UNIX System V

### 1. Введение

### 1.1. Цель данного руководства

Руководство по администрированию систем AT&T UNIX System V Release 4.0 Version 1.0, разработана для оказания помощи в администрировании системы, основанной на системе 386 и выполняемой в среде UNIX System V Release 4.0. Предполагается, что вы уже знакомы с другими системами UNIX. Задачи администрирования системы показаны с помощью команд sheel и меню sysadm, представленных в пакете прикладных программ "Operations, Administration and Maintenance" (OA&M). Пакет OA&M не является частью основной системы; вам следует инсталировать его, чтобы обеспечить доступ к интерфейсу.

Данное руководство необходимо использовать после того, как программные средства будут установлены в соответствии с указаниями Руководства по инсталяции систем AT&T UNIX System V Release 4.0 Version 1.0. Инструкции по инсталяции и инициализации расширенных прикладных программ включены в документацию, сопровождающую расширенный продукт.

#### 1.1.1. Что такое администрирование системы в процессоре 386?

Администратор системы включает задачи, обычно выполняемые в вычислительной системе: создание резервных копий и восстановление файлов, добавление и удаление пользователей, управление сетью, добавление и удаление аппаратных средств и т.д.

Для выполнения этих задач в системе имеется два интерфейса: команды shell, являющиеся непосредственным пользовательским интерфейсом с административными функциями среды системы UNIX, и меню sysadm, обеспечивающие интерфейс меню с теми же задачами с помощью команды sysadm. Опытные администраторы системы должны быть знакомы с обоими интерфейсами.

#### 1.1.2. Обслуживание, администрирование и сопровождение

Если вы инсталировали пакет прикладных программ "Operations, Administration and Maintenance" (OA&M), sysadm обращается к меню в формате окон, используя интерфейс форм, меню и языка (FMLI).

Примечание. Рекомендуется инсталировать пакет OA&M до инсталяции Экранной командной оболочки Framed Acces Command Environment (FACE) ("Файловый доступ к командной среде"). Если вы инсталируете пакет FACE первым, вы получите сообщения об ошибках при загрузке первого гибкого диска. Инсталяция завершится частичным сбоем установки.

При инсталяции пакета OA&M вам предлагается выбор между основным пакетом или пакетом расширений OA&M. Пакет расширений OA&M - выбор по умолчанию - дает более широкий диапазон услуг по созданию резервных копий и восстановлению файлов, плюс другие услуги - создание группы пользователей, сопровождение программ, сопровождение файлов и т.д.

#### *1.1.2.1. Инсталирование основного пакета и пакетов расширений OA&M*

При первоначальной инсталяции можно установить основной пакет OA&M или вместе и основной пакет и пакеты расширений OA&M. Если вы инсталировали оба пакета OA&M - и основной, и пакет расширений - и хотите удалить расширение для сохранения пространства, следует удалить и основной и пакеты расширений и заново инсталировать основной пакет OA&M. Расширение меню помечается #oam# на поврежденных строках меню в файлах меню OA&M.

Примечание. Файлы ".menu" object\_gen позволяют добавить исходные тексты FML (Form., Menu., или Text. файлы) с помощью пакетов расширений в структуре каталога расширений. Эти добавления можно удалять (в соответствии с правилами в Руководстве разработчиков пакетов расширений), когда удаляется пакет расширений (модульность). Меню усовершенствуются автоматически; прототип пакета расширений содержит файлы ".mi", определенные как OAMmif. Документы о действиях OAMmif, представленные в OA&M, корректируют главные меню, объявленные в пакете расширений как именующие специфическое расширение. Это вызывает переадресацию на местоположение специфических каталогов расширений, где должны постоянно находиться файлы FML.

Например, расширение Small Computer Systems Interfase (SCSI) (Интерфейса систем малых ЭВМ) добавляет выбор типов меню "buses" и "devices". Специфический выбор SCSI (строки) в меню "buses" и в меню "devices" определен в поле 4 с помощью метки #scsi#. #scsi# интерпретируется с помощью object\_gen и преобразуется в /usr/sadm/sysadm/add-ons/scsi (в противоположность /usr/sadm/sysadm/menu/\*) - местоположение специфического расширения SCSI. В дальнейшем другие пакеты расширений могут привносить расширения в те же элементы главного меню (они будут содержать соответствующее перемещение #add-on\_name#).

### 2. Использование файловых систем

В этом разделе описывается, как создавать, монтировать и демонтировать типы файловых систем s5 и ufs. Для этого необходимо сначала отформатировать гибкие и жесткие диски. Если используемые диски не были отформатированы, обратитесь к разделу 9 "Форматирование запоминающих устройств" и "Управление запоминающими устройствами".

Обратитесь к этим задачам из опции file\_system в меню UNIX System V Release 4.0 Version 1.0.

### 2.1. Создание файловой системы

2.1.1. Использование меню OA&M для создания файловой системы   
2.1.2. Использование команды mkfs для создания файловой системы

* 2.1.2.1. Создание файловой системы s5   
  2.1.2.2. Создание файловой системы ufs   
  2.1.2.3. Выбор логического размера блока.

Создание рабочей файловой системы проводится в несколько этапов:

1. Форматирование гибких дисков.
2. Создание файловой системы с использованием меню OA&M или команды mkfs.
3. Установка файловой системы.
4. Демонтирование файловой системы при отсутствии обращения к ней.

#### 2.1.1. Использование меню OA&M для создания файловой системы

Используйте следующие меню для создания файловой системы:

1. Выберите make из меню "Управление файловой системой" (Manage File System).

Система отображает экран "Создание файловой системы" (Create a File System) (make):

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Create a File System (make) |

------------------------------------------------

Device that will contain the file system: diskette1

File system type: a5

Lable for the file system:

Once created, should the new file system be mounted?yes

File system name when mounted: /install

Введите информацию файловой системы и нажмите SAVE.

Система отображает один из следующих экранов:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Create An a5 File System (make) |

------------------------------------------------

Name of prototype file:

Block Size in Bytes: 1024

Number of Blocks:

Number of Files:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Create a File System (make) |

------------------------------------------------

Number of Blocks in the File System:

Block Size IN bYTES: 8192

Fragment Size in bytes: 1024

1. Введите информацию и нажмите SAVE.

Система отображает команду вставить дискету в дисковод.

1. Нажмите SAVE, когда вставите дискету в дисковод.

Затем система создает файловую систему и пытается смонтировать ее.

#### 2.1.2. Использование команды mkfs для создания файловой системы

В данном подразделе сначала обсуждается общий формат команды mkfs, а затем ее специфическое применение для создания файловой системы s5 или ufs. Формат команды mkfs:

mkfs [-F filetype] [-V] [-m] [current\_options]

[-o specific\_options] special operands

где filetype - тип файловой системы - либо s5, либо ufs; -V - отображает полностью командную строку, включая информацию файла ufstab; -m - возвращает командную строку, используемую для создания существующей файловой системы. Этот параметр позволяет пользователю видеть атрибуты, из которых состоит файловая система; current\_options - параметры, поддерживаемые s5; specific\_options - атрибуты файловой системы должны быть введены в файл /etc/vfstab. special - имя элемента vfstab, содержащее атрибуты особой файловой системы; special\_operands - операнды, специфичные для типа создаваемой файловой системы.

#### *2.1.2.1. Создание файловой системы s5*

Введите следующие команды, чтобы создать новую файловую систему s5 или преобразовать старую с новым логическим размером блока:

1. Если новая файловая система будет создаваться в разделе диска, где постоянно находится старая файловая система, создайте резервную копию старой системы. Для создания резервных копий систем с одним или несколькими жесткими дисками можно использовать команду cpio(1).
2. Если новая файловая система будет создаваться из старой, выполните команду labelit, которая сообщает и имя уже смонтированной файловой системы и физическое имя тома старой файловой системы. Эти метки уничтожаются при создании новой файловой системы.

Вам следует специфицировать тип файловой системы, когда вы используете команду labelit. Например, если используется устройство f0q15d, файловой системой будет memo, а именем тома memo 2.0.

Введите:

labelit -F s5/dev/dsk/f0q15d memo memo 2.0

Если новую файловую систему необходимо создать из старой, и новая файловая система будет иметь больший логический размер блока, тогда благодаря фрагментации новая файловая система выделит больше блоков диска для хранения данных, чем старая система. Используйте команду fsba (1M), чтобы узнать требования области памяти старой файловой системы с новым размером блока. Используйте эту информацию также для того, чтобы убедиться, что часть диска, которая будет использоваться для новой файловой системы, достаточно большая. Используйте команду prtvtoc (1M), чтобы узнать размер частей текущего диска.

1. Используйте одну из следующих команд:
2. mkfs [-F s5] [-b blocksize] special blocks[:inodes]
3. [gap blocks/cyl]

или

mkfs [-F s5] [-b blocksize] special prototype

[gap blocks/cyl]

где blocksize - логический размер блока файловой системы.

Значение по умолчанию - 1024 байта. Система s5 поддерживает также 512 - байтовые и 2048 - байтовые блоки; special - вход в файле vfstab, содержащий атрибуты файловой системы; blocks - количество 512 - байтовых блоков, которые займет файловая система. Значением по умолчанию для inodes является один индексный дескриптор файла для каждых четырех логических блоков памяти; gap blocks/cyl - промежуток между записями и количество блоков в одном цилиндре. Эти цифры зависят от размера блока и размеров жесткого диска; prototype - имя файла, который может включать: количество блоков, необходимых для файловой системы, каталог и структуру файла для файловой системы, а также команды считывания содержимого существующих файлов в файловую систему.

Обратите внимание, что файловой системе не дается имя ни в одном из форматов команды mkfs; оно идентифицируется по имени файла специального устройства, в котором его имя находится. Файл специального устройства, обычно расположенный в каталоге /dev, привязан к идентифицирующему контроллеру и соответственно к типу и номеру физического устройства.

В первом формате единственной другой информацией, которая должна быть размещена на командной строке mkfs, является количество 512 - байтовых блоков, которые займет файловая система. Второй формат позволяет включать эту информацию в макетный файл, который может также определять каталог и файловую структуру для новой файловой системы; он также позволяет считывать содержание файлов из существующей файловой системы.

Оба формата позволяют определять информацию о промежутках между записями и блоках в одном цилиндре. Если такая информация не дается на командной строке, тогда используются значения по умолчанию.

Рекомендации зависят от логического размера блоков файловой системы. Опция -b команды mkfs позволяет определить логический размер блоков, используемых в файловой системе. По умолчанию логический размер блоков файловой системы составляет 1024 байта. С помощью параметра -b можно определить логический размер блоков от 1024 до 2048 байтов. Рекомендуемые значения отличаются от значений по умолчанию, используемых командами. Оптимальное значение зависит от типа прикладной задачи (например, у интенсивных прикладных программ считывания другие характеристики нежели у интенсивных прикладных программ записи), от используемого дискового контроллера и чередования дисков.

В первом формате mkfs, даже если и требуется число блоков в файле, количество индексных дескрипторов файла может быть опущено. Если количество индексных дескрипторов файла опускается, команда использует значение по умолчанию одного индексного дескриптора файла для каждых четырех логических блоков памяти.

Если вы используете первый формат mkfs, тогда файловая система создается с единственным каталогом. Если вы используете макетный файл, как указано выше, он может включать информацию, которая заставляет команду строить и инициализировать каталог и файловую структуру для файловой системы.

1. Прогоните команду labelit, чтобы восстановить файловую систему и имена томов.
2. Загрузите новую файловую систему - например, восстановите файловую систему из резервной копии или, если у вашей системы два жестких диска, выполните команду cpio (1M) из смонтированной файловой системы. (Команды volcopy (1M) и dd (1M) копируют образ файловой системы; они не могут преобразовать логический размер блоков).

#### *2.1.2.2. Создание файловой системы ufs*

При построении файловой системы ufs команда mkfs создает файловую систему с корневым каталогом и с каталогом lost+found. Число индексных дескрипторов файла высчитывается как функция размера файловой системы.

Введите следующие команды, чтобы создать новую файловую систему ufs или преобразовать старую в новый логический размер блока:

1. Если новая файловая система будет создаваться на части диска, где находится старая файловая система, создайте резервную копию старой системы.
2. Если новая файловая система будет создаваться из старой, выполните команду labelit, которая сообщает и имя смонтированной файловой системы и физическое имя тома старой файловой системы. Эти метки уничтожаются, когда создается новая файловая система.

Необходимо определить тип файловой системы при использовании labelit. Например, если у вас устройство f0q15d, файловой системой будет memo, а именем тома memo 2.0. Введите:

labelit -F ufs/dev/dsk/f0q15d memo memo 2.0

1. Используйте одну из следующих команд:
2. mkfs -F ufs [-o] [arguments special size

или

mkfs -F ufs [-o] [arguments special prototype

где special - вход в файле vfstab, содержащий атрибуты файловой системы; size - количество секторов в файловой системе; arguments - необязательные параметры - это список параметров, отделяемых запятыми и позволяющих настраивать файловую систему. Ниже приводится список наиболее важных параметров:

* + nsect - число секторов на одну дорожку на диске. Значение по умолчанию 18. Если вы выдаете команду prtvtoc -p для дискового запоминающего устройства, число секторов выводится как "# sectors";   
    ntrack - число дорожек на один цилиндр на диске. Значение по умолчанию 0. Команда prtvtoc -p выводится как "# heads";   
    bsize - первоначальный размер блоков для файлов файловой системы, выбираемый из 4096 (по умолчанию) или 8192;   
    fragsize - наименьшее пространство на диске, которое выделяется для файла. Значение должно быть степенью числа 2, выбранное из диапазона от 512 до 8192. Значение по умолчанию 1024;   
    cgsize - количество дисковых цилиндров на одну группу цилиндров. Это число должно быть в диапазоне от 1 до 32. Значение по умолчанию 16;   
    free - минимальный процент допустимого свободного дискового пространства. Если объем файловой системы достигает этого порога, вы должны быть привилегированным пользователем, чтобы выделить дисковые блоки. Значение по умолчанию 10.

Если списку параметров предшествует -o, тогда необходимо специфицировать только желаемые параметры, но каждый параметр должен быть явно маркирован. Иначе, параметры исследуются слева направо, где первым параметром считается nsect, вторым - ntrack и т.д.

Следующие две команды схожи по функциям:

mkfs -F ufs -o bsize=4096,nsect=18,

ntrack=9 /dev/rdsk/1s2 35340

mkfs -F ufs /dev/rdsk/1s2 35340 32 16 4096

prototype - имя файла, которое может включать: количество блоков, необходимое для файловой системы, каталог и файловая структура, а также команды считывания содержания соответствующих файлов в файловую систему.

1. Прогоните команду labelit, чтобы восстановить файловую систему и имена томов.
2. Заполните новую файловую систему - например, восстановите из резервной копии файловой системы или, если в вашей системе два жестких диска, выполните команду cpio (1M) из смонтированной файловой системы. (Команды volcopy (1M) и dd (1M) копируют образ файловой системы; они не могут преобразовывать логический размер блока).

