

ВОЕННО-КОСМИЧЕСКАЯ АКАДЕМИЯ ИМЕНИ А.Ф. МОЖАЙСКОГО

Кафедра № 63 Математического и программного обеспечения

УТВЕРЖДАЮ

Начальник 63 кафедры
полковник _____ С.Войцеховский

« ____ » _____ 2015 г.

Автор: преподаватель 63 кафедры
Кандидат технических наук
майор С.Краснов

Лекция № 16

Тема: «ОБЩИЕ СВЕДЕНИЯ ОБ ОПЕРАЦИОННОЙ СИСТЕМЕ МСВС 3.0»

по дисциплине: «Защита информации»

Обсуждено и одобрено на заседании 63 кафедры
протокол № ____ « ____ » _____ 2015 г.

Санкт-Петербург
2015

Содержание занятия и время

Введение – 5 мин.

Учебные вопросы (основная часть):

1. Общие сведения о реализации защиты ИПО в операционных системах – 30 мин.
2. Назначение состав и основные возможности ОС МСВС – 20 мин.
3. Основные задачи администратора системы (подключение пользователей, резервное копирование, мониторинг системы и др.) – 30 мин.

Заключение – 3-5 мин.

Литература:

Основная:

1. Войцеховский С.В., Воробьев Е.Г. Методы и средства защиты компьютерной информации: учебно-методическое пособие. – СПб.: ВКА имени А.Ф. Можайского 2013. – 134 с.
2. Воробьев Е.Г. Защита информации в автоматизированных системах военного назначения. Учебное пособие – СПб, ВКА им.А.Ф. Можайского, 2007. – 252 с.

Дополнительная:

1. Вихорев С.В. Классификация угроз информационной безопасности. - http://www2.cnews.ru/comments/security/elvis_class.shtml
2. Войцеховский С.В., Марковский А.С., Палагушин В.А. Защита информации в автоматизированных системах. / Под ред. профессора Хомоненко А.Д. – СПб.:НТЦ им. Л.Т. Тучкова, 2005. – 149 с.

Материально техническое обеспечение:

1. Технические средства обучения: ПЭВМ, мультимедиа проектор, экран, программное обеспечение.
2. Приложения (слайды).
3. Наглядные средства обучения – доска, мел.

Организационно-методические указания:

Цель лекции: *Дать знания в области защиты ОС МС ВС.*

Во введении сформулировать тему лекции, цель и название изучаемых вопросов. При этом произвести опрос курсантов по пройденному материалу.

Применяемым методическим приемом является рассказ.

В основной части сконцентрировать внимание курсантов механизмах защиты ОС МС ВС.

В заключительной части обобщить изложенный материал и осуществить контрольный опрос.

1. Перечислите особенности защиты в ОС МС ВС?
2. Охарактеризуйте ОС МС ВС?
3. Перечислите основные задачи администратора безопасности в ОС МС ВС?

Отвечаю на вопросы по теме занятия, даю задание на самостоятельную подготовку.

Тема № 5 Механизмы защиты операционных систем

«Общие сведения об ОС МСВС 3.0»

В. 1. Общие сведения о реализации защиты ИПО в ОС.

Рассмотрим два варианта подключения автоматизированной системы к сети Интернет. В одном случае сеть Интернет непосредственно примыкает к АС (рис.1 а)), в другом между ними находится система защиты (рис.1 б)).

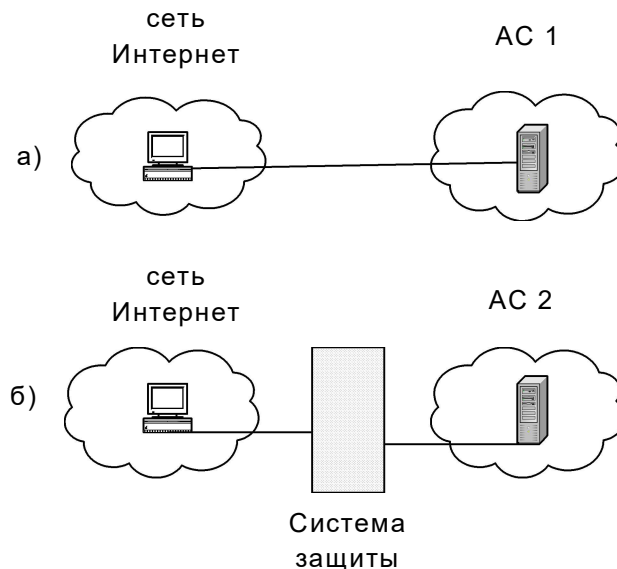


Рис 1. Варианты подключения АС к сети Интернет

Если все угрозы (атаки), исходящие из сети Интернет, принять за A , а достигшие АС за X , то

для рис. 1 а) $X=A$,

для рис. 1 б) $X<A$.

В идеальном случае количество угроз, достигших АС (X), должно быть равно нулю ($X=0$).

первый эшелон (защиты ядра) осуществляет защиту рабочих станций (ПЭВМ и серверов) от различных типов угроз. В него могут входить следующие средства защиты:

1. Штатные средства защиты ОС, в которые могут входить следующие механизмы защиты:
 - идентификации;
 - аутентификации;
 - разграничения доступа;
 - аудита;
 - блокировки бесконтрольного доступа и другие.
2. Антивирусные пакеты.
3. Дополнительные устройства аутентификации пользователя (смарт-карты, биометрические системы, персональный идентификатор и др.).
4. Системы разграничения доступа (локальные средства защиты от НСД).
5. Персональные межсетевые экраны.

Все эти средства защиты объединяет одно – они устанавливаются на локальные ПЭВМ или сервера и защищают только себя.

К ОС функционирующим в составе АС объектов ВТ КВ предъявляются (в соответствии с ТЗ на объект ВТ) определённые требования в соответствии с классом АС (например если

АС многопользовательская и в ней обрабатывается секретная информация, то класс АС должен быть не ниже 1В). Зачастую ОС (Windows) не имеет сертификата по нужному классу, тогда в АС устанавливают дополнительные средства ЗИ (например, Secret Net), которые призваны устранить имеющиеся уязвимости ОС и повысить до необходимого уровня класс АС.

С появлением первой отечественной ОС МСВС 3.0 имеющей множество различных механизмов защиты, сертифицированной по классу 1Б и позволяющей обрабатывать совершенно секретную информацию, необходимость в обязательном применении дополнительных средств ЗИ отпала, хотя данная ОС может дополнительно оснащаться СЗИ «Цезарь».

