тот самый Билл Джой, который, в научном руководстве Кена Томпсона, разрабатывал BSD.

III.2.3. Solaris

Solaris – операционная система, разработанная компанией Sun Microsystems для платформы SPARC, с 2008 года принадлежит вместе с активами Sun корпорации Oracle. Несмотря на то, что Solaris – операционная система с закрытым исходным кодом, большая его часть открыта и опубликована в проекте OpenSolaris.

В начале 1990-х годов Sun Microsystems заменила основанную на BSD SunOS 4 на UNIX System V Release 4 (SVR4), разрабатываемую совместно с AT&T, а также изменила имя SunOS 5 на Solaris 2. После выхода версии 2.6 Sun Microsystems отбросила из имени «2.», и следующая версия называлась уже Solaris 7.

В июне 2005 года Sun Microsystems приняла решение открыть значительную часть исходного кода последней версии системы – Solaris 10, и запустить проект OpenSolaris. Стандартные бинарные сборки Solaris 10 были размещены на веб-сайте Sun Microsystems и доступны всем желающим по лицензии CDDL, после регистрации на веб-сайте компании.

III.2.4. MacOS X

OS X (Mac OS X до версии 10.7 включительно) – проприетарная операционная система производства Apple. Является преемницей Mac OS 9.

Семейство операционных систем OS X является вторым по распространенности (после Windows). Рыночная доля OS X (учитываются все версии) по состоянию на июль 2015 составляла от 4.74% до 8.13%.

В OS X используется ядро XNU, основанное на микроядре Mach и содержащее программный код, разработанный компанией Apple, а также код из ОС NeXTSTEP и FreeBSD. До версии 10.3 OS X работала только на компьютерах с процессорами PowerPC. Выпуски 10.4 и 10.5 поддерживали как PowerPC-, так и Intel-процессоры. Начиная с 10.6, OS X работает только с процессорами Intel.

III.2.5. iOS

iOS (до 24 июня 2010 года – iPhone OS) – операционная система для смартфонов, электронных планшетов и носимых проигрывателей, разрабатываемая и выпускаемая американской компанией Apple. Была выпущена в 2007 году; первоначально – для iPhone и iPod touch, позже – для таких устройств, как iPad и Apple TV. В отличие от Windows Phone (Microsoft) и Android (Google), выпускается только для устройств, производимых фирмой Apple.

В iOS используется ядро XNU, основанное на микроядре Mach и содержащее программный код, разработанный компанией Apple, а также код из ОС NeXTSTEP и FreeBSD (да-да, как и MacOS X). Ядро iOS почти идентично ядру настольной операционной системы Apple OS X. Начиная с самой первой версии, iOS работает только на планшетных компьютерах и смартфонах с процессорами архитектуры ARM.

В последние годы отмечается взаимная интеграция OS X и iOS. Сама компания рассматривает две ОС как единую платформу (не удивительно). На презентации Back to the Mac в 2010 году Стив Джобс, анонсируя OS X Lion, упомянул о важности обмена наработками между OS X и iOS: так, в Lion появилась поддержка мультитач-жестов на трекпаде (аналогичных жестам на iPad). В последующих выпусках OS X также наблюдалась тенденция к заимствованию функций из iOS (и наоборот). Например, в OS X Yosemite и iOS

8 была добавлена технология Handoff, позволяющая «перехватывать» с одного устройства приложения, запущенные на другом: начать набирать письмо на Mac, а закончить на iPad; открыть веб-страницу на iPhone и продолжить чтение на Mac и т. д.

III.2.6. HP-UX

HP-UX – проприетарная версия операционной системы UNIX фирмы Hewlett-Packard (HP). Работает на архитектурах PA-RISC, Intel Itanium и Apollo/Domain. Ранние версии также работали на процессорах HP 9000 Series 200, 300 и 400, основанных на семействе процессоров Motorola 68000, и HP 9000 Series 500 собственной архитектуры HP FOCUS.

В HP-UX впервые была представлена система списков контроля доступа для контроля доступа к файлам вместо традиционной системы разрешений UNIX. HP-UX была среди первых UNIX-систем со встроенным менеджером логических дисков.

III.2.7. Minix

Minix – свободная Unix-подобная микроядерная операционная система, распространяемая по лицензии BSD. Эндрю Таненбаум создал первую версию Minix в 1987 в качестве «иллюстрации» учебника «Операционные системы: Разработка и реализация». Среди студентов, изучающих теорию операционных систем по исходникам Minix, наибольшей известности достиг Линус Торвалдс, разработчик Linux. Изначально, целью Minix являлось обучение студентов строению компьютеров и основ операционных систем. Исходные коды Minix совершенно отличны от кодов ОС Linux, но как и Linux, Minix является Unix-подобной операционной системой с аналогичным набором команд.