#### *2.1.2.3. Выбор логического размера блока*

Логический размер блока - это размер порций, которые ядро системы использует для считывания или записи файлов. Логический размер блока обычно отличается от физического размера блока - размера наименьшей порции, которую дисковый контроллер может считать или записать; как правило, 1024 байта.

Администратор, использующий команду mkfs для создания файловой системы, может определить логический размер блоков файловой системы. По умолчанию логический размер блоков составляет 1024 байта (1K) в системе s5 и 4096 байтов (4K) в системе ufs. Файловые системы root и usr поставляются как 1K файловые системы. Кроме 1К файловых систем, система s5 также поддерживает 2048 байтовые (2К) файловые системы.

Чтобы разумно выбрать логический размер блоков, необходимо учесть производительность и объем пространства. Для большинства систем ufs 8K файловая система с размером фрагментов 1К имеет наилучшую производительность, тогда как для большинства s5 систем 1К файловая система самая производительная. Для специальных прикладных программ, прогоняемых в системе s5 (как например, для s5 служебных файловых процессоров), которые используют большое количество выполнимых файлов или файлов данных, лучшим выбором может быть 2К файловая система.

### 2.2. Монтирование файловой системы

2.2.1. Использование меню OA&M для монтирования файловой системы   
2.2.2. Использование mount для монтирования файловой системы

При создании файловой системы единственным именем на командной строке (в отличие от имени файла макета, если вы использовали этот параметр) является имя специального файла устройства. Поскольку обычно к файловой системе UNIX System V обращаются по имени каталога наивысшего уровня в ее иерархии, следующим этапом после создания файловой системы должно быть соединение имени файловой системы с именем каталога.

Это выполняется путем "монтирования" файловой системы. При этом используются меню OA&M или команда mount (1M). Этот этап обязателен.

Выполнение команды mount требует попарного соединения смонтированного дискового устройства и вмонтированного каталога. Система UNIX обеспечивается информацией о типе файловой системы, о параметрах, используемых для монтирования и о времени, необходимом для монтажа. Эта информация хранится в файле /etc/mnttab.

Например, команда

mount -F s5 /dev/dsk/1s2/usr

просит систему смонтировать /dev/dsk/1s2 как s5 файловую систему, которая начинается в каталоге /usr.

Если вы попытаетесь заменить каталоги (при помощи команды cd) на каталог в файловой системе usr до выхода команды mount, то команда cd не выполнится. Пока не завершится команда mount, система не будет знать ни о каких каталогах в файловой системе usr.

Можно определить файловые системы, содержащиеся на гибком диске, и использовать их либо для хранения, либо для прямого доступа. Однако, общим для пользователей считается копирование файловой системы в каталог на жестком диске. Для этого файловая система должна быть сначала смонтирована.

Команда labelit также помогает осуществить связь между специальным файлом устройства и смонтированным именем файловой системы. Она записывает каталог наивысшего уровня файловой системы (т.е. ее имя) в поле в системном блоке тома.

*Примечание.* Когда команда labelit используется для монтируемых файловых систем, таких как, например, на гибких дисках, одним из фактических параметров командной строки может быть идентификационный номер тома. Этот номер тоже хранится в поле в системном блоке тома, хотя общепринятым считается просто записать имя на этикетке дискеты, где находится файловая система.

#### 2.2.1. Использование меню OA&M для монтирования файловой системы

Обратитесь к этим задачам из необязательного параметра file\_systems в меню System Administration. Используйте следующие меню для монтирования файловой системы:

1. Выберите mount из меню Manage File Systems.

Система отображает экран Mount a File System:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Mount a File System |

------------------------------------------------

Device that contains the file system: diskette 1

File system name when mounted:

1. Нажмите CHOICES для выбора правильных парметров, затем нажмите SAVE.

#### 2.2.2. Использование mount для монтирования файловой системы

Общая форма команды mount: mount.

Например, команда

mount -F s5/dev/dsk/1s4/home

просит систему смонтировать /dev/dsk/1s4 как файловую систему типа s5, которая начинается в каталоге /home.

Обычно пользователям удобнее обращаться к файлам с жесткого диска. Для этого часть используемых файлов копируется с лент или с гибких дисков на жесткий диск.

Введите следующие команды, чтобы смонтировать систему и скопировать содержание файлов на жесткий диск:

1. Создайте два каталога на жестком диске: один будет служить связующим звеном между гибким диском и жестким (точка монтирования), а другой будет корневым каталогом монтируемой файловой системы.

Например, команды на этапах, указанных ниже, используют /mnt как каталог точек монтирования и /myfs как корневой каталог.

1. Смонтируйте дискету. Например:
2. mount -F s5 -oro/dev/diskette /mnt
3. Перейдите к каталогу монтирования, в данном случае
4. cd/mnt
5. Скопируйте содержание файловой системы в каталог /myfs, используя
6. find . -print | cpio -pdm/myfs

Объяснение используемых параметров дано в find(1) и cpio(1).

### 2.3. Демонтирование файловой системы

2.3.1. Демонтирование файловой системы с помощью меню   
2.3.2. Использование команды umount для демонтирования файловой системы

Демонтирование файловой системы логически отсоединяет файловую систему от связанного с ней дискового устройства. У пользователей больше нет доступа к каталогам и файлам файловой системы.

Демонтирование часто является первым этапом перед использованием других команд, применяемых в файловых системах. Например, fsck (1M), которая проверяет и исправляет файловую систему, действует в демонтированных файловых системах. Демонтирование - это также важная часть процесса закрытия системы.

До начала процесса демонтирования все файлы в демонтируемой файловой системе должны быть закрыты и необходимо перейти к каталогу, которого нет в файловой системе. Например, если текущий каталог находится в файловой системе, которую вы хотите демонтировать, вам следует выйти из этого каталога до начала процесса демонтирования. Иначе вы получите следующее сообщение:

/etc/umount:device busy

#### 2.3.1. Демонтирование файловой системы с помощью меню

Обратитесь к этим задачам, используя параметр file\_system в меню System Administration.

Используйте следующие меню для демонтирования файловой системы:

1. Выберите unmount из меню Managing File Systems.

Система отображает экран Unmount a File System:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| Unmount a File System |

------------------------------------------------

Mountpoint of Device to be unmounted:

1. Нажмите CHOICES, чтобы отобразился список имеющихся файловых систем. Используйте клавиши со стрелками для перемещения курсора в соответствующую файловую систему и нажмите SAVE, чтобы выбрать ее.
2. Нажмите SAVE, чтобы начать процесс демонтирования. Система отображает экран верификации:
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. | Unmount a File System |
5. ------------------------------------------------
6. unmount your selection ? yes
7. Нажмите SAVE, чтобы продолжить работу. Система отображает экран верификации после того, как будет демонтирована файловая система.

#### 2.3.2. Использование команды umount для демонтирования файловой системы

Команда для демонтирования файловой системы требует только имя специального устройства или точки демонтирования.

Чтобы демонтировать файловую систему, введите следующую команду:

umount device\_name

Например, если файловая система была cмонтирована с дискеты, то команда

umount /dev/diskette

освобождает дисковое запоминающее устройство.

#### 2.4.1. Повреждение файловой системы

Файловая система может быть повреждена несколькими способами. Три наиболее общих:

* неправильное закрытие или запуск системы;
* удаление носителей данных до демонтирования файловой системы;
* выход из строя аппаратных средств.

Можно сохранить надежность файловых систем, выполняя следующие правила:

* всегда используйте процедуру shutdown перед выключением компьютера. Процедура закрытия демонтирует все файловые системы;
* всегда демонтируйте смонтированный гибкий диск системы UNIX перед его удалением;
* никогда не удаляйте гибкий диск, пока дисковод работает.

#### 2.4.2. Целостность файловой системы

У компьютера есть несколько встроенных признаков надежности. Ниже приводится краткий список этих признаков:

* когда файл записывается на жесткий диск, его индексный дескриптор и блоки записываются в том порядке, который гарантирует максимальную надежность. Это известно как упорядоченные записи;
* системные буферы периодически записываются на жесткий диск. Это известно как автоматическая модификация;
* если файловая система повреждается, необходимо прогнать программу fsck, чтобы очистить файловую систему перед ее монтированием.

#### 2.4.3. Программа fsck

Контролирующая программа (fsck) файловой системы - это интерактивная контрольно-исправительная программа файловой системы. Программа fsck использует информацию, находящуюся в самой файловой системе, для проверки целостности. Если обнаружено нарушение целостности, отображается сообщение, описывающее нарушение целостности. Рекомендуется выбрать параметр -y для программы fsck, чтобы эта програма автоматически исправила выявленные нарушения целостности.

При загрузке системы UNIX ваш компьютер осуществляет контроль целостности по состоянию корневой файловой системы. Если существуют потенциальные проблемы, программа fsck выполняется автоматически, чтобы исправить корневую файловую систему.

Чтобы вручную прогнать программу fsck, необходимо сначала демонтировать файловую систему (если вы проверяете корневую файловую систему, корень должен оставаться смонтированным).

Общий формат команды fsck:

fsck [-F FSType][-V][current\_options][-m]

[-o specific\_options][special...]

где -F - специфицирует тип FSType, с которым будут работать. FSType должен быть либо специфицирован здесь, либо определен из /etc/vfstab путем сопоставления special с элементом таблицы; -V - эхо-отображение всей командной строки; команду не выполнять; current\_options - параметры, поддерживаемые модулем программы fsck, специфичным для системы типа s5; -m - проверить, но не исправлять. Этот параметр проверяет, подходит ли файловая система для демонтирования; -o specific\_options - специфицирует подпараметры, которые специфичны для типа файловой системы - s5 или ufs.

Если файловая система сохраняет свою целостность, то дается отчет о количестве файлов, используемых блоках и свободных блоках. Если целостность файловой системы нарушена, пользователю дается подсказка выполнять операции паралельно перед каждой попыткой исправить систему.

#### *2.4.3.1. Использование команды fsck для проверки файловой системы s5*

Формат команды для файловых систем s5:

fsck [-F s5][generic\_options][-y][-n][-p][-sX][-tfile]

[-1][-q][-D][-f][-b][special]

Рекомендуется использовать параметр -y. Этот параметр отвечает "да" на все вопросы, выдаваемые командой fsck, и не требует вашего вмешательства. Другой рекомендуемый параметр -s, который форсирует перестройку свободного списка в оптимальном порядке. При использовании свободного списка он дезорганизуется; перестройка свободного списка улучшает работу с последовательно создаваемыми файлами. special дает имя специальному файлу устройства, связанному с файловой системой. Если имя устройства не определено, команда fsck проверяет все файловые системы, названные в /etc/vfstab с числовым полем fsckpass.

Следующий экран показывает команду fsck, введенную для проверки файловой системы usr. Никакие параметры не специфицируются. Отклик системы обозначает, что нарушения целостности обнаружено не было. Команда действует поэтапно; некоторые этапы команды прогоняются, только если это требуется или в ответ на параметр командной строки. По завершении каждого этапа отображается сообщение. В конце программы отображается заключительное сообщение, показывающее количество файлов (индексных дескрипторов файлов), блоков и свободных блоков.

# fsck -F s5 /dev/dsk/1s2

/dev/dsk/1s2

File System: usr Volume: usr

\*\*Phase 1 - Check Blocks and Sizes

\*\*Phase 2 - Check Pathnames

\*\*Phase 3 - Check Connectivity

\*\*Phase 4 - Check Reference Counts

\*\*Phase 5 - Check Free List

411 files 4394 blocks 8880 free

#

#### *2.4.3.2. Этапы программы fsck в системе s5*

Программа fsck прогоняется по этапам. На каждом этапе даются отчеты о любых ошибках, которые обнаружила программа. Если fsck может исправить ошибку, пользователя спрашивают, должно ли быть сделано исправление. Если вы определили параметр -y, тогда на все вопросы предполагается отвечать "да" и вам не будет дано приглашение. В последней части этого раздела описываются сообщения, выходящие на каждом этапе, возможные отклики и связанные с ними исключительные ситуации: ниже перечисляются сокращения, используемые в сообщениях об ошибках в программе fsck.

Аббревиатура сообщений об ошибках, выводимых fsck.

Следующая аббревиатура и соответствующий ей текст появляются в сообщении об ошибке:

|  |  |
| --- | --- |
| BLK | номер блока |
| DUP | номер блока копии |
| DIR | имя справочника |
| MTIME | время последней модификации |
| UNREF |  |
| CG | группа цилиндра |

Следующая однобуквенная аббревиатура заменяется на соответствующий ей текст при появлении на экране сообщения об ошибке:

|  |  |
| --- | --- |
| B | номер блока |
| F | имя файла |
| I | номер inode |
| M | режим файла |
| O | user-id владельца файла |
| S | размер файла |
| T | время последней модификации файла |
| X | счетчик связи |
|  | или число блоков BAD, DUP или MISSING |
|  | или число файлов (зависимых по контексту) |
| Y | номер счетчика корректного канала связи |
|  | или число блоков в файловой системе |
| Z | число свободных блоков |

#### *2.4.3.3. Этап инициализации*

Проверяется синтаксис командной строки. Перед проверкой файловой системы программа fsck устанавливает некоторые таблицы и открывает некоторые файлы. Программа fsck завершается, когда она сталкивается с ошибками на этапе инициализации.

#### *2.4.3.4. Общие ошибки*

Следующие три сообщения об ошибках могут появиться на любом этапе после инициализации. Их лучше рассматривать как фатальные, закончить выполнение программы и попытаться определить причину ошибки.

*Сообщение:*

CAN NOT SEEK: BLK B (CONTINUE?)

Просьба переместиться в указанный номер блока B в поврежденной файловой системе. Это сообщение указывает на серьезную ошибку, вероятно сбой в аппаратных средствах.

*Сообщение:*

CAN NOT READ: BLK B (CONTINUE?)

Просьба передать указанный номер блока B в поврежденную файловую систему. Сообщение указывает на серьезную ошибку, вероятно сбой в аппаратных средствах.

*Сообщение:*

CAN NOT WRITE: BLK B (CONTINUE?)

Просьба записать указанный номер блока B в поврежденную файловую систему. Диск может быть защищен от записи.

#### *2.4.3.5. Значение ответов Да/Нет*

Ответ n (нет) на приглашение CONTINUE? означает: *Закончите программу (Это рекомендуемый ответ)*

Ответ y (да) на приглашение CONTINUE? означает: *Попытайтесь продолжить проверку файловой системы.*

Обратите внимание, что ошибки часто могут повторятся. Эта исключительная ситуация предотвращает полную проверку файловой системы. Необходимо второй раз выполнить программу fsck, чтобы еще раз проверить файловую систему.