В. 2. Назначение состав и основные возможности ОС МСВС 3.0

ОС МСВС 3.0 ФЛИР.80001-01 - это мобильная, многопользовательская, многозадачная операционная система, поддерживающая симметричные многопроцессорные архитектуры и работающая как в режиме командной строки, так и в режиме графического интерфейса.

Производитель – концерн ВНИИНС г. Москва (Россия).

Основное *назначение ОС МСВС 3.0* - управление ресурсами системы и процессами, использующими эти ресурсы при вычислениях.

ОС МСВС 3.0 - это инструментальное средство для управления техническими средствами и задачами в следующих областях:

- автоматизированные системы управления производством;
- автоматизированные системы управления технологическим процессом;
- информационные системы;
- системы массового обслуживания;
- системы сбора и анализа информации;
- многопользовательские системы общего назначения.

ОС МСВС 3.0 - это программное изделие, поставляемое в виде загрузочного модуля и комплекта эксплуатационной документации. Загрузочный модуль поставляется на CD-ROM, а комплект эксплуатационной документации поставляется на бумажном носителе.

В состав ОС МСВС 3.0 входят четыре комплекса:

- Базовая конфигурация ОС ФЛИР.91100-01;
- Система графического интерфейса ФЛИР.91200-01;
- Система защиты от НСД ФЛИР.91300-01;
- Средства разработки ФЛИР.91400-01.

ОС МСВС 3.0 предоставляет различным категориям пользователей средства по реализации широкого спектра функциональных возможностей. Для программистов и системных программистов - это средства разработки и отладки прикладных и системных программ, для администраторов систем и сетей - это высокоэффективные средства установки и управления системами и сетями, а для рядовых пользователей - это средства эффективной работы с различными прикладными программами.

Системные возможности ОС МСВС 3.0:

- разработка и отладка программ;
- выполнение прикладных программ;
- работа в режиме командной строки;
- работа с файлами;
- работа с внешними устройствами;
- администрирование системы.

Сетевые возможности ОС МСВС 3.0

ОС МСВС 3.0 предоставляет различным категориям пользователей возможности по выполнению базовых сетевых функций, выполнению сетевых служб и приложений и по администрированию сетей. Сетевые возможности ОС МСВС 3.0:

- выполнение базовых сетевых функций;

- работа в режиме клиент-сервер;
- работа в режиме распределенных вычислений;
- передача файлов;
- удаленный терминальный доступ;
- передача почтовых сообщений;
- доступ к файловым системам удаленных ЭВМ;
- доступ к сетевой информационной службе;
- доступ к службе доменных имен;
- администрирование сети.

ОС MSVC 3.0 является сетевой операционной системой, в которой содержатся сетевые службы, реализующие функционально полный набор протоколов TCP/IP для организации работы в локальных (по протоколу Ethernet) и распределенных (по протоколам SLIP, PLIP и PPP) сетях.

В.3.. Основные задачи администратора системы

Администрирование системы осуществляется администратором системы, который должен выполнять такие *основные задачи*:

- управление пользователями, настройка устройств,
- создание резервных копий,
- выключение системы,
- обучение пользователей,
- обеспечение безопасности системы,
- ведение системного журнала и помощь пользователям.

1.2.1. Подключение и удаление пользователей

Создание бюджетов для новых пользователей и удаление бюджетов тех пользователей, которые уже не работают, является обязанностью администратора системы. Процесс подключения и удаления пользователей можно автоматизировать, но некоторые решения, от которых зависит включение нового пользователя, должен принимать администратор системы.

Если необходимо прекратить доступ пользователя к системе, его бюджет должен быть отключен. Все файлы, относящиеся к этому бюджету, необходимо удалить, чтобы они не занимали места на жестком магнитном диске (далее по тексту -жесткий диск).

1.2.2. Подключение и удаление технических средств

При подключении технических средств, систему нужно сконфигурировать таким образом, чтобы она распознала и использовала эти средства. Изменение конфигурации может быть как простой задачей (например, подключение принтера), так и более сложной (например, подключение дисководов).

1.2.3. Резервное копирование

Выполнение резервного копирования является одной из наиболее важных задач администраторов систем. Процедура резервного копирования довольно утомительна и занимает много времени, но выполнять ее необходимо. Ее можно автоматизировать или поручить оператору, но в любом случае системный администратор обязан убедиться в том, что резервное копирование выполнено правильно и в соответствии с установленным графиком.

1.2.4. Установка новых программных средств

После приобретения нового программного обеспечения его нужно установить и протестировать. Если программы работают нормально, пользователям необходимо сообщить необходимые данные о новых программах (какие программы, каталоги, где они находятся и т.д.). Целесообразно новое программное обеспечение установить там, где его можно будет легко отличить от программных средств, поставляемых в составе ОС MSVC 3.0.

1.2.5. Мониторинг системы

Существует множество обязательных для исполнения ежедневных операций. Это, например, проверка правильности функционирования электронной почты, просмотр регистрационных файлов на предмет наличия ранних признаков неисправностей, контроль за наличием системных ресурсов (в частности, контроль за наличием свободного пространства на диске) и контроль за подключением локальных вычислительных сетей.

1.2.6. Поиск неисправностей

Из-за неверных действий пользователей или из-за появления технических неисправностей в технических средствах, возможны сбои в системе. Задача администратора - диагностировать сбои в системе и при необходимости принять меры по их устранению.

1.2.7. Контроль защиты

Администратор системы должен реализовывать стратегию обеспечения безопасности и периодически проверять, не нарушена ли защита системы. В системах с низким уровнем безопасности эта процедура может быть сведена всего лишь к нескольким текущим проверкам на предмет несанкционированного доступа. В системах с высоким уровнем безопасности обычно применяется сложная система ловушек и программ контроля.

1.2.8. Ведение документации

Настраивая конфигурацию под конкретные требования, вы можете обнаружить, что она значительно отличается от первоначальной конфигурации, описанной в документации. Поэтому администратор системы должен документировать все устанавливаемые программные средства, не входящие в стандартный комплект поставки, документировать конфигурацию локальных вычислительных сетей, вести записи по обслуживанию всех технических средств, регистрировать состояние резервных копий, документировать процедуры и правила работы с системой.

1.2.9. Оказание помощи пользователям

Оказание помощи пользователям необходимо в решении различных проблем, которые у них возникают в процессе работы. Например: не выполняется прикладная программа, не печатает принтер, хотя задание поставлено в очередь на печать, нет доступа к удаленной ЭВМ и др. Обучение пользователей предусматривается для повышения эффективности их труда.

1.3. *Возможности администратора системы*

Административное управление в ОС МСВС 3.0 отделено от общепользовательского доступа.