Исходный код MINIX 3 с переведёнными на русский язык комментариями можно получить на CD вместе с книгой «Операционные системы: разработка и реализация» (Таненбаум Э. С., Вудхалл А. С., 3-е издание опубликовано в России в 2007 году), в которой детально расписано строение ОС на примере Minix.

1 ноября 2008 года Эндрю Таненбаум получил грант в 2,5 млн евро от Европейского исследовательского совета для работы над проектом: «Исследование действительно надёжного и безопасного системного программного обеспечения». Грант позволит продолжить работы по дальнейшему развитию ОС MINIX 3 в течение не менее пяти лет (до ноября 2013 года).

Изначально Таненбаум разработал Minix для компьютеров IBM PC и IBM PC/AT, доступных в то время. Minix версии 1.5 был портирован на Motorola 68000, что обеспечило совместимость с популярными платформами Amiga, Atari ST, Apple Macintosh. Кроме того, были сделаны версии для SPARC, National Semiconductor NS32032 и транспьютеров.

III.2.8. GNU/Linux

Linux – общее название Unix-подобных операционных систем, основанных на одноимённом ядре. Ядро Linux создаётся и распространяется в соответствии с моделью разработки свободного и открытого программного обеспечения. Поэтому общее название не подразумевает какой-либо единой «официальной» комплектации Linux; они распространяются в основном бесплатно в виде различных готовых дистрибутивов, имеющих свой набор прикладных программ и уже настроенных под конкретные нужды пользователя. Первый релиз ядра системы состоялся 5 октября 1991 (этот праздник IT-шники любят отмечать соответствующим образом).

На начальном этапе Linux бесплатно разрабатывался только энтузиастами-добровольцами, но с успехом Linux и его массовым коммерческим использованием дорабатывать ОС и вносить свой вклад стали и компании, со временем став значительной силой. Подавляющее большинство ПО в современных дистрибутивах по-прежнему доступно по свободным лицензиям, как правило, за исключением небольшого количества проприетарных компонентов. В 2008 году расчёты показывали, что для того, чтобы «с нуля» разработать систему, аналогичную Fedora 9, потребовалось бы затратить 10,8 млрд долл. Совокупная себестоимость ядра Linux оценена в

более чем 1 млрд евро (около 1,4 млрд долл.). Только за 2008 год себестоимость ядра Linux увеличилась на 225 млн евро. В системе Linux воплощён труд в эквиваленте 73 тыс. человеко-лет.

В настоящее время системы Linux лидируют на рынках смартфонов (Android занимает 85% рынка), интернет-серверов (60%), самых мощных суперкомпьютеров (97%), а также, согласно Linux Foundation, в дата-центрах и на предприятиях, занимают половину рынка встраиваемых систем, имеют значительную долю рынка нетбуков (32% на 2009 год). На рынке домашних компьютеров Linux прочно занимает 3 место (по разным данным, от 1 до 5%). Согласно исследованию Goldman Sachs, в целом, рыночная доля Linux среди электронных устройств составляет около 42%.

С тех пор, как ядро Linux было создано для x86-ПК, оно было портировано на множество платформ, включая x86-64, PowerPC и ARM. Linux работает в роутерах, телевизорах и игровых приставках. ОС на этом ядре продолжают быстро совершенствоваться (например, новая версия ядра выпускается каждые 2–3 месяца, с 2005 года в разработке ядра принимают участие более 7800 разработчиков из более чем 800 различных компаний) и набирает популярность (за 9 месяцев с мая 2011 по январь 2012 доля Linux выросла на 64%).

Наиболее популярными дистрибутивами являются (расположены в алфавитном порядке названия их пакетных форматов): deb-based (Debian, Mint, Ubuntu), pacman-based (Arch Linux, Chakra, Manjaro), RPM-based (RedHat, Fedora, Mageia, OpenSUSE), source-based (Slackware, Gentoo).

III.2.9. Android

Android – операционная система для смартфонов, интернет-планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков, очков Google, телевизоров и других устройств. В будущем планируется поддержка автомобилей и бытовых роботов. Основана на ядре Linux и собственной реализации виртуальной машины Java от Google. Изначально разрабатывалась компанией Android Inc.,

которую затем купила Google. Впоследствии Google инициировала создание альянса Open Handset Alliance (ОНА), который сейчас занимается поддержкой и дальнейшим развитием платформы. Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit позволяет портировать библиотеки и компоненты приложений, написанные на Си и других языках.