#### *2.4.3.6. Этап 1: Контроль блоков и размеров*

На этом этапе проверяется список индексных дескрипторов файлов. Даются отчеты об исключительных ситуациях, которые встречаются во время:

* контроля типов индексных дескрипторов файлов;
* установления таблицы подсчета нулевых связей;
* проверки номеров блоков индексных дескрипторов файлов на предмет плохих и идентичных блоков;
* проверки размеров индексных дескрипторов файлов;
* проверки формата индексных дескрипторов файлов.

*Типы сообщений об ошибках* - этап 1.

На этапе 1 выявляется 4 типа сообщений об ошибках:

* информационные сообщения;
* сообщения с приглашением CONTINUE?;
* сообщения с приглашением CLEAR?;
* сообщения с приглашением RECOVER?.

Между информационными сообщениями и сообщениями с приглашением CONTINUE? есть связь. Обычно приглашение CONTINUE? указывает на то, что был достигнут некоторый предел.

*Значение откликов* Да/Нет - этап 1.

Отклик n (нет) на приглашение CONTINUE? означает: *завершите программу.*

На этапе 1 отклик y (да) на приглашение CONTINUE? означает: *продолжайте выполнение программы.*

Когда возникает такая ошибка, полный контроль файловой системы невозможен. Чтобы повторно проверить файловую систему, нужно еще раз выполнить программу fsck.

Отклик n (нет) на приглашение RECOVER? означает: *восстановите все блоки, на которые указывает индексный дескриптор файла.*

Ответ "нет" подходит только в том случае, если пользователь намерен удалить избыточные блоки.

Ответ n (нет) на приглашение CLEAR? означает: *проигнорируйте исключительную ситуацию.*

Ответ "нет" уместен только в том случае, если пользователь намерен использовать другие средства для фиксации ошибки.

Ответ y (да) на приглашение CLEAR? означает: *освободите индексный дескриптор файла I путем обнуления его содержимого.*

Это может порождать исключительную ситуацию UNALLOCATED на этапе 2 для каждого элемента каталога, указывающего на этот индексный дескриптор файла.

Сообщения об ошибках на этапе 1.

*Сообщение:*

swt UNKNOWN FILE TYPE I- I (CLEAR?)

Видовое слово индексного днскриптора файла I указывает на то, что индексный дескриптор файла это не абстрактный файл, не специальный символьный индексный дескриптор файла, не регулярный индексный дескриптор файла и не индексный дескриптор файла каталогов. Если определен параметр -p, индексный дескриптор файла очищается.

*Сообщение:*

LINK COUNT TABLE OVERFLOW (CONTINUE?)

Во внутренней таблице для fsck, содержащей размещенные индексные дескрипторы файлов с нулевым количеством связей, больше нет места. Если определен параметр -p, программа завершится, а программу fsck нужно завершить вручную.

*Сообщение:*

B BAD I- I

Индексный дескриптор файла I содержит блок номеров B с номером меньше номера первого блока данных в файловой системе или больше номера последнего блока в файловой системе. Эта исключительная ситуация может порождать сообщение об ошибке EXCESSIVE BAD BLKS на этапе 1, если у индексного дескриптора файлов I слишком много номеров блоков за пределами файловой системы. Эта исключительная ситуация порождает сообщение об ошибке BAD/DUP на этапах 2 и 4.

*Сообщение:*

EXCESSIVE BAD BLOCKS I- I (CONTINUE?)

В файловой системе слишком много (обычно больше 10) блоков с номером меньше номера первого блока данных или больше номера последнего блока, связанного с индексным дескриптором файлов I. Если специфицирован параметр -p, программа завершается.

*Сообщение:*

B DUP I- I

Индексный дескриптор файла I содержит блок номер B, о котором заявляет такой же или другой индексный дескриптор файла или свободный список. Такая исключительная ситуация может порождать сообщение об ошибках EXCESSIVE DUP BLKS на этапе 1, если у индексного дескриптора файлов слишком много номеров блоков, о которых заявлено таким же или другим индексным дескриптором файлов или в свободном списке. Эта исключительная ситуация вызывает этап 1B и порождает сообщения об ошибках на этапах 2 и 4.

*Сообщение:*

EXCESSIVE DUP BLKS I- I (CONTINUE?)

Слишком много (обычно больше 10) блоков, о которых заявлено в том же или другом индексном дескрипторе файлов или в свободном списке. Если определен параметр -p, программа завершается.

*Сообщение:*

DUP TABLE OVERFLOW (CONTINUE?)

Во внутренней таблице в fsck, содержащей двойные номера блоков, больше нет места. Если определен параметр -p, программа завершается.

*Сообщение:*

DIRECTORY MISALIGNED I- I

Размер каталога индексных дескрипторов файлов не кратен 16. Если используется параметр -p, каталог восстанавливается автоматически.

*Сообщение:*

PARTIALLY ALLOCATED INODE I- I (CLEAR?)

Индексный дескриптор файла ни выделен, ни освобожден. Если специфицирован параметр -p, индексный дескриптор файла будет очищен.

*Сообщение:*

DIR/FILE SIZE ERROR

Файл ссылается на данные, а не обозначается индексным дескриптором файла.

*Сообщение:*

DELETE OR RECOVER EXCESS DATA

У пользователя есть право выбора между удалением или восстановлением избыточных блоков, на которые указывает индексный дескриптор файла.

*Сообщение:*

RECOVER?

Файл ссылается на данные большие, чем указывает на них индексный дескриптор файла. Пользователю дается право выбора коррекции информации индексного дескриптора файла. Если специфицирован параметр -p, данные восстанавливаются.

*Сообщение:*

DELETE?

Файл делает больше ссылок на данные, чем указывает индексный дескриптор файла. Пользователю дается право выбора между удалением блоков со ссылками и возможностью оставить нетронутыми данные индексного дескриптора файла.

#### *2.4.3.7. Этап 1B: Повторный поиск DUPS*

Когда в файловой системе находится дублетный блок, эта файловая система повторно просматривается, чтобы найти индексный дескриптор файла, который перед этим заявлял о данном блоке. После того, как будет найден дублетный блок, печатается следующее информационное сообщение:

*Сообщение:*

DUP I- I

Индексный дескриптор файла I содержит блок номер B, о котором уже было заявлено в том же или другом индексном дескрипторе файла или в свободном списке. Эта исключительная ситуация порождает сообщение об ошибке BAD/DUP на этапе 2. Индексные дескрипторы файлов, имеющие перекрывающиеся блоки, могут быть определены при изучении этой исключительной ситуации и исключительной ситуации DUP на этапе 1.

#### *2.4.3.8. Этап 2: Контроль полных имен файлов*

На этом этапе удаляются элементы каталога, указывающие на неудачные индексные дескрипторы файлов, обнаруженные на этапах 1 и 1B. Дается отчет об исключительных ситуациях, появившихся в результате следующих моментов:

* неправильный вид и состояние корневого индексного дескриптора файла;
* указатели каталога индексных дескрипторов файлов находятся за пределами допустимого диапазона;
* элементы каталога указывают на неверные индексные дескрипторы файлов.

*Типы сообщений об ошибках* - этап 2.

На этапе 2 имеется 4 типа сообщений об ошибках:

* информационные сообщения;
* сообщения с приглашением FIX?;
* сообщения с приглашением CONTINUE?;
* сообщения с приглашением REMOVE?.

*Значение откликов* Да/Нет - этап 2.

Отклик n (нет) на приглашение FIX? означает: *завершите программу, т.к. fsck не может быть продолжена.*

Отклик y (да) на приглашение FIX? означает: *замените тип корневого индексного дескриптора файла на "каталог".*

Если блоки данных корневого индексного дескриптора файла не являются блоками каталога, порождается слишком много сообщений об ошибках.

Отклик n (нет) на приглашение CONTINUE? означает: *завершите программу.*

Отклик y (да) на приглашение CONTINUE? означает: *проигнорируйте сообщение об ошибках DUPS/BAD IN ROOT INODE и продолжайте проверять файловую систему.*

Если корневой индексный дескриптор файла неправильный, может порождаться большое число других сообщений об ошибках.

Отклик n (нет) на приглашение REMOVE? означает: *проигнорируйте исключительную ситуацию.*

Ответ "нет" подходит только в том случае, если пользователь намерен предпринять другие меры по фиксации ошибки.

Отклик y (да) на приглашение REMOVE? означает: *Удалить идентичные или освобожденные блоки.*

Сообщения об ошибках - этап 2.

*Сообщение:*

ROOT INODE UNALLOCATED. TERMINATING

У корневого индексного дескриптора файла (обычно это индексный дескриптор файла номер 2) файловой системы нет выделенных режимных разрядов. Это сообщение указывает на серьезную ошибку, которая приводит к остановке программы.

*Сообщение:*

ROOT INODE NOT DIRECTORY (FIX?)

Корневой индексный дескриптор файла (обычно это индексный дескриптор файла номер 2) файловой системы не является типом индексного дескриптора каталога файлов. Если специфицирован параметр -p, программа завершится.

*Сообщение:*

DUPS/BAD IN ROOT INODE (CONTINUE?)

На этапе 1 или 1B найдены идентичные блоки или неисправные блоки в корневом индексном дескрипторе файла (обычно это индексный дескриптор файла номер 2) файловой системы. Если специфицирован параметр -p, программа завершится.

*Сообщение:*

I OUT OF RANGE I- I NAME- F (REMOVE?)

Элемент каталога F содержит индексный дескриптор файла номер I, который больше, чем конец списка индексных дескрипторов файла. Если специфицирован параметр -p, индексный дескриптор файла будет автоматически удален.

*Сообщение:*

UNALLOCATED I- I OWNER- O MODE- N SIZE- S MTIME- T

NAME- F (REMOVE?)

Элемент каталога F содержит индексный дескриптор файла I без выделенных режимных разрядов. Печатаются: владелец O, режим M, размер S, время модификации T и имя файла F. Если файловая система не смонтирована и параметр -n не специфицирован, элемент каталога автоматически удаляется, если у индексного дескриптора файла, который указывает на этот элемент, размер символа 0. Элемент удаляется, если специфицирован параметр -p.

*Сообщение:*

DUP/BAD I- I OWNER- O MODE- M SIZE- S MTIME- T DIR- F

(REMOVE?)

На этапе 1 или 1B найдены идентичные блоки или "неудачные" блоки, связанные с элементом F каталога, индексного дескриптора каталогов файлов I. Печатаются: владелец O, режим M, размер S, время модификации T, и имя каталога F. Если специфицирован параметр -p, идентичные/неудачные блоки удаляются.

*Сообщение:*

DUP/BAD I- I OWNER- O MODE- M SIZE- S MTIME- T FILE- F

(REMOVE?)

На этапе 1 или 1B найдены идентичные или "неудачные" блоки, связанные с элементом F файла, индексного дескриптора файла I. Печатаются: владелец O, режим M, размер S, время модификации T и имя файла F. Если специфицирован параметр -p, идентичные/неудачные блоки удаляются.

*Сообщение:*

BAD BLK IN DIR I- I OWNER- O MODE- SIZE- S MTIME- T

Это сообщение возникает только тогда, когда используется параметр -D. В каталоге с индексным дескриптором I был обнаружен физически поврежденный блок. Исключительными ситуациями, отыскиваемые в блоках каталогов, могут быть ненулевыми заполняемыми элементами, несогласованными "." и ".." элементами и вложенные наклонные черты вправо в поле имени. Это сообщение об ошибках означает, что пользователю нужно либо удалить индексный дескриптор каталога, если весь блок кажется поврежденным, либо заменить (или удалить) те элементы каталога, которые кажутся неудачными.

#### *2.4.3.9. Этап 3: Проверка связности*

На этом этапе проверяются каталоги, исследуемые на этапе 2. Даются отчеты об следующих исключительных ситуациях:

* каталоги без ссылок;
* отсутствующие или полные каталоги lost+found.

*Типы сообщений об ошибках* - Этап 3.

На этапе 3 два типа сообщений об ошибках:

* информационные сообщения;
* сообщения с приглашением RECONNECT?.

*Значение откликов* Да/Нет - Этап 3

Отклик n (нет) на приглашение RECONNECT? означает: *Проигнорируйте исключительную ситуацию.*

Этот ответ порождает сообщения об ошибках UNREF на этапе 4.

Ответ "нет" подходит только в том случае, если пользователь намерен предпринять другие меры для фиксации ошибки.

Ответ y (да) на приглашение RECONNECT? означает: *Заново подсоедините каталог с индексным дескриптором I к файловой системе в каталоге для потерянных файлов (обычно это каталог lost+found).*

Это может порождать сообщения об ошибках lost+found, если есть проблемы с соединением каталога с индексным дескриптором I с каталогом lost+found. Если связь удачна, появляется информационное сообщение CONNECTED.

Этап 3 - сообщения об ошибках.

*Сообщение:*

UNREF DIR I- I OWNER- O MODE- M SIZE- S MTIME- T

(RECONNECT?)

Каталог с индексным дескриптором I не был соединен с элементом каталога при прослеживании файловой системы. Печатаются владелец O, режим M, размер S и время модификации T каталога. Программа fsck форсирует повторное соединение непустого каталога. Если специфицируется параметр -p, заново соединяется непустой каталог.

*Сообщение:*

SORRY. NO lost+found DIRECTORY

В корневом каталоге файловой системы нет каталога lost+found; fsck игнорирует требование соединения любого каталога с каталогом lost+found. Это порождает сообщение об ошибках UNREF на этапе 4. Режимы доступа каталога lost+found могут быть неверными.

*Сообщение:*

SORRY. NO SPACE IN lost + found DIRECTORY

В корневом каталоге файловой системы нет места для добавления другого элемента к каталогу lost+found; fsck игнорирует требование соединения любого каталога с каталогом lost+found. Это порождает сообщение об ошибках UNREF на этапе 4. Очистить ненужные элементы в каталоге lost+found или расширить его.

*Сообщение:*

DIR I- I1 CONNECTED. PARENT WAS I- I2

Это справочное сообщение, указывающее, что каталог с индексным дескриптором I1 был успешно соединен с каталогом lost+found. Родительский индексный дескриптор файла I2 каталога с индексным дескриптором I1 замещен номером индексного дескриптора каталога lost+found.

#### *2.4.3.10. Этап 4: Проверка контрольного счета*

На этом этапе проверяется объединенная информация о счетах, полученная на этапах 2 и 3. Дается отчет об исключительной ситуации, которая возникла в результате:

* наличия файлов без ссылок;
* отсутствующего или заполненного каталога lost+found;
* неверного подсчета связей для файлов, каталогов или специальных файлов;
* наличия файлов и каталогов без ссылок;
* наличия неудачных или идентичных блоков в файлах и каталогах;
* неверного общего подсчета свободных индексных дескрипторов файлов.

*Типы сообщений об ошибках* - Этап 4.

На этапе 4 имеется пять типов сообщений об ошибках:

* информационные сообщения;
* сообщения с приглашением RECONNECT?;
* сообщения с приглашением CLEAR?;
* сообщения с приглашением ADJUST?;
* сообщения с приглашением FIX?.