Администраторам систем приходится отменять действие защитных механизмов ОС МСВС 3.0 в самых разных ситуациях. Для обеспечения такой возможности система выделяет среди всех идентификаторов пользователей особый идентификатор, который принадлежит привилегированному пользователю. ОС МСВС 3.0 определяет для этого UID пользовательский бюджет под именем "root".

ОС МСВС 3.0 позволяет привилегированному пользователю выполнять над файлом или процессом любую операцию. Кроме того, некоторые системные вызовы (обращения к ядру) может выполнять только привилегированный пользователь. Некоторые системные вызовы доступны всем пользователям, но имеют специальные опции для пользователя root.

Примеры операций, которые может выполнить только привилегированный пользователь:

- монтирование и демонтирование файловых систем;
- изменение командой `chroot` корневого каталога процесса;
- создание файлов устройств;
- установка системных часов;
- изменение принадлежности файлов;
- увеличение лимитов использования ресурсов и назначение приоритетов процессов;
- задание `host`-имени системы;
- конфигурирование сетевых интерфейсов;
- остановка системы.

1.4. *Пароль привилегированного пользователя*

Пароль пользователя root следует выбирать так, чтобы его нельзя было определить. Теоретически наиболее безопасный пароль состоит из случайной последовательности букв, знаков препинания и цифр. Такой пароль, однако, тяжело запомнить и, как правило, трудно вводить. Администратору системы не рекомендуется записывать пароль или вводить его слишком медленно.

Как правило, достаточно надежен пароль, состоящий из двух случайно выбранных слов, разделенных знаком препинания, или из первых букв какой-нибудь фразы, набранных в разных регистрах. Такой пароль соответствует требованиям, предъявляемым к паролям в некоторых системах, где требуется, чтобы каждый пароль содержал как строчные, так и прописные буквы или как минимум одну цифру или спецсимвол. Пароль пользователя root должен состоять из восьми символов, так как пароли, включающие семь символов, менее надежны. Задавать более длинный пароль не имеет смысла, потому что система обрабатывает только первые восемь символов.

Пароль привилегированного пользователя следует менять:

- минимум раз в три месяца;
- каждый раз, когда увольняется сотрудник организации, знающий пароль;
- когда безопасность системы поставлена под угрозу.

1.5. Доступ к бюджету привилегированного пользователя

Существует несколько способов доступа к бюджету привилегированного пользователя: регистрация под именем root, применение команды su и применение программы sudo.

Самый простой способ доступа к бюджету привилегированного пользователя - зарегистрироваться под именем root, но выход из собственного бюджета и регистрация в качестве привилегированного пользователя часто очень неудобны. Лучше использовать команду su.

Вызванная без аргументов, команда su пригласит вас ввести пароль привилегированного пользователя, а затем запустит оболочку shell с правами привилегированного пользователя. Привилегии этого интерпретатора команд остаются в силе до завершения его работы.

Команда su также может подставлять вместо root имена других пользователей. Зная чей-либо пароль, можно непосредственно войти в его бюджет, выполнив команду su и указав имя пользователя. Как и при входе в бюджет root, система запросит пароль пользователя и имя пользователя. В одних системах привилегированный пароль позволяет входить с помощью команд su или login в любой бюджет, а в других сначала нужно явно войти, воспользовавшись командой su, в бюджет root, а затем с помощью этой же команды перейти в другой бюджет.

Рекомендуем взять за правило при вводе команды использовать полное путевое имя /bin/su, а не просто su.

Во многих системах использовать команду su могут только члены группы wheel. Еще одно распространенное ограждение на пути использования команды su и бюджета root - ограничение их использования определенными терминалами.

Приложение!

ОС MCBC 3.0 - сложная ОС, поэтому если вы хотите, чтобы она работала корректно, необходимо внимательно выполнить настройку системы для начальной загрузки.

Разделы диска.

Linux поддерживает разделы, содержащие до 2^{32} секторов, т. е. размер раздела может достигать 2 Тбайт.

Поскольку в таблице разбиения отведено только 4 строки для задания разделов, число первичных разделов на диске с самого начала ограничено: их может быть не более 4. Когда стало ясно, что и 4-х разделов мало, были изобретены логические разделы. Для этого один из первичных разделов объявляется "расширенным" (тип раздела — 5, или F, или 85 в шестнадцатеричной системе), и в нем создаются "логические разделы". Расширенные разделы сами по себе не используются, они могут лишь хранить логические разделы. Первый сектор расширенного раздела хранит таблицу разделов с четырьмя входами: один используется для логического раздела, другой для еще одного расширенного раздела, а два не используются. Каждый расширенный раздел имеет свою таблицу разбиения, в которой, как и в первичном расширенном разделе, используются только две строки, задающие один

логический и один расширенный раздел. Таким образом, получается цепочка из таблиц разделов, где первая описывает три основных раздела, а каждая следующая — один логический раздел и положение следующей таблицы.

Число логических разделов в принципе не ограничено, потому что каждый логический раздел может содержать таблицу разделов и вложенные логические разделы. Однако реально ограничения все же существуют, например, Linux может работать не более чем с 15 разделами на SCSI-дисках и не более чем с 63-мя разделами на IDE-дисках.

Расширенный раздел как на физическом диске, так и в расширенном разделе вложенного расширенного раздела (предыдущего уровня) может быть только один: ни одна из существующих программ разбиения дисков (fdisk и ее усовершенствованные аналоги) не умеет создавать более одного расширенного раздела.

В Linux диск в целом (т. е. физический диск) доступен по имени устройства /dev/hda, /dev/hdb, /dev/sda, и т.п. Первичные разделы обозначаются дополнительной цифрой в имени устройства: /dev/hda1, /dev/hda2, /dev/hda3, /dev/hda4, а логические разделы в Linux доступны по именам /dev/hda5, /dev/hda6 ... (начиная с номера 5). Из сказанного выше должно быть ясно, почему могут быть пропущены имена /dev/hda3 и /dev/hda4 (третий и четвертый первичные разделы просто не были созданы) и сразу после /dev/hda2 вы увидите /dev/hda5 (логический раздел в расширенном разделе /dev/hda2), а далее нумерация идет последовательно.

2.1. Способы загрузки системы

Для работы пользователей необходимо загрузить ОС MCBC 3.0. Загрузка ОС MCBC 3.0 - это одна из задач администратора системы. Следует помнить, что на компьютерах многих пользователей, помимо ОС MCBC 3.0, может быть установлена, по крайней мере, еще одна ОС. Это означает, что при загрузке, компьютеру надо каким-то образом дать знать какую именно ОС следует загружать. Существует два основных способа загрузки ОС MCBC 3.0:

- загрузка с дискеты и
- загрузка с жесткого диска с помощью менеджеров загрузки.