В 86% смартфонов, проданных во втором квартале 2014 года, была установлена операционная система Android. При этом за весь 2014 год было продано более 1 миллиарда Android-устройств.

Таким образом, nix-подобных операционных систем количественно гораздо больше, чем операционных систем для настольных компьютеров компании Microsoft, да и современные Windows-системы, основанные на технологии NT, на самом деле «растут» из философии nix и развиваются в сторону nix. Из всего многообразия nix-подобных ОС наиболее ярко выделяются системы, основанные на ядре Linux, поэтому их стоит рассмотреть чуть подробнее.

III.2.10. Файловые системы unix-подобных ОС

Unix File System (UFS) – файловая система, созданная для операционных систем семейства BSD и используемая в переработанном и дополненном виде на данный момент как основная в операционных системах-потомках (FreeBSD, OpenBSD, NetBSD). Поддержка данной файловой системы имеется также в ядре Linux и операционной системе Solaris.

Физически UFS состоит из следующих частей:

- несколько блоков в начале раздела отводится под загрузочную область (которая должна инициализироваться отдельно от файловой системы);
- суперблок, включающий магическое число, инициализирующее файловую систему, и некоторые другие важные числа, описывающие геометрию и настройку некоторых параметров файловой системы;
- описание групп цилиндров. Каждая группа включает следующие компоненты:

1. Резервную копию суперблока
2. Заголовок группы цилиндров, статистические данные и т. д., информацию аналогичную содержащейся в суперблоке, но для конкретной группы
3. Некоторое количество индексных дескрипторов, каждый из которых содержит атрибуты файлов
4. Некоторое количество блоков данных

Индексные дескрипторы нумеруются последовательно. Несколько первых индексных дескрипторов сохранены по историческим причинам, далее следуют индексные дескрипторы корневого каталога.

Каталог файлов содержит только список файлов в директории и индексный дескриптор, связанный с каждым файлом. Все метаданные файла хранятся в индексном дескрипторе.

XFS – начало разработки 1993 год, фирма Silicon Graphics, в мае 2000 года предстала в GNU GPL, для пользователей большинства Linux систем стала доступна в 2001-2002 гг. Отличительная черта системы – прекрасная поддержка больших файлов и файловых томов, 8 эксбибайт – 1 байт (8260-1 байт) для 64-х битных систем. Ко всему прочему обладает другими немаловажными особенностями – непрерывные области дискового пространства, задержка выделения пространства и онлайн дефрагментация. Является одной из старейших журналируемых файловых систем для nix, и содержит в себе наиболее отлаженный, в этом контексте, исходный код.

ReiserFS (Reiser3) – одна из первых журналируемых файловых систем под Linux, разработана Namesys. Имеет некоторые врождённые головные боли, но в целом неплохая система, ведущая отсчёт дней своих с 2001 года. Оговорюсь, что смысл журналируемых систем заключается в дисковых транзакциях, которые последовательно пишутся в специальную зону диска (журнал, он же лог), перед тем как данные попадают в конечные точки файловой системы. Максимальный объём тома для этой системы равен 16 тебибайт (16240 байт).

JFS (Journaled File System) – файловая система, детище IBM, явившееся миру в далёком 1990 году для ОС AIX (Advanced Interactive eXecutive). В виде первого стабильного релиза, для пользователей Linux, система стала доступна в 2001 году. Из плюсов системы – неплохая масштабируемость. Из минусов – не особо активная поддержка на протяжении всего жизненного цикла. Максимальный размер тома 32 пэббайта (32250 байт).

ext (extended filesystem) – появилась в апреле 1992 года, это была первая файловая система, изготовленная специально под нужды Linux ОС. Разработана Remy Card с целью преодолеть ограничения файловой системы Minix.

ext2 (second extended file system) – была разработана Remy Card в 1993 году. Не журналируемая файловая система, это был основной её недостаток, который исправит ext3.

ext3 (third extended filesystem) – по сути расширение исконной для Linux ext2, способное к журналированию. Разработана Стивенем Твиди (Stephen Tweedie) в 1999 году, включена в основное ядро Linux в ноябре 2001 года. На фоне других своих сослуживцев обладает более скромным размером пространства, до 4 теббайт (4240 байт) для 32-х разрядных систем. На данный момент является наиболее стабильной и поддерживаемой файловой системой в среде Linux.