*Значение откликов* Да/Нет - Этап 4.

Отклик n (нет) на приглашение RECONNECT? означает: *Проигнорировать эту исключительную ситуацию.*

Этот ответ в дальнейшем порождает сообщение об ошибках CLEAR на этапе 4.

Отклик y (да) на проиглашение RECONNECT? означает: *Заново соедините индексный дескриптор файла I с файловой системой в каталоге для потерянных файлов (это обычно каталог lost+found).*

Это может порождать сообщение об ошибках lost+found на этом этапе, если есть проблемы соединения индексного дескриптора файла I с каталогом lost+found.

Отклик n (нет) на проиглашение CLEAR? означает: *Проигнорировать эту исключительную ситуацию.*

Этот ответ подходит только в том случае, если пользователь намерен принять другие меры по фиксации ошибки.

Отклик y (да) на приглашение CLEAR? означает: *Освободите индексный дескриптор файла путем обнуления его содержания.*

Отклик n (нет) на приглашение ADJUST? означает: *Проигнорируйте исключительную ситуацию.*

Этот ответ уместен только в том случае, когда пользователь намерен принять другие меры по фиксации ошибки.

Отклик y (да) на приглашение ADJUST? означает: *Заменить подсчет связей файла индексных дескрипторов файлов I на Y.*

Отклик n (нет) на приглашение FIX? означает: *Проигнорируйте эту исключительную ситуацию.*

Этот ответ уместен только в том случае, когда пользователь намерен принять другие меры по фиксации ошибки.

Отклик y (да) на приглашение FIX? означает: *Замените подсчет в системном блоке тома на правильный подсчет.*

Этап 4 - Сообщения об ошибках.

*Сообщение:*

UNREF FILE I- I OWNER- O MODE- M SIZE- S MTIME- T

(RECONNECT?)

Индексный дескриптор файла I не был соединен с элементом каталога при проверке файловой системы. Печатаются владелец O, режим M, размер S и время модификации T индексного дескриптора файла I. Если параметр -n опущен, и файловая система не смонтирована, пустые файлы автоматически очищаются. Непустые файлы не очищаются. Если специфицирован параметр -p, индексный дескриптор файла заново подсоединяется.

*Сообщение:*

SORRY. NO lost+found DIRECTORY

В корневом каталоге файловой системы нет каталога lost+found; fsck игнорирует требование соединить файл с каталогом lost+found. В дальнейшем это порождает сообщение об ошибках на этапе 4. Режимы доступа каталога lost+found могут быть неправильными.

*Сообщение:*

SORRY. NO SPACE IN lost+found DIRECTORY

Нет места, чтобы присоединить другой элемент к каталогу lost+found в корневом каталоге файловой системы; fsck игнорирует требование связать файл с каталогом lost+found. Это порождает в дальнейшем сообщение об ошибках CLEAR на этапе 4. Проверьте размер и содержание каталога lost+found.

*Сообщение:*

(CLEAR)

Индексный дескриптор файла, упомянутый в сообщении об ошибках UNREF, невозможно заново соединить.

*Сообщение:*

LINK COUNT FILE I- I OWNER- O MODE- M SIZE- S MTIME- T

COUNT- X SHOULD BE Y (ADJUST?)

Подсчет связей для файла с индексным дескриптором I - X, а должен быть Y. Печатаются владелец O, режим M, размер S и время модификации T. Если специфицирован параметр -p, корректируется подсчет связей.

*Сообщение:*

LINK COUNT DIR I- I OWNER- O MODE- M SIZE- S MTIME- T

COUNT- X SHOULD BE Y (ADJUST?)

Подсчет связей для каталога с индексным дескриптором I - X, а должен быть Y. Печатаются владелец O, режим M, размер S и время модификации T. Если специфицирован параметр -p, корректируется подсчет связей.

*Сообщение:*

UNREF FILE I- I OWNER- O MODE- M SIZE- S MTIME- T

(CLEAR?)

Файл с индексным дескриптором I не был соединен с элементом каталога при отслеживании файловой системы. Печатаются владелец O, режим M, размер S и время модификации T индексного дескриптора файла I. Если параметр -n пропущен и файловая система не смонтирована, пустые файлы автоматически очищаются. Непустые каталоги не очищаются. Если специфицирован параметр -p, файл очищается, если он не был заново соединен.

*Сообщение:*

UNREF DIR I- I OWNER- O MODE- M SIZE- S MTIME- T

(CLEAR?)

Каталог с индексным дескриптором I не был соединен с элементом каталога при отслеживании файловой системы. Печатаются владелец O, режим M, размер S и время модификации T индексного дескриптора файла I. Если параметр -n пропущен и файловая система не смонтирована, пустые каталоги автоматически очищаются. Непустые каталоги не очищаются. Если специфицирован параметр -p, каталог очищается, если его невозможно заново соединить.

*Сообщение:*

BAD/DUP FILE I- I OWNER- O MODE- M SIZE- S MTIME- T

(CLEAR?)

На этапе 1 или на этапе 1B были найдены идентичные блоки или неудачные блоки, связанные с файлом с индексным дескриптором I. Если специфицирован параметр -p, файл очищается.

*Сообщение:*

BAD/DUP DIR I- I OWNER- O MODE- M SIZE- S MTIME- T

(CLEAR?)

На этапе 1 или на этапе 1B были найдены идентичные блоки или неудачные блоки, связанные с каталогом с индексным дескриптором I. Печатаются владелец O, режим M, размер S и время модификации T индексного дескриптора I. Если специфицирован параметр -p, каталог очищается.

*Сообщение:*

FREE INODE COUNT WRONG IN SUPERBLK (FIX?)

Фактический подсчет свободных индексных дескрипторов файлов не соответствует подсчету в системном блоке тома файловой системы. Если специфицирован параметр -q или -p, подсчет в системном блоке тома будет автоматически зафиксирован.

#### *2.4.3.11. Этап 5: Проверка списка свободной памяти*

На этом этапе проверяется список свободных блоков. Дается отчет об исключительной ситуации, возникшей в результате:

* неудачных блоков в списке свободных блоков;
* неверного подсчета свободных блоков;
* идентичных блоков в списке свободных блоков;
* неиспользованных блоков из файловой системы, которых нет в списке свободных блоков;
* неверного общего подсчета свободных блоков.

*Типы сообщений об ошибках* - Этап 5.

На этапе 5 четыре типа сообщений об ошибках:

* информационные сообщения;
* сообщения с приглашением CONTINUE?;
* сообщения с приглашением FIX?;
* сообщения с приглашением SALVAGE?;

*Значение ответов* Да/Нет - Этап 5.

Ответ n (нет) на приглашение CONTINUE? означает: *Завершите программу.*

Ответ y (да) на приглашение CONTINUE? означает: *Проигнорируйте остаток списка свободных блоков и продолжите выполнение fsck.*

Это порождает в дальнейшем сообщение об ошибках

BAD BLKS IN FREE LIST на этапе 5.

Ответ n (нет) на приглашение FIX? означает: *Проигнорируйте эту исключительную ситуацию.*

Этот ответ уместен только в том случае, если пользователь намерен принять другие меры по фиксации ошибки.

Ответ y (да) на приглашение FIX? означает: *Замените подсчет в системном блоке тома на правильный.*

Ответ n (нет) на приглашение SALVAGE? означает: *Проигнорируйте эту исключительную ситуацию.*

Этот ответ уместен только в том случае, если пользователь намерен принять другие меры по фиксации ошибки.

Ответ y (да) на приглашение SALVAGE? означает: *Замените фактический список свободных блоков новым списком свободных блоков.*

Новый список свободных блоков упорядочивается в соответствии со спецификацией промежутков и цилиндров параметра -s или -S, чтобы сократить время, затрачиваемое на циклический сдвиг диска в некоторое положение.

Этап 5 - Сообщения об ошибках.

*Сообщение:*

EXCESSIVE BAD BLKS IN FREE LIST (CONTINUE?)

В списке свободных блоков слишком много блоков со значением, меньше первого блока данных в файловой системе или больше последнего блока. Если специфицирован параметр -p, программа завершается.

*Сообщение:*

EXCESSIVE DUP BLKS IN FREE LIST (CONTINUE?)

В списке свободных блоков слишком много блоков, объявленных индексными дескрипторами файлов или в предыдущих частях списка свободных блоков. Если специфицирован параметр -p, программа завершается.

*Сообщение:*

BAD FREEBLK COUNT

Подсчет свободных блоков в списке свободных блоков больше 50 или меньше 0. Эта ситуация порождает сообщение BAD FREE LIST на этапе 5.

*Сообщение:*

X BAD BLKS IN FREE LIST

X блоков в списке свободных блоков имеют номер меньше первого блока данных или больше последнего блока в файловой системе. Такая ситуация порождает сообщение BAD FREE LIST на этапе 5.

*Сообщение:*

X DUP BLKS IN FREE LIST

X блоков, объявленных индексными дескрипторами файлов или в предыдущих частях свободного списка блоков, были обнаружены в списке свободных блоков. Эта ситуация порождает сообщение BAD FREE LIST на этапе 5.

*Сообщение:*

X BLK(S) MISSING

X блоков, не использованны файловой системой, не были найдены в списке свободных блоков. Эта ситуация порождает в дальнейшем сообщение BAD FREE LIST на этапе 5.

*Сообщение:*

FREE BLK COUNT WRONG IN SUPERBLOCK (FIX?)

Действительный подсчет свободных блоков не соответствует подсчету свободных блоков в системном блоке тома файловой системы. Если был специфицирован параметр -p, подсчет свободных блоков в системном блоке тома фиксируется автоматически.

*Сообщение:*

BAD FREE LIST (SALVAGE?)

Этому сообщению всегда предшествует одно или несколько информационных сообщений этапа 5. Если был специфицирован параметр -q или -p, список свободных блоков восстанавливается автоматически.

#### *2.4.3.12. Этап 6: Восстановление списка свободных блоков*

На этом этапе восстанавливается список свободных блоков. Может быть отображено справочное сообщение о значениях пропускаемых (игнорируемых) блоков или блоков на один цилиндр.

Этап 6 - Сообщения об ошибках.

*Сообщение:*

DEFAULT FREE-BLOCK LIST SPACING ASSUMED

Это справочное сообщение, указывающее, что пропускаемые блоки (промежуток) больше блоков на один цилиндр; значение игнорируемых блоков меньше 1, и блоков на цилиндр - меньше 1, либо значение блоков на цилиндр больше 500. Используются значения по умолчанию: 10 игнорируемых блоков и 162 блока на цилиндр.

*Примечание.* Так как используемые значения по умолчанию могут быть неточными для вашей системы, необходимо быть осторожными при спецификации значений с параметром -s в командной строке.

#### *2.4.3.13. Этап доводки*

После проверки файловой системы выполняется несколько функций очистки. Ниже перечисляются справочные сообщения о файловой системе и о модифицированном состоянии файловой системы.

**\*\*\*\*\* FILE SYSTEM STATE SET TO OKAY \*\*\*\*\***

Устанавливается флаг в системном блоке тома, указывающий, что файловая система не нарушена и может быть смонтирована.

**X files Y blocks Z free**

Это сообщение указывает, что файловая система, которая была проверена, содержала X файлов, использующих Y блоков; при этом Z блоков оставалось свободными в файловой системе.

**\*\*\*\*\* FSCK and the ROOT FILE SYSTEM \*\*\*\*\***

root - единственная файловая система, которую можно (и нужно) проверять во время монтирования. Для проверки файловой системы root имеются автоматические механизмы. Эти механизмы служат для обработки корневой файловой системы при начальной загрузке и для периодической проверки во время закрытия системы. Можно также форсировать проверку закрытия системы. Эти механизмы упрятывают сообщения от fsck. Если они не упрятаны, вы увидите следующее сообщение об ошибках.

**\*\*\*\*\* ROOT FILE SYSTEM WAS MODIFIED \*\*\*\*\***

Это справочное сообщение указывает, что корневая файловая система была модифицирована с помощью программы fsck. Если необходима повторная загрузка системы, fsck с параметром -b форсирует автоматическую перезагрузку и печатает следующее сообщение:

**\*\*\*\*\* SYSTEM WILL REBOOT AUTOMATICALLY \*\*\*\*\***

Если вы решили не использовать автоматические механизмы, если параметр -b не используется и требуется перезагрузка системы, нажмите RESET.   
Можно автоматически заново смонтировать root без перезагрузки после повреждения этой файловой системы. Появляется следующее сообщение:

**\*\*\*\*\* ROOT REMOUNTED \*\*\*\*\***

Автоматические процедуры устанавливают соответствующие условия (процессы не связаны с файлами) для проверки root.

*Примечание.* Всегда используйте автоматические процедуры для root. Никогда не используйте команду fsck в других файловых системах, когда они монтируются. Если вы попытаетесь использовать команду fsck в смонтированной файловой системе, отличающейся от корневой (root) файловой системы, отображается следующее сообщение:

/dev/dsk/ ?? is a mounted file system, ignored. ?? is

the special device name.

#### 2.4.4. Использование команды fsck для проверки файловой системы ufs

Ниже приводится формат для использования fsck в файловых системах ufs:

fsck [-F ufs][generic\_options][current\_options]

[-o p,b=#,w][special....]

где -F ufs - специфицирует тип файловой системы ufs; generic\_options - параметры, поддерживаемые командой fsck; current\_options - параметры, поддерживаемые специфическим модулем команды fsck; -o - специфицирует параметры, специфичные для файловой системы ufs. Этими подпараметрами может быть любая комбинация из приводимых ниже:

* p - Проверяет диски в параллели, максимально используя перекрытие ввода/вывода для более быстрой проверки файловой системы.   
  b=# - Использовать блок, специфицированный непосредственно за флагом, как суперблок файловой системы. Блок 32 всегда является альтернативным для суперблока.   
  w - Проверить только записываемые файловые системы.

Команда fsck проверяет и исправляет в диалоговом режиме несогласованные условия в файловых системах. При каждом исправлении необходимо подождать, пока пользователь ответит "да" или "нет". Если использовались параметры -y, команда fsck допускает ответ "да" на каждую возможную взаимосвязь и не делают паузы для ответа.

fsck - многопроходная команда контроля файловых систем. Каждый проход файловой системы активизирует различные этапы программы fsck. После инициализации команда fsck выполняет последовательность проходов для каждой файловой системы, проверяя блоки и размеры, полные имена файлов, связность, подсчет ссылок и карту свободных блоков (возможно, перестраивая ее) и выполняет очистку.

Далее объясняется значение каждого сообщения об ошибках, возможных откликов и соответствующих исключительных сиуаций. Исключительные ситуации организуются на том этапе программы fsck, на котором они могут появиться. Исключительные ситуации, которые могут возникать на нескольких этапах, обсуждаются в разделе об инициализации.