2.1.1. Загрузка с дискеты

Для загрузки ОС MCBC 3.0 можно воспользоваться загрузочной дискетой. Такая загрузочная дискета содержит копию ядра ОС MCBC 3.0, которая указывает на корневую файловую систему ОС MCBC 3.0 на соответствующем разделе жесткого диска. Установочная программа ОС MCBC 3.0 предоставляет возможность создать такую загрузочную дискету при установке.

Вы должны создать загрузочную дискету в процессе установки, даже если не намерены применять ее для загрузки, т.к. в случае неприятностей использование этой дискеты может оказаться единственным способом загрузить ОС (см. документ ФЛИР.80001-01 32 01 "ОС MCBC 3.0. Руководство системного программиста").

После такой загрузки система предоставит вам минимальные возможности оболочки ash и некоторые другие утилиты, перечисленные в таблице 1, которых должно быть достаточно для восстановления системы.

Таблица 1

Утилита	Описание
fdisk	Работа с разделами жесткого диска
gzip/gunzi	Сжатие и распаковка файлов
Insmmod	Установка загружаемых модулей ядра
Is	Вывод списка файлов
mkdir	Создание каталога
mke2fs	Создание файловой системы ext2fs
mount	Подключение файловой системы
rm	Удаление файлов

rmmod	Выгрузка загружаемых модулей ядра
-------	-----------------------------------

В системах Intel с корневым разделом на жестком диске IDE можно воспользоваться для загрузки установочной дискетой ОС MCBC 3.0, введя при загрузке следующую команду:

MCBC single root=/dev/hda1

Подставьте вместо /dev/hda1 корректный раздел диска с корневой файловой системой ОС MCBC 3.0. Эта команда подключает корневой раздел, переводит систему в однопользовательский режим работы и опускает выполнение остальных действий по установке ОС MCBC 3.0.

Примечание. Описанный метод не работает, если корневой раздел находится на диске SCSI.

2.1.2. Загрузка с помощью менеджеров загрузки

Другой путь загрузки ОС MCBC 3.0 - использование менеджера загрузки.

1. В составе ОС MCBC 3.0 есть менеджер загрузки **LILO**. Эта программа модифицирует главную загрузочную запись жесткого диска и позволяет выбрать ОС для загрузки. Кроме LILO для загрузки Linux можно использовать и другие загрузчики.
2. Если у вас до установки Linux уже стояла ОС Windows NT, то вторым доступным для вас загрузчиком является **OS Loader от NT**. Преимущества: Во-первых, сохраняется вся старая конфигурация и, во-вторых, можно установить Linux на диск, который не может быть загрузочным в Linux, например, второй диск на втором контроллере (Secondary Slave).
3. **Другие** программы загрузки.

Использование менеджера загрузки имеет как преимущества, так и недостатки. При применении менеджера нет необходимости в загрузочной дискете; кроме того, вы можете выбрать ОС в процессе загрузки либо загрузить ОС по умолчанию. Из недостатков следует отметить, что менеджер загрузки вносит определенную сложность в процесс загрузки. Менеджер может быть модифицирован или даже переустановлен при добавлении, удалении или обновлении версии ОС. В связи с тем, что менеджер загрузки изменяет главную загрузочную запись, может оказаться, что всё, что вы можете сделать, - загрузиться с дискеты для форматирования жесткого диска. Следует оценить все преимущества и недостатки менеджера загрузки перед тем, как принять решение об использовании того или иного пути загрузки.

LILO представляет собой менеджер загрузки, являющийся частью ОС MCBC 3.0. Он может быть установлен:

1. в главной загрузочной записи,
2. на дискете,
3. в разделе диска с ОС MCBC 3.0 для запуска из менеджера загрузки другой ОС.

После установки LILO при загрузке вы можете выбрать одну из нескольких установленных на компьютере ОС. В зависимости от конфигурации LILO ожидает выбора в течение определенного времени и при его отсутствии загружает ОС по умолчанию.

Простейший путь установки LILO - установка ОС MCBC 3.0.

На следующем шаге нужно заставить LILO загружать ОС по выбору.

2.1.3. Настройка LILO

LILLO считывает конфигурационный файл /etc/lilo.conf и использует его для определения того, какие ОС установлены на компьютере и где расположена их загрузочная информация. Файл начинается с описания общих параметров работы LILO, за которыми идут разделы, описывающие ОС и содержащие специфическую информацию об их загрузке.

Ниже приведен пример двух разделов из конфигурационного файла LILO:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
lba32
```

```

vga=771
timeout=50
-----
other=/dev/hda1
label=win
table=/dev/hda
-----
image=/boot/vmlinuz-2.2.20-MCBC
label=MCBC
root=/dev/hda2
initrd=/boot/initrd-2.2.20-MCBC.img
read-only

```

Данный конфигурационный файл определяет, что lilo использует основную загрузочную запись (MBR) на устройстве /dev/hda.

В начале работы менеджер загрузки будет ожидать нажатия на клавишу [Shift] 4 секунды. После нажатия начнёт загружаться первый указанный образ ядра (/vmlinuz-2.2.20-MCBC). Если Вы нажали на Shift, то менеджер спросит, какой образ загружать. В случае, если Вы забыли возможные варианты загрузки, нажмите на [TAB] (или [?]) (если клавиатура соответствует стандарту US), и будет выведено меню. В файле lilo.conf может быть указано до 16-и образов.

Конфигурационный файл начинается с некоторого числа общих параметров (верхние 7 строк в примере (см. выше)), сопровождаемых описаниями параметров разных образов. Параметры в описании каждого образа имеют приоритет над параметрами в общей секции. Если Вы хотите чтобы какой-то образ загружался первым, то он должен быть первым описан в файле или необходимо ввести следующий параметр (т. е., используя метку из соответствующей строки label):

default=*название* (например default=win),

А строка timeout=50 означает, что через 5 секунд будет загружен первый образ.

В разделе WIN строка other определяет раздел /dev/hda1 как раздел дополнительной ОС, расположенный в разделе диска hda. Строка table определяет расположение таблицы разделов жесткого диска для раздела /dev/hda1. Не забывайте, что необходимо выполнить команду /sbin/lilo, если вы изменили ссылку на таблицу разбиения, задаваемую переменной table. Строка label описывает в каждом разделе имя ОС - как оно будет представлено в меню LILO.

Второй раздел содержит загрузочную информацию ОС MCBC 3.0. Строка image информирует LILO о расположении ядра ОС MCBC 3.0. Строка root определяет расположение корня файловой системы.

Если вы хотите грузить Windows непосредственно из LILO, то добавьте в /etc/lilo.conf еще одну секцию:

```

other = /boot/bootsect.dos
label = win,

```

где файл bootsect.dos берется из корневого каталога того диска, на котором стоит NT Loader.

После того, как вы откорректировали файл /etc/lilo.conf, необходимо *выполнить команду /etc/lilo*, чтобы изменения вступили в силу. Эта команда (которая в руководстве называется map-installer) устанавливает вторичный загрузчик системы, который будет активизирован во время следующей загрузки машины.

2.2. Начальная загрузка системы

На этапе начальной загрузки системы загружается и начинает выполняться ее ядро, затем запускается ряд инициализационных задач. После этого система готова к обслуживанию пользователей.

Начальная загрузка - это период особой уязвимости в жизни системы. Ошибки в конфигурационных файлах, сбой в работе оборудования, повреждения файловых систем могут помешать компьютеру нормально начать работу. Настройка режимов загрузки во многих случаях является одной из первых задач, которую приходится выполнять администратору в новой системе.

ОС MCBC 3.0 может загружаться либо в автоматическом, либо в ручном режиме. В автоматическом режиме система загружается самостоятельно, без какого-либо вмешательства извне. В ручном режиме до определенного момента система также загружается автоматически, но перед выполнением основных инициализационных командных файлов (сценариев) управление передается оператору. В это время система находится в так называемом "однопользовательском режиме". Большинство системных процессов не выполняется и вход других пользователей в систему невозможен.

В повседневной работе почти всегда применяется автоматическая загрузка. Типичная процедура загрузки выглядит так: пользователь включает питание и ждет, пока система выйдет в диалоговый режим. Системный администратор, однако, обязан не только понимать, как проходит процесс автоматической загрузки, но и знать, как загрузить систему вручную. Загружать систему вручную чаще всего приходится при возникновении проблем, прерывающих автоматический процесс загрузки. Это могут быть, например, повреждения файловой системы или ошибки в конфигурации сетевого интерфейса.

Обычно процесс начальной загрузки состоит из шести этапов:

- загрузка и инициализация ядра;
- распознавание и конфигурирование устройств;
- создание системных процессов;
- выполнение команд оператора (только при ручной загрузке);
- выполнение командных файлов запуска системы;
- переход в многопользовательский режим.

Администратор системы может управлять процессом загрузки, редактируя файлы запуска системы.

2.2.1. Инициализация ядра

Первый этап начальной загрузки заключается в считывании ядра ОС MCBC 3.0 в память для последующего выполнения.

В ОС MCBC 3.0 процесс загрузки ядра выполняется в два этапа. Сначала в память машины с диска считывается небольшая программа начальной загрузки, которая затем выполняет собственно загрузку ядра.

Ядро выполняет тестовые программы, позволяющие определить, сколько памяти имеется в наличии. Большинство ядер занимают объем памяти фиксированного размера и точно знают, сколько ее нужно зарезервировать для внутренних массивов и буферов ввода-вывода. В большинстве систем ядро выдает на консоль сообщение об общем объеме физической памяти и объеме памяти, не занятой ядром.

2.2.2. Конфигурация технических средств

Одна из первых задач, стоящих перед ядром - выявление компонентов технического обеспечения. Когда ядро начинает выполняться, оно пытается найти и инициализировать все устройства, о которых вы ему сообщили. Ядро ОС MCBC 3.0 выводит на консоль краткую информацию о каждом обнаруженном устройстве, в том числе тип жесткого диска, наличие манипулятора "мышь", подключение к сети и т.д. ОС MCBC 3.0 не хранит эту информацию от загрузки к загрузке, поэтому она вынуждена выполнять эту проверку при каждом запуске.

Сообщения, выдаваемые ядром, различаются в зависимости от конфигурации компьютера и версии ядра. Когда ОС MCBC 3.0 запускается впервые, ядро выдает на экран ряд сообщений (стартовую последовательность), которые можно не успеть просмотреть. ОС MCBC 3.0 содержит специальный файл /var/log/dmesg, в котором находятся все эти сообщения для последующего просмотра.

2.2.3. Создание системных процессов

После завершения базовой инициализации ядро ОС MCBC 3.0 создает в области памяти, выделенной для процедур пользователя, несколько процессов: swapper (процесс 0), init (процесс 1) и ragedaemon (процесс 2).

Из всех процессов только init является полноценным пользовательским процессом, а другие фактически представляют собой части ядра ОС, которые выглядят как процессы.

Создание процесса init осуществляется путем запуска программы init, находящейся в каталоге /sbin.

init представляет собой первую программу, которую выполняет ядро системы после загрузки. Соответственно идентификатор данного процесса (PID) равен 1. Именно этот процесс становится родительским для любого другого процесса, запущенного в ОС MCBC 3.0.

Идентификатор процесса (PID) представляет собой номер, используемый ОС для идентификации процесса. Многие команды ОС MCBC 3.0 применяют PID в качестве параметра для указания процесса, с которым следует произвести те или

иные действия.

После этого ядро больше не принимает участия в процедуре начальной загрузки системы. К этому моменту, однако, еще не создан ни один из процессов, управляющих базовыми операциями (например, входом пользователей в систему), и большинство сервисных программ не запущены. Обо всех этих задачах позаботится (в некоторых случаях косвенно) процесс init.

init запускает все процессы, необходимые ОС для выполнения ее обязанностей, например, использования манипулятора типа "мышь" (далее по тексту - "мышь"), основных функций наподобие ввода-вывода на терминал и сетевых операций. Программа init узнает о том, какие процессы ей следует запустить, из конфигурационных файлов, расположенных в каталоге /etc/red.

2.2.4. Действия оператора при ручной загрузке

Если систему нужно запустить в однопользовательском режиме, оператор указывает при запуске специальный флаг в командной строке, а ядро передает информацию процессу init. При загрузке в однопользовательском режиме init запускает на системной консоли интерпретатор команд и ждет, пока он завершит работу (после нажатия клавиш <Ctrl+D> или ввода команды exit), а затем продолжает выполнять процесс запуска. В однопользовательском режиме всегда используется sh (B-shell - интерпретатор команд Борна (Bourne), а не csh (интерпретатор C-shell)). Он выполняется с правами пользователя root, то есть привилегированного пользователя ОС MCBC 3.0.