Reiser4 – первая попытка создать файловую систему нового поколения для Linux. Впервые представленная в 2004 году, система включает в себя такие передовые технологии как транзакции, задержка выделения пространства, а так же встроенная возможность кодирования и сжатия данных. Ханс Рейзер (Hans Reiser), главный разработчик системы, рекламировал использовать своё детище непосредственно как БД с улучшенными метаданными. После того, как Ханс Рейзер был осуждён за убийство в 2008 году, дальнейшая судьба системы стала сомнительной.

ext4 – попытка создать 64-х битную ext3 способную поддерживать больший размер файловой системы (1 эксбайт). Позже добавились возможности – непрерывные области дискового пространства, задержка

выделения пространства, онлайн дефрагментация и прочие. Обеспечивается прямая совместимость с системой ext3 и ограниченная обратная совместимость при недоступной способности к непрерывным областям дискового пространства.

UPD: Btrfs (B-tree FS или Butter FS) – проект изначально начатый компанией Oracle, впоследствии поддержанный большинством Linux систем. Многие считают систему эдаким ответом на ZFS. Ключевыми особенностями данной файловой системы являются технологии: copy-on-write, позволяющая сделать снимки областей диска (снапшоты), которые могут пригодиться для последующего восстановления; контроль за целостностью данных и метаданных (с повышенной гарантией целостности); сжатие данных; оптимизированный режим для накопителей SSD (задаётся при монтировании) и прочие. Немаловажным фактором является возможность перехода с ext3 на Btrfs. С августа 2008 года данная система выпускается под GNU GPL.

Tux2 – известная, но так и не анонсированная публично файловая система. Создатель Дэниэл Филипс (Daniel Phillips), система базируется на алгоритме «Фазового Древа», который как и журналирование защищает файловую систему от сбоев. Организована как надстройка на ext2.

Tux3 – наступая на пятки Btrfs, представлена новая файловая система. Система создана на основе FUSE (Filesystem in Userspace), специального модуля для создания файловых систем на Unix платформах. Данный проект ставит перед собой цель избавиться от привычного журналирования, взамен предлагая версионное восстановление (состояние в определённый промежуток времени). Преимуществом используемой в данном случае версионной системы, является способ описания изменений, где для каждого файла создаётся изменённая копия, а не переписывается текущая версия. Такой подход позволяет более гибко управлять версиями.

UPD: Xiafs – задумка и разработка данной файловой системы принадлежат Frank Xia, основана на файловой системе MINIX. В настоящее время считается устаревшей и практически не используется. Наряду с ext2

разрабатывалась, как замена системе ext. В декабре 1993 года система была добавлена в стандартное ядро Linux. И хотя система обладала большей стабильностью и занимала меньше дискового пространства под контрольные структуры – она оказалась слабее ext2, ведущую роль сыграли ограничения максимальных размеров файла и раздела, а так же способность к дальнейшему расширению.

UPD: ZFS (Zettabyte File System) – изначально созданная в Sun Microsystems файловая система, для небезызвестной операционной системы Solaris в 2005 году. Отличительные особенности – отсутствие фрагментации данных как таковой, возможности по управлению снапшотами (snapshots), пулами хранения (storage pools), варьируемый размер блоков, 64-х разрядный механизм контрольных сумм, а так же способность адресовать 128 бит данных. В Linux системах может использоваться посредством FUSE.

III.3. Linux-системы

Среди всего множества систем, основанных на Linux, выделим, так называемую, deb-ветку, которая заняла наибольший объем использования. Это Debian, Ubuntu и другие. Rpm и другие версии в данном методическом пособии рассматриваться не будут. Не реально рассмотреть все Linux и в толстом учебнике.

III.3.1. Debian

Debian – операционная система, состоящая из свободного ПО с открытым исходным кодом. В настоящее время Debian GNU/Linux – один из самых популярных и важных дистрибутивов GNU/Linux, в первичной форме оказавший значительное влияние на развитие этого типа ОС в целом. Также существуют проекты на основе других ядер: Debian GNU/Hurd, Debian GNU/kFreeBSD и Debian GNU/kNetBSD. Debian может использоваться в качестве операционной системы как для серверов, так и для рабочих станций.

Debian имеет наибольшее среди всех дистрибутивов хранилище пакетов – готовых к использованию программ и библиотек – и, если даже не по их числу,

то по числу поддерживаемых архитектур: начиная с ARM, используемой во встраиваемых устройствах, наиболее популярных x86 и PowerPC, новых 64-разрядных AMD, и заканчивая IBM S/390, используемой в мейнфреймах. Для работы с хранилищем разработаны разные средства, самое популярное из которых – Advanced Packaging Tool (APT).