#### *2.4.4.1. Этап инициализации*

Перед проверкой файловой системы должны быть установлены определенные таблицы и открыты определенные файлы. Следующие сообщения связаны с исключительными ситуациями, вытекающими из параметров командной строки, запроса памяти, открытия файлов, состояния файлов, проверки размера файловой системы и создания рабочего файла.

*Сообщение:*

cannot alloc NNN bytes for blockmap

cannot alloc NNN bytes for freemap

cannot alloc NNN bytes for statemap

cannot alloc NNN bytes for lncntp

Запрос команды fsck на память для таблиц виртуальной памяти был неудачным. Такого никогда не должно быть. Если все же это случается, fsck завершается. Это серьезная системная ошибка, которую нужно немедленно исправлять.

*Сообщение:*

Can't open checklist file: F

Контрольную таблицу файловой системы или файл F со значением по умолчанию (обычно /etc/vfstab) невозможно открыть для считывания. Когда это происходит, fsck завершается. Проверьте режимы доступа файла F.

*Сообщение:*

Can't stat root

Запрос fsck статических данных о корневом каталоге был неудачным. Если это случается, fsck завершается.

*Сообщение:*

Can't stat F

Can't make sense out of name F

Запрос команды fsck статических данных о файловой системе F был неудачным. При интерактивном прогоне команда игнорирует эту файловую систему и продолжает проверять другую файловую систему. Проверьте режимы доступа файла F.

*Сообщение:*

Can't open F

Команда fsck неудачно пытается открыть файловую систему F. При диалоговом режиме прогона команды, она игнорирует эту файловую систему и продолжает проверку следующей файловой системы. Проверьте режимы доступа F.

*Сообщение:*

F: (NO WRITE)

Либо был специфицирован флаг -n, либо команда fsck не смогла открыть файловую систему F для записи. При диалоговом режиме прогона команды распечатывается вся диагностика, но fsck не пытается ничего зафиксировать.

*Сообщение:*

file is not a block or character device; OK

Пользователь ошибочно присвоил команде fsck имя регулярного файла. Проверьте специфицированный тип файла.

Возможными отетами на приглашение OK являются:

*YES* (да) - проигнорировать эту исключительную ситуацию;

*NO* (нет) - проигнорировать эту файловую систему и продолжить проверку следующей файловой системы.

*Сообщение:*

UNDEFINED OPTIMIZATION IN SUPERBLOCK (SET TO DEFAULT)

Параметром оптимизации системного блока тома не является ни OPT\_TIME, ни OPT\_SPACE.

Возможными ответами на приглашение SET TO DEFAULT являются:

*YES* - установить системный блок тома для запроса оптимизации с целью сокращения времени выполнения системы. (Если желательна оптимизация для минимизации использования пространства на диске, системный блок тома можно установить с помощью tunefs (1M);

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

IMPOSSIBLE MINFREE-D IN SUPERBLOCK (SET TO DEFAULT)

Минимальный процент пространства системного блока тома больше 99% или меньше 0%.

Возможные ответы на приглашение SET TO DEFAULT:

*YES* - установить параметр minfree на 10%. (Если нужен другой процентный показатель, его можно установить, ипользуя tunefs (1M));

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

MAGIC NUMBER WRONG

NCG OUT OF RANGE

CPG OUT OF RANGE

NCYL DOES NOT JIVE WITH NCG+CPG

SIZE PREPOSTEROUSLY LARGE

TRASHED VALUES IN SUPER BLOCK

за ним идет следующее сообщение:

F: BAD SUPER BLOCK: B

USE -b OPTION TO FSCK TO SPECIFY LOCATION OF AN

ALTERNATE

SUPER-BLOCK TO SUPPLY NEEDED INFORMATION; SEE fsck (1M)

Системный блок тома был испорчен. Из имеющихся копий следует выбрать альтернативный системный блок тома. Выберите альтернативный системный блок тома путем просчитывания его смещения.

*Сообщение:*

INTERNAL INCONSISTENCY: M

В команде fsck была внутренняя несогласованность, сообщение о которой M. Это не должно происходить, в противном случае обратитесь к специалисту.

*Сообщение:*

CAN NOT SEEK: BLK B (CONTINUE)

Запрос команды fsck на перемещение в определенный блок с номером B в файловой системе был неудачным. Если это произошло, обратитесь к специалисту.

Возможные ответы на приглашение CONTINUE:

*YES* - попытаться продолжить проверку файловой системы. (Обратите внимание, что часто ошибка продолжает существовать). Эта исключительная ситуация предупреждает полный контроль файловой системы. Необходимо второй раз прогнать fsck для повторной проверки файловой системы. Если блок был частью буферного кеша виртуальной памяти, fsck завершится со следующим сообщением:

Fatal I/O error

*NO* - завершить программу.

*Сообщение:*

CAN NOT READ: BLK B (CONTINUE)

Запрос команды fsck на считывание опеделенного блока номер B в файловой системе был неудачным. Обратитесь к специалисту.

Возможные ответы на приглашение CONTINUE:

*YES* - попытаться продолжить проверку файловой системы. Fsck делает еще одну попытку считывания и распечатывает

*Сообщение:*

THE FOLLOWING SECTORS COULD NOT BE READ: N

где N обозначает секторы, которые невозможно считать.

Если fsck пытается снова записать один из блоков, считывание которых было невозможным, появляется следующее

*Сообщение:*

WRITING ZERO'ED BLOCK N TO DISK

где N обозначает сектор, который был записан с нулями.

Если в отношении диска есть технические ошибки, они будут продолжать существовать. Эта исключительная ситуация предупреждает полный контроль файловой системы. Для повторной проверки файловой системы необходимо второй раз прогнать fsck. Если блок был частью буферного кеша виртуальной памяти, fsck завершится со следующим сообщением:

Fatal I/O error

*NO* - завершить программу.

*Сообщение:*

CAN NOT WRITE: BLK B (CONTINUE)

Запрос fsck на запись специфицированного блока номер B в файловой системе был неудачным. Проверьте, чтобы убедиться, что диск не защищен от записи.

Возможные ответы на приглашение CONTINUE:

*YES* - попытаться продолжить проверку файловой системы. Делается еще одна попытка записать операцию. Секторы, которые невозможно записать, будут указаны в сообщении:

THE FOLLOWING SECTORS COULD NOT BE WRITTEN: N

где N обозначает секторы, которые невозможно записать.

Если есть технические трудности, связанные с диском, ошибка продолжает существовать. Эта исключительная ситуация предупреждает полный контроль файловой системы. Для повторной проверки этой файловой системы необходимо второй раз прогнать fsck. Если блок был частью буферного кеша виртуальной памяти, fsck завершится сообщением:

Fatal I/O error

*NO* - завершить программу.

*Сообщение:*

bad inode number DDD to ginode

Внутренняя ошибка была вызвана попыткой считать несуществующий индексный дескриптор файла DDD. Эта ошибка приводит к выходу программы fsck. Обратитесь к специалисту.

#### *2.4.4.2. Этап 1: Контроль блоков и размеров*

На этом этапе проверяется список индексных дескрипторов файлов. Дается отчет об исключительных ситуациях, возникших при:

* прверке типов индексных дескрипторов файлов;
* установлении таблицы подсчета нулевых связей;
* изучении номеров блоков индексных дескрипторов файлов и выявлении неудачных или идентичных блоков;
* проверке размера индексных дескриптров файлов;
* проверке формата индексных дескрипторов файлов.

Все ошибки на этом этапе, за исключением INCORRECT BLOCK COUNT и PARTIALLY TRUNCATED INODE, являются фатальными, если очищается файловая система.

Этап 1 - Сообщения об ошибках.

*Сообщение:*

UNKNOWN FILE TYPE I- I (CLEAR)

Режим индексного дескриптора файла I обозначает, что индексный дескриптор файла не является индексным дескриптором специального блокового файла или специального символьного файла, или индексным дескриптором файла socket, регулярным индексным дескриптором файла, символьной связью, файлом FIFO или индексным дескриптором каталога.

Возможные ответы на приглашение CLEAR:

*YES* - освободить индексный дескриптор файла I путем обнуления его содержимого. Это всегда порождает сообщение об ошибках UNALLOCATED на этапе 2 для каждого элемента каталога, указывающего на этот индексный дескриптор файла;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

PARTIALLY TRUNCATED INODE I-I (SALVAGE)

Команда fsck обнаружила индексный дескриптор файла I, размер которого меньше, чем число блоков, выделенных для него. Это условие должно появляться только в том случае, когда система терпит крах при укорачивании файла. При очистке файловой системы fsck завершает укорачивание до заданного размера.

Возможные ответы на приглашение SALVAGE:

*YES* - завершить укорачивание до размера, заданного в индексном дескрипторе файла.

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

LINK COUNT TABLE OVERFLOW (CONTINUE)

Во внутренней таблице для fsck, содержащей выделенные индексные дескрипторы файлов с подсчетом нулевых связей, больше нет места.

Возможные ответы на приглашение CONTINUE:

*YES* - продолжить программу. Эта исключительная ситуация предупреждает полный контроль файловой системы. Для повторной проверки файловой системы необходимо второй раз прогнать fsck. Если найден другой выделенный индексный дескриптор файла с подсчетом нулевых связей, сообщение об ошибках повторяется.

*NO* - завершить программу.

*Сообщение:*

B BAD I-I

Индексный дескриптор файла I содержит блок B с номером меньше номера первого блока данных в файловой системе или больше номера последнего блока в файловой системе. Эта исключительная ситуация может порождать на этапе 1 сообщение об ошибках EXCESSIVE BAD BLKS, если у индексного дескриптора файла I слишком много блоков с номерами за пределами файловой системы. Эта исключительная ситуация порождает на этапах 2 и 4 сообщение об ошибках BAD/DUP.

*Сообщение:*

EXCESSIVE BAD BLKS I-I (CONTINUE)

В файловой системе слишком много (обычно больше 10) блоков с номером меньше номера первого блока данных или больше номера последнего блока в файловой системе, связанной с индексным дескриптором файла I.

Возможные ответы на приглашение CONTINUE:

*YES* - проигнорировать остаток блоков в этом индексном дескрипторе файла и продолжить проверку со следующего индексного дескриптора файла в файловой системе. Эта исключительная ситуация предупреждает полный контроль файловой системы. Для повторной проверки этой файловой системы следует прогнать fsck второй раз.

*NO* - завершить программу.

*Сообщение:*

BAD STATE DDD TO BLKERR

Внутренняя ошибка зашифровала карту состояния fsck, как имеющую невозможное значение DDD. fsck немедленно завершает работу. Если это происходит, обратитесь к специалисту.

*Сообщение:*

B DUP I-I

Индексный дескриптор файла I содержит блок номер B, который уже заявлен другим индексным дескриптором файла. Эта исключительная ситуация может порождать на этапе 1 сообщение об ошибках EXCESSIVE DUP BLKS, если у индексного дескриптора файла I слишком много блоков, заявленных другими индексными дескрипторами файлов. Эта исключительная ситуация активизирует этап 1B и порождает на этапах 2 и 4 сообщение об ошибках BAD/DUP.

*Сообщение:*

BAD MODE: MAKE IT A FILE?

Это сообщение возникает, когда состояние данного индексного дескриптора файла устанавливается на все единицы, указывая на повреждение файловой системы. Это сообщение не указывает на повреждение диска, если оно появляется многократно после прогона fsck -y. Ответ y заставляет команду fsck заново инициализировать индексный дескриптор файла до разумного значения.

*Сообщение:*

EXCESSIVE DUP BLKS I-I (CONTINUE)

Слишком много блоков (обычно больше 10) заявлено другими индексными дескрипторами файлов.

Возможные ответы на приглашение CONTINUE:

*YES* - проигнорировать остаток блоков в этом индексном дескрипторе файла и продолжить проверку со следующего индексного дескриптора файла в файловой системе. Эта исключительная ситуация предупреждает полный контроль файловой системы. Для повторной проверки файловой системы следует второй раз прогнать команду fsck.

*NO* - завершить программу.

*Сообщение:*

DUP TABLE OVERFLOW (CONTINUE)

Во внутренней таблице в fsck, содержащей числа идентичных блоков, больше нет места.

Возможные ответы на приглашение CONTINUE:

*YES* - продолжить программу. Эта исключительная ситуация предупреждает полный контроль файловой системы. Для повторной проверки файловой системы необходимо второй раз прогнать fsck. Если найден другой идентичный блок, это сообщение об ошибке повторяется;

*NO* - завершить программу.

*Сообщение:*

PARTIALLY ALLOCATED INODE I-I (CLEAR)

Индексный дескриптор файла I ни выделен, ни освобожден.

Возможные отклики на приглашение CLEAR:

*YES* - освободить индексный дескриптор файла I путем обнуления его содержимого;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

INCORRECT BLOCK COUNT I-I (X should be Y) (CORRECT)

Подсчет блоков для индексного дескриптора файла I - X блоков, а должно быть Y блоков. При очистке подсчет корректируется.

Возможные ответы на приглашение CORRECT:

*YES* - заменить подсчет блоков индексного дескриптора файла I на Y; проигнорировать эту исключительную ситуацию.

#### *2.4.4.3. Этап 1B: Повторный поиск DUPS*

Когда в файловой системе обнаруживается идентичный блок, эта файловая система повторно просматривается с целью поиска того индексного дескриптора файла, который до этого заявил об этом блоке. Когда будет найден идентичный блок, появится следующее информационное сообщение:

*Сообщение:*

B DUP I-I

Индексный дескриптор файла I содержит блок номер B, о котором было уже заявлено другим индексным дескриптором файла. Эта исключительная ситуация порождает на этапе 2 сообщение об ошибках BAD/DUP. Можно определить, у каких индексных дескрипторов файлов есть перекрывающиеся блоки, путем изучения этой исключительной ситуации и исключительной ситуации DUP на этапе 1.

#### *2.4.4.4. Этап 2: Проверка полного имени пути*

На этом этапе удаляются элементы каталога, указывающие на "плохие" индексные дескрипторы файлов, найденные на этапах 1 и 1B. Даются отчеты об исключительных ситуациях, появившихся в результате:

* неверного состояния корневого индексного дескриптора файла;
* указатели каталога индексных дескрипторов файлов за пределами файла;
* элементы каталога указывают на "плохие" индексные дескрипторы файлов;
* контроля целостности каталога;

Все ошибки на этом этапе считаются фатальными, если очищается файловая система, за исключением каталогов не кратных размеру блоков и посторонних прочных связей.

Этап 2 - Сообщения об ошибках.

*Сообщение:*

ROOT INODE UNALLOCATED (ALLOCATE)

Корневой индексный дескриптор (обычно это индексный дескриптор номер 2) не имеет выделенных битов режима.

Возможные ответы на приглашение ALLOCATE:

*YES* - выделить индексный дескриптор 2 как корневой. Файлы и каталоги обычно находящиеся в корне, восстанавливаются на этапе 3 и помещаются в каталог lost+found. Если попытка выделить корень не удается, тогда fsck выходит с сообщением:

CANNOT ALLOCATE ROOT INODE

*NO* - завершить программу.