В однопользовательском режиме оператор может выполнять команды почти так же, как и в многопользовательском. Однако обычно автоматически монтируется только раздел диска с корневым каталогом. Другие файловые системы оператор должен смонтировать вручную для того, чтобы использовать программы, находящиеся вне каталогов /bin, /sbin или /etc. Сервисные программы в однопользовательском режиме еще не запущены, поэтому команды, зависящие от некоторых обслуживающих процессов (например, mail), работать не будут.

Команда fsck, которая проверяет и восстанавливает поврежденные файловые системы, обычно выполняется в процессе автоматической загрузки. Если система запускается в однопользовательском режиме, команду fsck нужно "прогнать" вручную.

2.2.5. Выполнение командных файлов запуска

Следующий этап процесса начальной загрузки - выполнение командных файлов запуска. Эти файлы, по сути, представляют собой обычные командные файлы, и для их выполнения процесс init запускает процесс sh.

Ниже приведен **перечень задач**, которые часто **выполняются инициализационными командными файлами**:

- установка имени компьютера;
- установка часового пояса;
- проверка дисков командой fsck (только в автоматическом режиме);

- монтирование системных дисков;
- удаление файлов из каталога /tmp;
- конфигурирование сетевых интерфейсов;
- запуск сервисных процессов и сетевых служб;
- включение учета и контроля квот.

Большинство командных файлов запуска выводит на консоль подробную информацию о выполняемых ими задачах. Это может оказать существенную помощь при отладке или поиске причин зависания системы в процессе начальной загрузки.

2.2.6. Работа в многопользовательском режиме

После выполнения инициализационных командных файлов система полностью готова к работе, за единственным исключением: никто не может в нее войти. Для того чтобы с конфетного терминала можно было войти в систему терминал должен иметь свой процесс `getty`. После выполнения командных файлов запуска процесс `init` порождает эти процессы `getty`, завершая процесс загрузки. Необходимо помнить, что процесс `init` продолжает играть важную роль даже после завершения начальной загрузки.

В ОС MCBC 3.0 процесс `init` имеет всего два состояния: однопользовательское и многопользовательское.

2.3. Настройка режимов загрузки

Программа `init` запускает сценарии `/etc/rc.d/rc.sysinit`, все сценарии, определенные для уровня запуска по умолчанию и сценарий `/etc/rc.d/rc.local` (см. п. 2.3.3), которые находятся в каталоге `rc.d`.

Администратор системы может изменить процесс загрузки путем изменения сценариев `/etc/rc.d/rc.sysinit` и `/etc/rc.d/rc.local`.

2.3.1. Файл `rc.sysinit`

Файл `rc.sysinit` представляет собой сценарий, который `init` запускает первым в процессе загрузки, `rc.sysinit` устанавливает различные системные переменные и выполняет другую инициализацию системы. Здесь выполняются различные функции: установка системных переменных (типа имени машины), проверка файловой системы и запуск восстановления после сбоев, включение пользовательских квот и построение файловой системы `/proc`.

2.3.2. Сценарии перезапуска системы

Сценарий `rc` отвечает за перезапуск системы с другим уровнем запуска. Файлы ОС дополнительно разделены согласно уровням запуска по различным подкаталогам.

Уровень запуска (`run level`) определяет доступные типы служб, от однопользовательского режима (уровень 0) до полностью многозадачного, многопользовательского режима (уровень 3).

В таблице 1 перечислены доступные в ОС MCBC 3.0 уровни запуска.

Таблица 1

Уровень	Описание	Примечание
0	Остановка	(halt)
1	Однопользовательский	Текстовый режим без сети
2	Многопользовательский	Текстовый режим без сети
3	Полностью	Многозадачный, многопользовательский режим с сетью. Обычный режим.
4	Не используется	резерв
5	X11	устанавливается по умолчанию графический режим, нельзя войти под пользователем «root».

6	Перезагрузка	(reboot)
---	--------------	----------

Для того чтобы установить другой режим необходимо:

1. создать пользователя (чл. гр. администраторы),
2. отредактировать файл `inittab`, находящийся в директории `/etc/`. Для этого в строке этого файла

`id:3:initdefault:`

вместо цифры 3 нужно поставить номер требуемого режима.

В. 5. Файловая система ОС MCBC 3.0

Файловая система - это методы и структуры данных, которые используются операционной системой для хранения файлов на диске или его разделе.

Перед тем, как раздел или диск могут быть использованы в качестве файловой системы, он должен быть инициализирован, а требуемые данные перенесены на этот диск. Этот процесс называется созданием файловой системы.

Файловая система ОС MCBC 3.0 соответствует типу **EXT2** FS, обеспечивает поддержку длинных имен, символических связей, а также обеспечивает поддержку файловых систем ISO9660, FAT (MS-DOS), NTFS (Windows NT). В файловой системе ОС MCBC 3.0 также предусмотрена возможность представления имен файлов русскими буквами.

Файловые системы являются основой для хранения всех данных в ОС MCBC 3.0. Программы, библиотеки, системные и пользовательские файлы - все они располагаются в файловых системах.

При работе с ОС MCBC 3.0 видимое пользователем файловое пространство имеет древовидную структуру. На ее вершине находится корень. Каталоги и файлы образуют ответвления от этого корня. Самый верхний каталог известен как корневой каталог (root directory).

Изначально, файловая система состоит из одного каталога, называемого корневым (root). Внутри этого каталога есть еще каталоги, внутри которых могут быть файлы и даже другие каталоги. Может быть, а может и не быть ограничение на "глубину" файловой системы. Можно создать несколько уровней вложенности каталогов.

Пользователи видят дерево каталогов и файлов как единую цельную структуру. На самом деле различные каталоги этого дерева могут соответствовать различным разделам диска, находиться на других дисках и даже на других компьютерах. Раздел диска присоединяется к дереву в каталоге, называемом точкой монтирования (mount point). Точка монтирования и все расположенные ниже каталоги образуют файловую систему.

ОС MCBC 3.0 состоит из нескольких каталогов и множества файлов. В зависимости от выбора, сделанного в процессе установки, эти каталоги могут относиться к различным файловым системам.

После начальной установки файловая система ОС MCBC 3.0 может состоять например, из следующих частей: файловая система root, состоящая из каталогов `/bin` `/dev`, `/tmp`, `/etc`, `/lib`, `/proc`, `/sbin` и некоторых других; файловая система `/usr`, где хранятся различные программы и данные, не подлежащие изменению, файловая; система `/var`, где содержатся изменяемые файлы (такие как log файлы и др.) и файловая система `/home`, которая состоит из личных каталогов пользователей.

В каталоге `/bin` находятся выполняемые программы (точнее, их двоичные файлы). Они необходимы для работы системы. Многие команды ОС MCBC 3.0 на самом деле являются программами из этого каталога.

В каталоге `/dev` расположены особые файлы, называемые файлами устройств (device files). С их помощью осуществляется доступ ко всем физическим устройствам, установленным в системе.