Debian стал основой целого ряда дистрибутивов. Самые известные из них (в алфавитном порядке) – Knoppix, Linux Mint, Maemo, MEPIS, SteamOS, Ubuntu.

Название «Debian» составлено из имён основателя проекта Яна Мёрдока (Ian Murdock) и его подруги (впоследствии – жены, ныне – бывшей) Дебры Линн (Debra Lynn).

Создание Debian было начато в августе 1993 года Яном Мёрдоком. Он руководствовался желанием создать дистрибутив, процесс рождения которого будет, с одной стороны, открытым и свободным в духе GNU/Linux, и одновременно исключительно тщательным и добросовестным. Сначала над проектом работала небольшая сплочённая группа хакеров мира свободного ПО, которая постепенно разрасталась, став большим организованным сообществом разработчиков и пользователей.

III.3.2. Ubuntu

Ubuntu (с зулусского ubuntu – человечность, как бы намекая, что эта ОС для людей) – операционная система, основанная на Debian GNU/Linux. Основным разработчиком и спонсором является компания Canonical. В настоящее время проект активно развивается и поддерживается свободным сообществом.

По утверждениям Canonical, Ubuntu используется примерно 20 миллионами пользователей. Он является 4-м в списке самых популярных дистрибутивов Linux для веб-серверов. По версии DistroWatch.com (на январь 2016 года) занимает 3-е место по популярности для десктопов.

Изначально Ubuntu создавалась как временное ответвление от Debian с целью регулярно выпускать новую версию операционной системы каждые

шесть месяцев. В отличие от других ответвлений Debian общего назначения, таких как Xandros, Linspire и Libranet, Canonical осталась близка к философии Debian и включает в Ubuntu в основном свободное программное обеспечение вместо того, чтобы частично положиться на несвободные добавления. Пакеты Ubuntu по большей части базируются на пакетах из нестабильной (unstable) группы пакетов Debian. В Ubuntu используется Advanced Packaging Tool от Debian для управления установленными пакетами. Тем не менее, пакеты для Ubuntu и Debian не обязательно совместимы друг с другом. Некоторые разработчики Ubuntu также занимаются ключевыми пакетами Debian, поэтому в случае внесения изменений в собираемые программы они вносятся в оба проекта. Однако в апреле 2005 основатель Debian Ян Мёрдок критиковал Ubuntu за несовместимость с пакетами Debian, говоря, что Ubuntu слишком далеко отклонился от Debian Sarge, чтобы остаться совместимым.

Заключение

«Теоретические аспекты ОС» – часть методического комплекса по учебной дисциплине «Операционные системы и среды» для специальности «Прикладная информатика (по отраслям)» Волгоградского филиала «Московский государственный гуманитарно-экономический университет», разрабатываемого преподавателем А.Б. Вахраневым.

В данной части описаны основы дисциплины, раскрыты теоретические аспекты дидактических единиц, обозначенных в стандарте ФГОС 09.02.05.

Цель разработки методического комплекса – помочь обучающимся при подготовке к экзамену по данной дисциплине, запланированному в соответствии с учебным планом данной специальности. В конце пособия приведен краткий список рекомендуемой литературы и Интернет-ресурсов.

Рекомендуемая литература

1. Э. Таненбаум. Современные операционные системы, 4-е издание. Питер, 2015
2. Р. Моримото, М. Ноэл, Г. Ярдени. Microsoft Windows Server 2012. Полное руководство. Вильямс, 2013
3. М. Матвеев, М. Юдин, Р. Прокди. Windows 8. Полное руководство. Наука и техника, 2013
4. В. Комягин, Ф. Резников. Устанавливаем и настраиваем Ubuntu Server 2012-2015 и офисные ПК с Ubuntu. Триумф, 2012
5. Д. Колисниченко. FreeBSD. От новичка к профессионалу. БХВ, 2012
6. Э. Немец, Г. Снайдер, Т. Хейн. Unix и Linux. Руководство системного администратора. Вильямс, 2015

Интернет-источники:

1. В мире с компьютером – www.whatis.ru
2. Свободная Интернет-энциклопедия «Википедия» – ru.wikipedia.org
3. Интернет-университет информационных технологий «Интуит» – intuit.ru
4. Свободная Интернет-библиотека «Викиучебник» – ru.wikibooks.org
5. Free eBooks from Microsoft Press – mva.microsoft.com/ebooks
6. IT-публикации «Хабрахабр» – habrahabr.ru