*Сообщение:*

ROOT INODE NOT DIRECTORY (REALLOCATE)

Корневой индексный дескриптор (обычно это индексный дескриптор номер 2) файловой системы не является индексным дескриптором каталога.

Возможные отклики на приглашение REALLOCATE:

*YES* - очистить содержимое корневого индексного дескриптора и освободить его. Файлы и каталоги, обычно находящиеся в корне, восстанавливаются на этапе 3 и помещаются в каталог lost+found. Если попытка выделить корень неудачна, fsck выйдет с сообщением:

CANNOT ALLOCATE ROOT INODE

*NO* - команда fsck делает приглашение с FIX.

Возможные ответы на приглашение FIX:

*YES* - изменить тип корневого индексного дескриптора на каталог. Если блоки данных корневого индексного дескриптора не являются блоками каталога, порождается много сообщений об ошибках.

*NO* - завершить программу.

*Сообщение:*

DUPS/BAD IN ROOT INODE (REALLOCATE)

На этапе 1 или этапе 1B были найдены идентичные блоки или "плохие" блоки в корневом индексном дескрипторе (обычно это индексный дескриптор номер 2) файловой системы.

Возможные отклики на приглашение REALLOCATE:

*YES* - очистить содержание корневого индексного дескриптора и освободить его. Файлы и каталоги, обычно находящиеся в корне, восстанавливаются на этапе 3 и помещаются в каталог lost+found. Если попытка выделить корень неудачна, fsck выходит с сообщением:

CANNOT ALLOCATE ROOT INODE

*NO* - команда fsck выдает приглашение CONTINUE.

Возможные ответы на приглашение CONTINUE:

*YES* - проигнорировать исключительную ситуацию DUPS/BAD в корневом индексном дескрипторе и попытаться продолжить проверку файловой системы. Если корневой индексный дескриптор неверный, это может порождать много других сообщений об ошибках.

*NO* - завершить программу.

*Сообщение:*

NAME TOO LONG F

Было найдено чрезвычайно длинное полное имя пути. Обычно этим обозначаются циклы в пространстве имен файловой системы. Это может возникать, если привилегированный пользователь осуществил циклические связи с каталогами. Эти связи можно удалить.

*Сообщение:*

I OUT OF RANGE I-I NAME-F (REMOVE)

Элемент каталога F имеет индексный дескриптор файла номер I, который больше максимально возможного индексного дескриптора файла.

Возможные ответы на приглашение REMOVE:

*YES* - удалить элемент каталога F;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

UNALLOCATED I-I OWNER-O MODE-M SIZE-S MTIME-T TYPE-F

(REMOVE)

Каталог или элемент файла F указывает на невыделенный индексный дескриптор файла I. Печатаются: владелец O, режим M, размер S, время модификации T и имя F.

Возможные отклики на приглашение REMOVE:

*YES* - удалить элемент каталога F;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

DUP/BAD I-I OWNER-O MODE-M SIZE-S MTIME-T TYPE-F

(REMOVE)

На этапе 1 или этапе 1B найдены идентичные блоки или "плохие" блоки, связанные с каталогом или элементом каталога F. Печатаются: владелец O, режим M, размер S, время модификации T и имя каталога F.

Возможные отклики на приглашение REMOVE:

*YES* - удалить элемент каталога F;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

ZERO LENGTH DIRECTIRY I-I OWNER-O MODE-M SIZE-S

MTIME-T DIR-F (REMOVE)

Элемент каталога F имеет размер S, который равен 0. Печатаются: владелец O, режим M, размер S, время модификации T и имя каталога F.

Возможные отклики на приглашение REMOVE:

*YES* - удалить элемент каталога F; это порождает на этапе 4 сообщение об ошибках BAD/DUP;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

DIRECTORY TOO SHORT I-I OWNER-O MODE-M SIZE-S MTIME-T

DIR-F (FIX)

Был найден каталог F, размер которого S меньше минимального размера каталога. Печатаются: владелец O, режим M, размер S, время модификации T и имя каталога F.

Возможные ответы на приглашение FIX:

*YES* - увеличить размер каталога до размера минимального каталога;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

DIRECTORY F LENGTH S NOT MULTIPLE OF B (ADJUST)

Был найден каталог F с размером S, не кратным размеру B блока каталога.

Возможные отклики на приглашение ADJUST:

*YES* - округлить длину до соответствующего размера блока. При очистке файловой системы печатается только предупреждение и корректируется каталог;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

DIRECTORY CORRUPTED I-I OWNER-O MODE-M SIZE-S MTIME-T

DIR-F (SALVAGE)

Был найден каталог с несогласованным внутренним состоянием.

Возможные отклики на приглашение SALVAGE:

*YES* - отбросить все элементы до следующей границы каталога (обычно это 512-байтовая граница). Можно отбросить до 42 элементов; это следует делать только после того, как другое восстановление окажется неудачным;

*NO* - перейти к следующей границе каталога и продолжить считывание, но не модифицировать каталог.

*Сообщение:*

BAD INODE NUMBER FOR '.' I-I OWNER-O MODE-M SIZE-S

MTIME-T DIR-F (FIX)

Был найден каталог I, номер индексного дескриптора которого для '.' не равен I.

Возможные отклики на приглашение FIX:

*YES* - изменить номер индексного дескриптора файла для '.' так, чтобы он был равен I;

*NO* - оставить неизменным номер индексного дескриптора файла для '.'.

*Сообщение:*

MISSING `.' I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F (FIX)

Был найден каталог I, первый элемент которого не выделен.

Возможные ответы на приглашение FIX:

*YES* - построить элемент для `.' с номером индексного дескриптора файла, равным I;

*NO* - оставить каталог неизменным.

*Сообщение:*

MISSING `.' I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F

CANNOT FIX, FIRST ENTRY IN DIRECTORY CONTAINS F

Был найден каталог I, первый элемент которого F. Команда fsck не может решить эту проблему. Должна быть смонтирована файловая система, а элемент F перемещен куда либо. Затем файловая система должна быть демонтирована и снова прогоняется команда fsck.

*Сообщение:*

MISSING `.' I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F

CANNOT FIX, INSUFFICIENT SPACE TO AND `.'

Был найден каталог I, первый элемент которого не `.'. Это никогда не должно происходить. Команда fsck не может решить эту проблему. Если это все же случается, обратитесь к квалифицированному специалисту.

*Сообщение:*

EXTRA `.' ENTRY I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F

(FIX)

Был найден каталог I, первый элемент которого не `.'.

Возможные ответы на приглашение FIX:

*YES* - удалить дополнительный элемент для `.';

*NO* - оставить каталог неизменным.

*Сообщение:*

BAD INORE NUMBER FOR `.' I=I OWNER=O MODE=M SIZE=S MTIME=T

DIR=F (FIX)

Был найден каталог I, номер индексного дескриптора которого для `.' не равен владельцу I.

Возможные отметки на приглашение FIX:

*YES* - Изменить номер индексного дескриптора для `.' так, чтобы он был равен владельцу I. (Обратите внимание, что ".." в корневом индексном дескрипторе указывает на себя).

*NO* - оставить номер индексного дескриптора для '..' неизменным.

*Сообщение:*

MISSING `..' I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F (FIX)

Был найден каталог I, второй элемент которого не выделен.

Возможные отметки на приглашение FIX:

*YES* - построить элемент для '.' с номером индексного дескриптора файла, равным владельцу I. (Обратите внимание, что ".." в корневом индексном дескрипторе файла указывает сам на себя);

*NO* - оставить каталог не измененным.

*Сообщение:*

MISSING `..' I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F

CANNOT FIX, SECOND ENTRY IN DIRECTORY CONTAINS F

Был найден каталог I, второй элемент которого F. Команда fsck не может решить эту проблему. Должна быть смонтирована файловая система и куда-нибудь перемещен элемент F. Затем следует демонтировать файловую систему и снова прогнать fsck.

*Сообщение:*

MISSING `..' I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F

CANNOT FIX, INSUFFICIENT SPACE TO ADD `..'

Был найден каталог I, второй элемент которого не '..' (каталог владельца).

MISSING `.' I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F

CANNOT FIX, INSUFFICIENT SPACE TO ADD `.'

Был найден каталог I, первый элемент которого не '.' Это никогда не должно происходить. Команда fsck не может решить эту проблему. Если это все же случается, обратитесь к квалифицированному специалисту.

*Сообщение:*

MISSING `..' I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F (FIX)

Был найден каталог I, второй элемент которого не выделен.

Возможные ответы на приглашение FIX:

*YES* - Построить элемент для `..' с номером индексного дескриптора, равным I. (Обратите внимание, что ".." в корневом индексном дескрипторе указывает сам на себя);

*NO* - оставить каталог неизменным.

*Сообщение:*

MISSING `..' I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F

CANNOT FIX, SECOND ENTRY IN DIRECTORY CONTAINS F

Был найден каталог I, второй элемент которого не '..' (каталог владельца). Команда fsck не может решить эту проблему. Должна быть смонтирована файловая система, а второй элемент в каталоге куда-нибудь перемещен. Затем файловая система должна быть демонтирована и снова выполнена команда fsck.

*Сообщение:*

EXTRA `..' ENTRY I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F

(FIX)

Был найден каталог I, у которого больше одного элемента для `..'(каталог владельца).

Возможные ответы на приглашение FIX:

*YES* - удалить дополнительный элемент для `..' (каталог владельца);

*NO* - оставить каталог неизменным.

*Сообщение:*

N IS AN EXRANEONOUS HARD LINK TO A DIRECTORY D (REMOVE)

С помощью команды fsck была обнаружена прочная связь N с каталогом D. При очистке посторонние связи игнорируются. Возможные ответы на приглашение REMOVE:

*YES* - удалить посторонний элемент N;

*NO* - проигнорировать исключительную ситуацию.

*Сообщение:*

BAD INODE S TO DESCENT

Внутренняя ошибка привела к тому, что невозможное состояние S перешло к программе, которая сократила структуру каталога файловой системы. Команда fsck завершается. Если такое случается, обратитесь к специалисту.

*Сообщение:*

BAD RETURN STATE S FROM DESCEND

Внутренняя ошибка привела к тому, что невозможное состояние S было возвращено к программе, сокращающей структкру каталога файловой системы. Команда fsck завершается. Если такое случается, обратитесь к специалисту.

*Сообщение:*

BAD STATE S FOR ROOT INODE

Внутренняя ошибка привела к тому, что невозможное состояние S было присвоено корневому индексному дескриптору файла. Команда fsck завершается. Если такое случается, обратитесь к специалисту.

#### *2.4.4.5. Этап 3: Проверка связности*

На этом этапе проверяются каталоги, изученные на втором этапе. Дается отчет об исключительных ситуациях, являющихся результатом:

* каталогов без ссылок;
* пропущенных или заполненных lost+found каталогов.

Этап 3: Сообщения об ошибках.

*Сообщение:*

UNREF DIR I=I OWNER=O MODE=M SIZE=S MTIME=T (RECONNECT)

Индексный дескриптор I каталога не был соединен с элементом каталога при прослеживании файловой системы. Печатаются владелец O, режим M, размер S и время модификации T индексного дескриптора I каталога. При очистке каталог заново соединяется, если его размер не нулевой; иначе он очищается.

Возможные ответы на приглашение RECONNECT:

*YES* - заново соединить индексный дескриптор I каталога с файловой системой в каталоге для потерянных файлов (обычно это каталог lost+found). Это может порождать на этапе 3 сообщения об ошибках lost+found, если есть проблемы в соединении индексного дескриптора I каталога с каталогом lost+found. Это также может порождать на этапе 3 сообщение об ошибке CONNECTED, если связь была удачной;

*NO* - проигнорировать эту исключительную ситуацию. Это порождает на этапе 4 сообщение об ошибке UNREF.

*Сообщение:*

NO lost+found DIRECTORY (CREATE)

В корневом каталоге файловой системы нет каталога lost+found. При очистке команда fsck пытается создать каталог lost+found.

Возможные ответы на приглашение CREATE:

*YES* - создать каталог lost+found в корне файловой системы.

Это может порождать сообщение:

NO SPACE LEFT IN / (EXPAND)

Ниже приведены возможные ответы. Невозможность создать каталог lost+found порождает сообщение:

SORRY. CANNOT CREATE lost+found DIRECTORY

и срывает попытку подсоединить индексный дескриптор файла. Это порождает на этапе 4 сообщение об ошибке UNREF;

*NO* - прекратить попытку связать потерянный индексный дескриптор файла. Это порождает на этапе 4 сообщение об ошибке UNREF.

*Сообщение:*

lost+found IS NOT A DIRECTORY (REALLOCATE)

Элемент для lost+found не является каталогом.

Возможные ответы на приглашение REALLOCATE:

*YES* - выделить индексный дескриптор файла каталога и изменить lost+found, чтобы обратится к нему. Предыдущий индексный дескриптор файла, к которому обратился каталог lost+found, не очищен. Таким образом, либо о нем снова заявляют как об индексном дескрипторе файла UNREF, либо позднее на этом этапе корректируется подсчет его связей. Невозможность создать каталог lost+found порождает сообщение:

SORRY. CANNOT CREATE lost+found DIRECTORY

и прерывает попытку связать потерянный индексный дескриптор файла. Это, в свою очередь, порождает на этапе 4 сообщение об ошибке UNREF;

*NO* - прервать попытку связать потерянный индексный дескриптор файла. Это порождает на этапе 4 сообщение об ошибке UNREF.

*Сообщение:*

NO SPACE LEFT IN / lost+found (EXPEND)

В корнвом каталоге файловой системы нет места, чтобы добавить другой элемент к каталогу lost+found. При очистке каталог lost+found расширяется.

Возможные ответы на приглашение EXPAND:

*YES* - расширить каталог lost+found, чтобы найти место для нового элемента. Если попытка расширения неудачна, команда fsck печатает сообщение:

*Сообщение:*

SORRY. NO SPACE IN lost+found DIRECTORY

и прерывает попытку соединить потерянный индексный дескриптор файла. Это, в свою очередь, попрождает на этапе 4 сообщение об ошибке UNREF. Очистить ненужные элементы в каталоге lost+found. Эта ошибка является фатальной если очищается файловая система;

*NO* - прервать попытку связать потерянный индексный дескриптор файла. Это порождает на этапе 4 сообщение об ошибке UNREF.

*Сообщение:*

DIR I=I1 CONNECTED. PARENT WAS I=I2

Это справочное сообщение, указывающее, что индексный дескриптор файла I1 каталога был успешно соединен с каталогом lost+found. Родительский индексный дескриптор файла I2 индексного дескриптора файла I1 замещается номером индексного дескриптора каталога lost+found.