Каталог `/tmp` используется для хранения временных файлов, создаваемых программами в процессе своей работы. Работая с программами, создающими много больших

временных файлов, лучше иметь отдельную файловую систему, нежели простой каталог корневой файловой системы. В случае записи в простой каталог корневой файловой системы при большом объеме данных файловая система может переполниться.

Каталог `/etc` содержит конфигурационные файлы системы. Здесь находится файл паролей `passwd`, а также список файловых систем, подключаемых при начальной загрузке `fstab`. В этом же каталоге хранятся сценарии загрузки (startup scripts), список узлов `hosts` с их IP-адресами и множество других данных о конфигурации.

В каталоге `/lib` содержатся разделяемые библиотеки, используемые многими программами во время своей работы. Применяя разделяемые библиотеки, хранящиеся в общедоступном месте, можно уменьшить размер программ за счет повторного использования одного и того же кода.

Каталог `/rpgos` является виртуальной файловой системой и используется для чтения из памяти информации о системе.

В каталоге `/sbin` хранятся системные двоичные файлы (большинство из них используется для нужд системного администрирования).

Каталог `/usr` и его подкаталоги необходимы для функционирования системы, так как содержат наиболее важные программы. Обычно это крупные программные пакеты, установленные в системе. Каталог `/usr` почти всегда является отдельной файловой системой.

В каталоге `/var` содержатся файлы, имеющие тенденцию изменять свой размер с течением времени. Обычно здесь располагаются различные системные журналы. Для данных преходящей природы, таких как недавно полученные или ждущие отправки почта и новости, используется каталог `/var/spool` и его подкаталоги.

Каталог `/home` предназначен для личных каталогов пользователей. Общепринято иметь здесь отдельную файловую систему, чтобы обеспечить пользователям достаточное пространство для размещения своих файлов. Если пользователей в системе много, возможно, придется разделить этот каталог на несколько файловых систем. Тогда, например, можно создать подкаталоги `/home/staff` и `/home/admin` для персонала и администрации, соответственно, установить каждый как самостоятельную файловую систему и уже в них создавать рабочие каталоги пользователей.

В личных каталогах каждого пользователя наряду с другими файлами имеются несколько конфигурационных файлов, которые для практических целей являются скрытыми. Они модифицируются редко. Файл становится скрытым, если поставлена точка в начале имени файла. Можно увидеть эти файлы, введя команду `ls -a`.

5.2. Типы файловых систем

ОС MCBC 3.0 поддерживает несколько типов файловых систем. Наиболее важные из них рассмотрены ниже.

Файловая система устанавливается, т.е. инициализируется, при помощи команды `mkfs(8)`. В действительности, существуют отдельные программы для каждого типа файловой системы. Команда `mkfs` только запускает требуемую программу в зависимости от типа устанавливаемой системы. Тип файловой системы указывается при помощи опции `-t fstype`.

Перед работой с файловой системой, она должна быть смонтирована. При этом ОС MCBC 3.0 выполняет некоторые действия, обеспечивающие функционирование монтируемой системы. Так как все файлы в ОС MCBC 3.0

принадлежат одной структуре каталогов, то эта операция обеспечивает работу с файловой системой, как с каталогом уже смонтированной.

Для подключения (монтирования) файловой системы к дереву каталогов ОС MCBC 3.0 необходимо иметь место на жестком диске, компакт-диск или гибкий магнитный диск, который необходимо подключить. Следует убедиться также, что каталог, к которому необходимо подключить файловую систему (точка подключения), действительно существует.

Предположим, что требуется подключить компакт-диск в дисковод `/dev/sr0` к точке подключения `/mnt`. Каталог `/mnt` должен уже существовать, иначе подключение окончится неудачей. После подключения к этому каталогу все файлы и подкаталоги файловой системы появятся в каталоге `/mnt`. В противном случае каталог `/mnt` будет пустым.

Для того чтобы узнать, какой файловой системе принадлежит текущий каталог, воспользуйтесь командой `df`. Вы увидите файловую систему и объем свободного пространства.

В ОС МСВС 3.0 для подключения файловых систем используется команда `mount`. Синтаксис команды `mount`:

`mount device mountpoint`

где `device` обозначает физическое устройство, которое необходимо подключить, а `mountpoint` - точку подключения.

В целях системной безопасности использовать команду `mount` могут только привилегированные пользователи.

Кроме аргументов, указанных выше, команда `mount` может иметь в командной строке еще несколько аргументов, приведенных в таблице 9. Если необходимый аргумент не указан, `mount` попытается определить его по файлу `/etc/fstab`.

Приведем несколько распространенных форм команды `mount`:

`mount /dev/hdb3/mnt` подключает раздел жесткого диска `/dev/hdb3` к каталогу `/mnt`;

`mount -r -t iso9660 /dev/sr0/mnt` подключает привод компакт-дисков SCSI только для чтения с форматом файлов (ISO 9660) к каталогу `/mnt`;

`mount -vaf nfs` подключает все файловые системы NFS, перечисленные в файле `/etc/fstab`.

Таблица 9

Аргумент	Описание
<code>-f</code>	Имитирует подключение файловой системы. Выполняются все действия, кроме системного вызова для настоящего подключения
<code>-V</code>	Подробный отчет, <code>mount</code> предоставляет дополнительную информацию о своих действиях
<code>-w</code>	Подключает файловую систему с доступом для чтения и записи
<code>-r</code>	Подключает файловую систему с доступом только для чтения
<code>-n</code>	Выполняет подключение без записи в файл <code>/etc/mtab</code>
<code>-t type</code>	Указывает тип подключаемой файловой системы. Допустимыми являются типы <code>minix</code> , <code>ext2</code> , <code>xiafs</code> , <code>insdos</code> , <code>hpfs</code> , <code>proc</code> , <code>nfs</code> , <code>umsdos</code> , <code>sysv</code> и <code>iso9660</code> (тип по умолчанию)
<code>-a</code>	Указывает <code>mount</code> подключить все файловые системы, перечисленные в <code>/etc/fstab</code>
<code>-o list_of_options</code>	Указывает <code>mount</code> применить список опций к подключаемой файловой системе. Опции в списке перечислены через запятую. За полным списком возможных опций обратитесь к руководству по <code>mount</code>

Если правильно подключить файловую систему не удастся, воспользуйтесь командой `mount -vt device mountpoint`, чтобы узнать, что именно пытается сделать команда `mount`. В этом случае команда `mount` выполнит все действия кроме подключения и даст о них подробный отчет.

5.4.2. Подключение файловых систем при загрузке

Если список используемых файловых систем изменяется редко (а это бывает в большинстве случаев), то удобно указать ОС МСВС 3.0 подключать их сразу же при загрузке и отключать при завершении работы. Эти файловые системы перечисляются в специальном конфигурационном файле `/etc/fstab`.

Файловые системы перечисляются в файле `/etc/fstab` по одной файловой системе в строке. Поля в строках разделяются пробелами или символами табуляции.

Файловая система – Подключаемое блочное устройство или удаленная файловая система

Точка подключения – Место подключения файловой системы. Чтобы сделать систему невидимой в дереве каталогов (например, для файлов подкачки), используйте слово `none`

Тип – Указывает тип подключаемой файловой системы. В настоящее время поддерживаются системы таких типов:

- minix, локальная файловая система с именами файлов длиной до 14 или 30 символов;
- ext, локальная файловая система с длинными именами файлов (была заменена файловой системой ext2 и более не должна использоваться);
- ext2, локальная файловая система с длинными именами файлов и другими возможностями;
- xiafs, локальная файловая система;
- msdos, локальная файловая система для разделов MS-DOS;
- hpfs, локальная файловая система для разделов OS/2 (OS/2 High Performance File System);
- iso9660, локальная файловая система, используемая с компакт-диском;
- nfs, файловая система для подключения разделов удаленных систем;
- swap, раздел или файл подкачки;
- umsdos, файловая система UMSDOS;
- sysv, файловая система System V
- ntfs, файловая система NTFS.

Опции – Список разделенных запятыми опций для подключаемой подключения файловой системы должен содержать, по крайней мере, тип подключения. За более подробной информацией обращайтесь к странице руководства man команды mount.

Периодичность резервного копирования – Указывает, как часто следует выполнять резервное копирование с помощью команды dump. Если это поле отсутствует, dump считает, что файловая система не нуждается в резервном копировании

Номер прохода – Задаёт порядок проверки целостности файловых систем при загрузке с помощью команды fsck. Для корневой файловой системы следует указывать значение 1, для остальных - 2. Если значение не указано, целостность файловой системы при загрузке проверяться не будет

Рекомендуется подключать файловые системы во время загрузки через /etc/fstab вместо mount. Помните, что команда mount доступна только привилегированным пользователям.

Пример файла fstab:

# device	directory	type	options
/dev/hda1	/	ext2	defaults
/dev/hda2	/usr	ext2	defaults
/dev/hda3	none	swap	sw
/dev/sda1	/dos	msdos	defaults
/proc	/proc	proc	none

Прежде всего, обратите внимание, что комментарии в этом файле начинаются с символа "#". В примере перечислено несколько файловых систем. Сначала идут две обычные файловые системы ОС МСВС 3.0 - дисковые разделы /dev/hda1 и /dev/hda2. Они имеют тип ext2 и подключаются к корневому каталогу / и к каталогу /usr.

Слово **defaults** в поле options указывает, что при подключении файловой системы следует применить набор опций по умолчанию, а именно - файловую систему следует подключить с разрешенным доступом для чтения и записи; она должна рассматриваться как отдельное блочное устройство; весь файловый ввод-вывод должен выполняться асинхронно; разрешено выполнение программных файлов; файловая система может подключаться с помощью команды mount -a; биты UID и GID файлов в этой файловой системе интерпретируются; обычным пользователям не разрешено подключать эту файловую систему. Как видите, гораздо проще написать здесь defaults.

Однако при выборе опции defaults у Вашей монтируемой файловой системы будет отсутствовать поддержка русского языка (ОС не будет распознавать файлы и каталоги на русском языке), для того чтобы всё отображалось корректно необходимо в поле опции записать:

/dev/hdaxxx	/mnt/disk/xxx	vfat	codepage=866,ioccharset=koi8-r	0	0
		ntfs	user,umask=111, ioccharset=koi8-r	0	0

В примере, раздел подкачки /dev/hda3 используется ядром ОС MCBC 3.0 для организации виртуальной памяти (см. документ ФЛИР.80001-01 32 01). Он должен присутствовать в файле /etc/fstab, чтобы система знала, где он находится. Чтобы он не появлялся в дереве каталогов, точка подключения указана как none. Кроме того, разделы подкачки подключаются с опцией "sw".

Виртуальная файловая система /proc указывает на информационное пространство процессов в памяти. Соответствующий физический раздел для нее отсутствует.

Для получения полной информации о допустимых в файле /etc/fstab опциях см. пользовательские инструкции man для fstab.

5.5. Отключение файловой системы

Для отключения файловых систем используется команда umount. Отключение может понадобиться для проверки и восстановления файловой системы с помощью команды fsck. Удаленные файловые системы отключаются в случае неполадок в сети.

Команда umount имеет три основные формы:

umount device: mountpoint

umount -a

umount -t fstype

где device означает физическое устройство, которое необходимо отключить, а mountpoint - имя каталога точки подключения (указывайте только device или mountpoint). У команды umount всего два параметра. Параметр "-a" отключает все файловые системы, а параметр "-t fstype" - только файловые системы указанного типа.

Команда umount не отключает файловые системы, если они используются в текущий момент.

Назначение службы Samba

Служба Session Message Block (SMB), известная также как Samba, предоставляет следующие возможности:

- разделение файловых систем ОС MCBC 3.0 операционными системами Windows 9X, Windows NT/2000/XP;

- совместное использование принтеров, подключенных к системе ОС MCBC 3.0, операционными системами Windows 9X, Windows NT/2000/XP;

Samba представляет собой протокол, используемый Microsoft для разделения файлов и служб печати.

Состав программ службы Samba

ОС MCBC 3.0 обеспечивает поддержку протокола Samba. Протокол Samba реализуется комплексом программ, который состоит из нескольких компонентов:

- сервисная служба smbd;
- сервисная служба nmbd;
- сервисная служба smbclient;
- утилита testparm;
- утилита smbstatus.

Сервисная служба smbd обеспечивает работу службы печати и разделения файлов для клиентов Samba, таких как Windows for Workgroups, Windows NT и LanManager. Конфигурационные параметры сервисной службы smbd описываются в файле smb.conf.

Сервисная служба nmbd обеспечивает работу службы имен NetBIOS, а также может использоваться для запроса других сервисных служб имен.

Сервисная служба smbclient реализует простейший FTP-подобный клиент, который может использоваться для доступа к другим серверам Samba и для печати на принтерах, подключенных к серверам Samba.

Утилита testparm позволяет протестировать конфигурационный файл smb.conf.

Утилита smbstatus сообщает, кто в настоящее время пользуется сервером smbd.