*Сообщение:*

DIRECTORY F LENGTH 5 NOT MULTIPLE OF S (ADJUST)

Был найден каталог F с размером S, не кратным размеру B блока каталога. (Учтите, что это может возникнуть снова на этапе 3, если исключительная ситуация не будет исправлена на этапе 2).

Возможные ответы на приглашение ADJUST:

*YES* - округлить длину до соответствующего размера блока. При очистке файловой системы печается только предупреждение и корректируется каталог;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

BAD INODE S TO DESCEND

Внутренняя ошибка привела к тому, что невозможное состояние S перешло к программе, сокращающей структуру каталога файловой системы. Команда fsck завершается. Если это случится, обратитесь к специалисту.

#### *2.4.4.6. Этап 4: Проверка подсчета ссылок*

На этом этапе проверяется информация о подсчете связей, полученная на этапах 2 и 3. Дается отчет об исключительных ситуациях, возникших в результате наличия:

* файлов без ссылок;
* отсутствующего или заполненного каталога lost+found;
* неправильного подсчета связей для файлов, каталогов, символьных связей или специальных файлов;
* файлов без ссылок, символьных связей и каталогов;
* неудачных или идентичных блоков в файлах, символьных связях и каталогах.

Все ошибки на этом этапе (за исключением работы в каталоге lost+found) можно исправить, если будет очищена файловая система.

Этап 4. Сообщения об ошибках.

*Сообщение:*

UNREF FILE I=I OWNER=O MODE=M SIZE=S MTIME=T

(RECONNECT)

Индексный дескриптор файла I не был соединен с элементом каталога при проверке файловой системы. Печатаются: владелец O, режим M, размер S и время модификации T индексного дескриптора файла I. Файл очищается, когда либо его размер, либо подсчет его связей нулевые; в противном случае он заново соединяется.

Возможные ответы на приглашение PROMPT:

*YES* - заново соединить индексный дескриптор файла I с файловой системой в каталоге для потерянных файлов (обычно это каталог lost+found). Это может порождать на этапе 4 сообщение об ошибках lost+found, если есть проблемы соединения индексного дескриптора файла I с каталогом lost+found;

*NO* - проигнорировать эту исключительную ситуацию. Это всегда вызывает ситуацию CLEAR на этапе 4.

*Сообщение:*

(CLEAR)

Индексный дескриптор файла, упомянутый в только что напечатанном сообщении об ошибках, невозможно снова соединить. Это сообщение не может появиться, если очищается файловая система, потому что отсутствие пространства для повторного соединения файлов является фатальной ошибкой.

Возможнные ответы на приглашение CLEAR:

*YES* - освободить индексный дескриптор файла путем обнуления его содержания;

*NO* - проигнорировать эту ситуацию.

*Сообщение:*

NO lost+found DIRECTORY (CREATE)

В корневом каталоге файловой системы нет каталога lost+found. При очистке команда fsck пытается создать каталог lost+found.

Возможные ответы на приглашение CREATE:

*YES* - создать каталог lost+found в корне файловой системы. Это может порождать следующее сообщение:

NO SPACE LEFT IN / (EXPAND)

Невозможность создать каталог lost+found порождает сообщение:

SORRY. CANNOT CREATE lost+found DIRECTORY

и прерывает попытку подсоединить потерянный индексный дескриптор файла. Это в свою очередь порождает на этапе 4 сообщение об ошибках UNREF.

*NO* - прекратить попытку присоединить потерянный дескриптор файла. Это порождает на этапе 4 сообщение об ошибках UNREF.

*Сообщение:*

lost+found IS NOT A DIRECTORY (REALLOCATE)

Элемент для lost+found не является каталогом.

Возможные ответы на приглашение REALLOCATE:

*YES* - выделить индексный дескриптор файла каталога и изменить каталог lost+found для ссылки на него. Предыдущая ссылка на индексный дескриптор файла в каталоге lost+found не очищается. Таким образом, о нем либо снова заявлено как об индексном дескрипторе файла UNREF, либо подсчет его связей корректируется (ADJUST) позднее на этом этапе. Невозможность создать каталог lost+found порождает сообщение:

SORRY. CANNOT CREATE lost+found DIRECTORY

и прерывает попытку подсоединить потерянный индексный дескриптор файла. Это порождает на этапе 4 сообщение об ошибках UNREF.

*NO* - прекратить попытку присоединить потерянный индексный дескриптор файла. Это порождает на 4 этапе сообщение об ошибках UNREF.

*Сообщение:*

NO SPACE LEFT IN /lost+found (EXPAND)

В корневом каталоге файловой системы нет места для добавления другого элемента к каталогу lost+found. Каталог lost+foun расширяется.

Возможные ответы на приглашение EXPAND:

*YES* - расширить каталог lost+found, чтобы найти место для нового элемента. Если попытка расширения неудачна, команда fsck распечатывает сообшение:

SORRY. NO SPACE IN lost+found DIRECTIRY

и прерывает попытку присоединить потерянный индексный дескриптор файла. Это порождает на этапе 4 сообщение об ошибках UNREF. Очистить ненужные элементы в каталоге lost+found. Эта ошибка фатальная, если очищается файловая система.

*NO* - прекратить попытку присоединить потерянный дескриптор файла. Это порождает на этапе 4 сообщение об ошибках UNREF.

*Сообщение:*

LINK COUNT TYPE I=I OWNER=O MODE=M SIZE=S MTIME=T

COUNT=X

SHOULD BE Y (ADJUST)

Подсчет связей для индексного дескриптора файла I - X, а должен быть Y. Печатаются: владелец O, режим M, размер S, время модификации T. При очистке подсчет связей корректируется, если не возрастает число ссылок; условие, которое не должно никогда возникать, если не усугубляется ошибкой в технических средствах. Когда число ссылок возрастает во время очистки, команда fsck завершается с сообщением:

LINK COUNT INCREASING

Возможные ответы на приглашение ADJUST:

*YES* - заменить подсчет связей индексного дескриптора файла I на Y;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

UNREF TYPE OWNER-O MODE-M SIZE-S MTIME-T (CLEAR)

Индексный дескриптор файла I не был присоединен к элементу каталога при проверке файловой системы. Печатаются: владелец O, режим M, размер S и время модификации T индексного дескриптора файла I. Поскольку этот файл не был присоединен из-за нулевых размера или подсчета связей он очищается.

Возможные ответы на приглашение CLEAR:

*YES* - освободить индексный дескриптор файла I путем обнуления его содержимого;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

BAD/DUP TYPE OWNER-O MODE-M SIZE-S MTIME-T (CLEAR)

На этапе 1 или 1B были обнаружены идентичные или неудачные блоки, связанные с индексным дескриптором файла I. Печатаются: владелец O, режим M, размер S и время модификации T индексного дескриптора файла I. Это сообщение не может появится при очистке файловой системы, поскольку оно ранее вызвало фатальную ошибку.

Возможные ответы на приглашение CLEAR:

*YES* - освободить индексный дескриптор файла I путем обнуления его содержимого; проигнорировать эту исключительную ситуацию.

#### *2.4.4.7. Этап 5: Проверка групп цилиндров*

На этом этапе проверяются свободные блоки и используемые таблицы индексных дескрипторов файлов. Даются отчеты об исключительных ситуациях, возникших в результате появления:

* выделенных блоков в таблицах свободных блоков;
* свободных блоков, отсутствующих в таблицах свободных блоков;
* неверного общего подсчета свободных блоков;
* свободных индексных дескрипторов файлов в используемых таблицах индексных дескрипторов файлов;
* выделенных индексных дескрипторов файлов, отсутствующих в используемых таблицах индексных дескрипторов файлов;
* неверного общего подсчета используемых индексных дескрипторов файлов.

Сообщения об ошибках.

*Сообщение:*

CG C: BAD MAGIC NUMBER

Магическое число группы цилиндров C неверное. Обычно это указывает на то, что таблицы групп цилиндров были разрушены. При диалоговом режиме работы группа цилиндров помечается как требующая восстановления. При очистке файловой системы эта ошибка считается фатальной.

*Сообщение:*

BLK (S) MISSING IN BYT MAPS (SALVAGE)

В таблице блоков групп цилиндров отсутствуют некоторые свободные блоки. Во время очистки таблицы восстанавливаются.

Возможные ответы на приглашение SALVAGE:

*YES* - восстановить таблицу свободных блоков;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

SUMMARY INFORMATION BAD (SALVAGE)

Итоговая информация оказалась неверной. При очистке итоговая информация проверяется еще раз.

Возможные ответы на приглашение SALVAGE:

*YES* - восстановить итоговую информацию;

*NO* - проигнорировать эту исключительную ситуацию.

*Сообщение:*

FREE BLK COUNT (S) WRONG IN SUPERBLOCK (SALVAGE)

Информация свободного блока системного блока тома оказалась неверной. При очистке информация свободного блока системного блока тома проверяется еще раз.

Возможные ответы на приглашение SALVAGE:

*YES* - восстановить информацию свободного блока системного блока тома; проигнорировать эту исключительную ситуацию.

#### *2.4.4.8. Этап очистки*

При проверке файловой системы выполняется несколько функций очистки. Далее перечисляются справочные сообщения о файловой системе и измененном состоянии файловой системы.

*Сообщение:*

V files W used X free (Y flags, & blocks)... Q %

fragmentation

Это справочное сообщение указывает на то, что проверяемая файловая система содержала V файлов, использующих блоки с размером фрагментов W, оставляя блоки с размером фрагментов X свободными в файловой системе. Числа в круглых скобках разбивают свободный подсчет на Y свободных фрагментов и Z свободных блоков с полным размером. Наконец, эти числа используются для разложения процентного содержания фрагментации файловой системы (Q).

*Сообщение:*

\*\*\*\*\* REBOOT THE SYSTEM \*\*\*\*\*

Это справочное сообщение указывает на то, что корневая файловая система root была модифицирована с помощью команды fsck. Если вы сразу же не выполняете повторную начальную загрузку, работа, проделанная командой fsck, может быть невыполненной из-за использования копий таблиц, находящихся в оперативной памяти.

*Сообщение:*

\*\*\*\*\* FILE SYSTEM WAS MODIFIED \*\*\*\*\*

Это справочное сообщение указывает на то, что текущая файловая система была модифицирована с помощью команды fsck. Если монтируется эта файловая система или если она является текущей корневой файловой системой root, команда fsck должна быть остановлена или произведена повторная начальная загрузка.

#### 2.4.5. Сохранение надежности корневой файловой системы root

Ниже предлагаются способы сохранения надежности и эффективности функционирования корневой файловой системы root.

#### *2.4.5.1. Очистка свободного пространства в корне*

Для корневой файловой системы выделен определенный и окончательный объем пространства на диске. Незанятое пространство на диске в этой области, называемое свободным пространством, допускает наличие дополнительных и временных файлов и часто служит сверхоперативной памятью для определенных программ системы. Системное администрирование и другие типы программ требуют однократного выполнения корневой файловой системы root на свободном пространстве. Рекомендуется избегать использования всего пространства в корневой файловой системе root. Если выполнение выходит за пределы пространства в корне, будет отображено следующее сообщение:

no space on Fixed Disk Device 0x1

Когда вы увидите это сообщение, следует вручную удалить ненужные файлы из корневой файловой системы root. Поскольку система создает файл /etc/mnttab во время запуска, рекомендуется сохранить, по крайней мере, 10 свободных блоков в корневой файловой системе root до отключения машины. Можно использовать команду df с параметром файла, чтобы узнать сколько свободных блоков имеется в ваших файловых системах.

#### *2.4.5.2. Сохранение последовательности новых файлов в корне*

Система UNIX считывает файлы и работает с ними быстрее, если они даны в последовательности. Изначально упорядочивается свободный список корневой файловой системы, так что новые файлы даются в последовательности, но создание файлов и /или их удаление могут дезорганизовать свободный список. Имеются автоматизированные механизмы, которые позволяют периодически перестраивать свободный список корневой файловой системы. Если в вашей машине запускаются другие активные файловые системы, периодическое выполнение команды fsck -s в них при их демонтировании повышает эффективность работы диска.

### 3. Установка характеристик системы

### 3.1. Повторная установка даты и времени

3.1.1. Использование меню OA&M   
3.1.2. Использование команд shell

Для доступа к меню установки характеристик системы используется параметр system\_setup в главном меню System Administration в системе UNIX System V Release 4.0 Version 1.0.

#### 3.1.1. Использование меню OA&M

Для повторной установки даты и времени используйте следующие меню:

1. Выберите datetime из меню System Name, Date/Time and Initial Password Setup.
2. Выберите set из меню Display and Set System Date and Time и нажмите ENTER.

Система отображает экран Set System Date and Time Information:

4 Set System Date and Time Information

--------------------------------------------------------

Date: December 15, 1989

Time: 05:15

AM/PM: PM

Timezone: EST

Is daylight savings time in effect?: yes

Please Note: This task should be done in single-user

mode.

Warning: Setting the date/time while the system is in

multi-user state can be dangerous.

1. Введите информацию о дате и времени, затем нажмите SAVE.

#### 3.1.2. Использование команд shell

Установите заново дату и время при входе в систему как root, используя команду date следующего формата:

MMddhhmmyy

где MM - месяц;   
dd - день;   
hh - часы;   
mm - минуты;   
yy - год.

Например, чтобы заново установить дату 10 мая 1990 г. время 9:12 напечатайте:

date 0510091290

### 3.2. Установка имени системы и имени узла

3.2.1. Установка имен систем через меню OA&M   
3.2.2. Использование команд shell

Имя вашей системы может быть в длину до 8 символов и должно отличаться от имен систем в других вычислительных машинах, с которыми вы обмениваетесь информацией.

Если ваша машина является частью действующей сети и вы изменяете имя узла системы, необходимо попросить других пользователей сети заменить имя информационных файлов сети на новое.

#### 3.2.1. Установка имен систем через меню OA&M

Чтобы установить имя системы и имя узла выполните следующие меню:

1. Выберите nodename из меню System Name, Date/Time and Initial Password Setup.
2. Выберите set из меню Display and Set System Name and Network Node Name of the Machine и нажмите ENTER.

Система отображает экран Set System Name and Network Node Name of the Machine:

4 Set System Name and Network Node Name of the Machine

---------------------------------------------------------

System Name: Summit

Network Node Name: SF1

Warning: (Re) setting the network node name if the

machine is in a network can be dangerous.

1. Введите имя системы и /или имя узла, затем нажмите SAVE.

Система отображает экран с целью сообщения пользователям измененного имени.

1. Нажмите CONT, чтобы вернуться к экрану 4, или CANCEL, чтобы вернуться к экрану 3.

#### 3.2.2. Использование команд shell

Команда uname используется для установки имени системы. Эту команду могут выполнять только привилегированные пользователи. Следующая команда, выполняемая суперпользователем root, присваивает вашей системе имя mysys:

### 4. Управление портами

### 4.1. Управление средством доступа к сервису (SAF)

4.1.1. Контроллер доступа к сервису   
4.1.2. Файл конфигурации системы.   
4.1.3. Командные файлы конфигурации мониторов порта   
4.1.4. Командные файлы конфигурации сервиса   
4.1.5. Управляющий файл SAC   
4.1.6. Управляющий файл монитора порта   
4.1.7. Установка связи с RS-232

После добавления сетевого сервиса к UNIX System V появились нестандартные методы доступа к сервису системы, давшие непоследовательные интерфейсы и среду выполнения, затрудняющие работу пользователей. SAF (средство доступа к сервису) обеспечивает механизм унифицированного доступа к сервису. Управляющие компоненты этого средства - это команды инсталяции, создания конфигурации и поддержки мониторов порта и сервиса, а также файлов, в которых хранится сервисная информация и информация монитора порта.

Способ управления и организации монитором порта доступа к порту зависит от конкретного монитора порта, а не отдельного компонента SAF. Следовательно, пользователи могут расширять свои системы, разрабатывая и устанавливая свои собственные мониторы порта. Пользователям, которые хотят написать собственные мониторы порта, следует обратиться к книге "Руководство программиста: Сетевые интерфейсы". В этом разделе описание конкретных мониторов порта ограничено теми программами, которые поставляются вместе с системой UNIX, ttymon и "приемником".

С точки зрения SAF сервис - это процесс, который запускается. Ограничений функций, которые предоставляет сервис, не существует.

SAF состоит из управления процессом - контроллера доступа к сервису (SAC), а также двух управляющих уровней, соответствующих двум уровням в поддерживаемой структуре каталогов. Верхний управляющий уровень соотносится с управлением монитором порта, нижний уровень - с управлением сервисом.

SAF состоит из следующих компонентов:

* контроллер доступа к сервису;
* командный файл конфигурации системы;
* управляющий файл SAC;
* команда администратора SAC sacadm;
* мониторы портов;
* необязательные файлы конфигурации мониторов портов;
* управляющий файл для каждого монитора порта;
* команда администратора pmadm;
* необязательные файлы конфигурации сервиса.

В этом разделе описываются SAC, управляющие файлы, а также файлы конфигурации системы, мониторов порта и сервиса.

#### 4.1.1. Контроллер доступа к сервису

Контроллер доступа к сервису (SAC) управляет механизмом обслуживания. Это процесс управления средством доступа к сервису (SAF). SAC запускается с помощью команды init (1M) посредством входа в /sbin/inittab. Его функция - поддерживать мониторы порта системы в состоянии, которое вы указываете. Эти состояния включают: STARTING, ENABLED, DISABLED, STOPPING, NOTRUNNING и FAILED (Монитор порта вводит состояние FAILED, когда SAC не может запустить его после указанного числа попыток).

Команда администратора sacadm используется для того, чтобы сообщить SAC о необходимости изменить состояние монитора порта, sacadm также можно использовать для добавления или удаления монитора порта из области управления SAC и просмотра информации о мониторах портов, известных SAC.

Управляющий файл SAC содержит уникальный тег для каждого монитора порта, известного SAC, и полное имя команды, используемой для запуска каждого монитора порта.

SAC:

* настраивает свою собственную среду;
* запускает требуемые мониторы портов;
* опрашивает свои мониторы порта и инициирует процедуры восстановления при необходимости.

При инициации SAC настраивает свою собственную среду запуска командного файла конфигурации системы. Затем он считывает свой управляющий файл, чтобы определить те мониторы портов, которые следует запустить. Для каждого монитора порта, который он запускает, он интерпретирует файл конфигурации этого монитора порта, если таковой существует. Наконец мониторы портов, указанные в управляющем файле (например, ttymon) запускаются.

После запуска мониторов портов SAC периодически опрашивает их для получения информации о состоянии. Параметр командной строки sac (1M), -t, позволяет администратору системы управлять частотой опроса. Когда монитор порта получает запрос о состоянии из SAC, он должен ответить сообщением, содержащим его текущее состояние (например, ENABLED). Если SAC не получит ответа, он предполагает, что монитор порта не запущен. Если монитор порта, который должен прогоняться, остановился, SAC предполагает, что он допустил сбой и предпринимает требуемое действие восстановления.

SAC повторно запустит монитор порта с отказом, если для этого монитора порта был указан ненулевой счетчик повторного запуска, когда он создавался.

SAC представляет собой административную точку управления для всех мониторов портов, (а, следовательно, для всех портов системы). Команды администратора sacadm (1M) и pmadm (1M) передают запросы в SAC, который в свою очередь устанавливает связь с мониторами портов. Эти запросы включают разрешение запрещенного монитора порта, и тот начинает прием запросов сервиса в свои порты, запуск мониторов портов, которые до этого были остановлены, а также просмотр текущего состояния всех мониторов порта системы.

#### 4.1.2. Файл конфигурации системы

Файл конфигурации системы - /etc/saf/\_sysconfig - поставляется пустым. Его может использовать администратор системы, чтобы настроить среду для всего сервиса системы, написав командный файл на интерпретированном языке. Командный файл конфигурации системы интерпретируется контроллером доступа к сервису после запуска SAC. SAC запускается, когда система переходит в многопользовательский режим.

#### 4.1.3. Командные файлы конфигурации мониторов порта

Командные файлы конфигурации мониторов порта (/etc/saf/pmtag/\_config) являются необязательными. Они позволяют администратору системы настраивать среду на любой заданный монитор порта и сервис, который есть в наличии, благодаря точкам доступа, за которые этот монитор порта несет ответственность. Командные файлы конфигурации мониторов портов написаны на том же языке, который используется для файлов конфигурации системы.

Командный файл конфигурации мониторов портов интерпретируется, когда запускается монитор порта. Командный файл конфигурации мониторов портов может замещать значения по умолчанию, поддерживаемые командным файлом конфигурации системы.

#### 4.1.4. Командные файлы конфигурации сервиса

Файлы конфигурации сервиса позволяют вам настраивать среду для конкретного сервиса. Например, сервис может потребовать каких-то специальных привилегий, которые не доступны обычному пользователю. Используя язык, описанный на странице руководства doconfig (3N), вы можете написать командный файл, который предоставит или ограничит такие специальные привилегии конкретного сервиса, предложенные через конкретный монитор порта.

Файл конфигурации сервиса может игнорировать безусловные значения, поддерживаемые файлами конфигурации более высокого уровня. Например, файл конфигурации сервиса может указать множество модулей STREAMS, отличное от безусловного множества.

#### 4.1.5. Управляющий файл SAC

Управляющий файл SAC содержит информацию о всех мониторах порта, за которые несет ответственность SAC. Этот файл существует в поставляемой системе. Изначально он является пустым, за исключением одной строки комментария, которая содержит номер версии контроллера доступа к сервису. Администратор системы добавляет мониторы портов к системе, осуществляя вводы в управляющий файл SAC. Эти вводы осуществляются с помощью команды sacadm с параметром -a. Команда sacadm также используется для удаления вводов из управляющего файла SAC.

Некоторые программные пакеты добавляют свои собстввенные вводы мониторов портов в ходе инсталяции, в других - вам придется добавлять их вручную.

Каждый ввод в управляющий файл SAC содержит следующую информацию:

PMTAG -

уникальный тег, который идентифицирует конкретный монитор порта. Этот тег затем используется контроллером доступа к сервису (SAC) для идентификации монитора порта при всех целях администрирования. PMTAG может включать до 14 буквенных символов;

PMTYPE -

тип монитора порта. В добавок к своему уникальному тегу, каждый монитор порта имеет указатель типа. Указатель типа идентифицирует группу мониторов портов, которые являются различными вызовами одного и того же объекта. ttymon и listen являются примерами действительных типов мониторов портов. Указатель типа используется, чтобы облегчить администрирование групп соответственных мониторов портов. Без указателя типа у вас нет возможности узнать какие теги мониторов портов соответствуют мониторам портов того же типа. PMTYPE может включать до 14 буквенных символов;

FLGS -

флаги, которые определяются в текущий момент, это: d - если запускается, не разрешайте монитор порта; x - не запускайте монитор порта. Если флаг не указывается, предпринимается безусловное действие. Монитор порта запускается и разрешается по умолчанию;

RCNT -

число раз, которое монитор порта может допустить сбой, прежде чем будет помещен в состояние отказа. После того как монитор порта ввел состояние отказа, SAC не будет пытаться повторно его запустить. Если счет не указан при создании ввода, это поле устанавливается на 0. Счет повторного запуска 0 указывает, что монитор порта не следует повторно запускать, если он дает отказ;

COMMAND -

символьная строка, представляющая команду, которая будет запускать монитор порта. Первый компонент символьной строки, сама команда, должен представлять собой полное имя пути.

Рис. 1 показывает содержимое образца управляющего файла SAC так, как он просматривается с помощью команды sacadm. Символ # в конце каждой строки является ограничителем комментария.

Вывод sacadm -1

------------------------------------------------------------

PMTAG PMTYPE FLGS RCNT STATUS COMMAND

starlan listen - 0 ENABLED /usr/lib/saf/listen -x

slan starlan \

# starlan liatener

ttymon1 ttymon d 0 DISABLED /usr/lib/saf/ttymon

# ttymon1

ttymon3 ttymon - 0 ENABLED /usr/lib/saf/ttymon

# ports board

------------------------------------------------------------

*Рис. 1*

#### 4.1.6. Управляющий файл монитора порта

Каждый монитор порта имеет свой собственный управляющий файл. Команда pmadm используется для добавления, удаления или модификации элементов в этом файле. При внесении каждого изменения соответствующему монитору порта сообщается о необходимости повторного считывания его управляющего файла.

Каждый элемент, вводимый в управляющий файл монитора порта определяет способ обработки монитором конкретного порта и сервис, который следует выполнять для этого порта. Некоторые поля должны присутствовать во всех типах мониторов портов. Каждый ввод должен включать тег сервиса для идентификации сервиса как уникального и значение, которое следует присвоить сервису, когда он будет запускаться.

*Примечание.* Комбинация тега сервиса и тега монитора порта уникально определяют случай использования сервиса. Тот же самый тег сервиса может использоваться для идентификации какого-то сервиса при другом мониторе порта.

Запись также должна содержать конкретные данные монитора порта, такие как строку подсказки, которая является значимой для ttymon. В целом, каждый тип монитора порта обеспечивает какую-то команду, которая воспринимает конкретные данные нужного монитора как аргументы и выводит эти данные в форме, соответствующей хранению в файле. Команда ttyadm (1M) делает это вместо ttymon, а nlsadmin (1M) - вместо listen.

*Примечание.* Если ПО при инсталяции добавляет требуемые вводы сервиса в управляющий файл монитора порта, вам не нужно добавлять ввод в ручную pmadm с параметром -a. Например, при инсталяции совместного использования дистанционных файлов (RFS), пакет ПО устанавливает требуемый сервис при каждом мониторе listen-типа.

Каждый ввод в управляющий файл монитора порта содержит следующую информацию:

SVCTAG -

уникальный тег, который идентифицирует сервис. Этот тег является уникальным только для монитора порта, через который этот сервис становится доступным. Другие мониторы порта могут предложить тот же или другой сервис с тем же самым тегом. Сервис требует наличия и тега монитора порта, и тега сервиса для идентификации его как уникального. SVCTAG может включать до 14 буквенных символов;

FLGS -

флаги с нижеследующими значениями могут быть включены как текущие в это поле:

x - Не разрешайте этот порт. Порт разрешается по умолчанию

u - Создавайте ввод utmp для этого сервиса. Обратите внимание, что мониторы портов могут игнорировать флаг u, если создание ввода utmp для сервиса не соответствует способу, в котором следует вызывать этот сервис. Некоторые программы сервиса не могут запускаться нужным образом, если для них не созданы вводы utmp (например, login);

ID -

значение под которым следует запускать сервис. Значение имеет ту форму имени протокола, в которой оно появляется в /etc/passwd;

PMSPECIFIC -

примерами конкретной информации мониторов порта являются адреса, имя процесса, который должен выполняться или имя программного канала STREAMS, через который устанавливается связь между процессами;

COMMENT -

комментарий, относящийся к элементу сервиса.

*Примечание.* Каждый управляющий файл монитора порта должен содержать один специальный комментарий следующей формы:

#VERSION = value

где value - это целое число, которое представляет номер версии монитора порта.

Номер версии определяет формат управляющего файла монитора порта. Эта строка комментария создается автоматически, когда монитор порта добавляется к системе. Он появляется на строке сам собой до появления элементов сервиса.

На рис. 2 показаны строки из образца управляющего файла ttymon. Обратите внимание, что вся информация в колонке PMSPECIFIC является конкретной информацией для монитора порта ttymon. К примеру, перечень управляющего файла listen будет содержать другое множество элементов в этой колонке. Конкретная информация монитора порта форматируется с помощью команды администратора монитора, в данном примере - командой ttyadm. Команда ttyadm включена как часть команды pmadm, когда она используется с параметром -a.

Рис. 2 показывает содержимое образца управляющего файла ttymon, просматриваемом с помощью команды pmadm. Символ # является ограничителем комментария.

------------------------------------------------------------

PMTAG PMTYPE SVCTAG FLGS ID PMSPECIFIC

ttymon3 ttymon 31 ux root /dev/tarm/31--/usx/bin/login

- 9600-login: -#/dev/tarm/31

ttymon3 ttymon 32 ux root /dev/tarm/32--/usx/bin/login

- 9600-login: -#/dev/tarm/32

ttymon3 ttymon 33 ux root /dev/tarm/33--/usx/bin/login

- 9600-login: -#/dev/tarm/33

ttymon3 ttymon 34 ux root /dev/tarm/34--/usx/bin/login

- 9600-login: -#/dev/tarm/34

------------------------------------------------------------

*Рис. 2*

*Примечание.* Чтобы поддержать целостность системы, настоятельно рекомендуется делать изменения в управляющих файлах SAC и мониторов портов с помощью команд sacadm и pmadm, а не редактирования файлов. SAC не распознает изменения в некоторых полях этих файлов, если они не выполнены с помощью требуемой команды администратора. Прямое редактирование файлов может привести к непредвиденным результатам.

#### 4.1.7. Установка связи с RS-232

При подсоединении устройства RS-232 к вашей системе, подключение должно осуществляться из вашего компьютера к этому устройству или прямо (жесткое соединение), или с помощью модема. После установки связи система должна знать какой тип соединения существует и какой тип устройства находится на другом конце связи.

Обычно при подсоединении RS-232 к вашему устройству предполагаются следующие подсказки:

* кабели RS-232 должны иметь длину менее 16 метров. Кабели, длина которых превышает 16 метров, могут привести к помехам на линии. В некоторых случаях, вы можете использовать более длинный кабель, если сократите скорость передачи (в бодах);
* обычно требуется линейный адаптер, называемый адаптер нулевого модема, если терминал или компьютер подсоединяется прямо к последовательному порту вашего компьютера;
* нулевой модем не следует использовать, если вы подсоединяете модем к своему компьютеру.