Общие параметры Х Window

ПАРАМЕТР ЗНАЧЕНИЕ

-background

красный зеленый сипни

Установка циста фона

-background цвет Установка цвета фона окна

-bg цвет Установка цвета фона окна

-display

система.'номер сервера

Использование Х сервера с заданным номером (обычно — 0) на заданной системе

-fg цвет Установка основного цвета окна

-fn шрифт Использование указанного шрифта

-font шрифт Использование указанного шрифта

-foreground color

красный зеленый синий

Установка основного цвета окна

-foreground цвет Установка основного цвета окна

-geometry

ширинахвысота+х+у

Установка размера и расположения окна

-geometry ширинахвысота Установка размеров окна

-geometry +x+y Установка положения верхнего левого угла окна

-height строки Установка размера окна по вертикали, в строках

-position x у Установка положения верхнего левого угла окна, в пикселах

Глава 1. Основные понятия и команды

-reverse Замена местами цвета фона

и основного цвета окна

-rv Замена местами цвета фона

и основного цвета окна

-size ширина высота Установка размера окна,

в пикселах

-Wb красный зеленый синий Установка цвета фона.

-Wf красный зеленый синий Установка основного цвета

окна

-WG ширинахвысота+х+у Установка размера

и расположения окна

-WG ширина^высота Установка размеров окна

-WG +.г-+г/ Установка положения

верхнего левого угла окна,

в пикселах

-Wh строки Установка размера окна

по вертикали, в строках

-Wi Запуск программы со

свернутым в значок окном

-width столбцы Установка ширины окна, в столбцах

-Wp л' г/ Установка положения

верхнего левого угла окна,

в пикселах

-Wr

система -.номерсервера Использование Х сервера

с заданным номером (обычно — 0)

на заданной системе

Компиляция С программ дсс

Опрос сервера DNS dnsquery

Создание FIFO mkfifo

FTP сервер ftpd

Компиляция программ на языке Fortran д77

1МАР сервер imapd

MIME-кодирование электронной почты mimencode, mmencode

Запуск интерпретатора Тс1 tclsh.wish

Отправка новостей Usenet Pnews, postnews

Проверка новостей Usenet checkhews

Вывод списка групп новостей Usenet getlist

Чтение новостей Usenet trn

WWW сервер httpd

Запуск X Window startx

Вывод названия аппаратной платформы arch

Создание архива cpio.tar

Работа с архивами ar

Создание архивов на ленте cpio.tar

Управление авторизацией xauth

Архивация файлов gzip.znew

Архивация исполняемых файлов gzexe

Вывод базы данных цветов showrgb

Интернетброузер lynx

Вывод содержимого буфера обмена xclipboard

Вывод информации о видеокарте SuperProbe

Смена владельца chown

Запуск команды в определенное время at

Вход в систему login

Вход в удаленную систему riogin

Выполнение вычислений be, pname, xcalc

Загрузка изображения в главное окно xpmroot

Установка атрибутов главного окна xsetroot

Смена группы chgrp, newgrp

Изменение и вывод текущей даты date

Деархивация файла gunzip, gzip, zcat,

unzip

Декодирование файлов uudecode

Вывод объема свободного места на диске df

Вывод информации о дисплее xdpyinfo

Просмотр документации apropos, info, man, whatis.xman

Управление доступом к серверу xhost

Отправка данных в два файла tee

Удаление задания на печать Iprm

Печать заголовка banner

Вывод информации о загрузке системы tload, xload

Захват изображения на экране xwd, xv

Глава 2. Краткая справка по командам Linux

Установка имени системы hostname

Вывод имени файла без пути basename

Интернетсервер inetd

Изменение

информации о пользователе chfn

Вывод информации о пользователе finger

Смена

интерпретатора командной строки chsh

Вывод информации

об использовании системы uptime, w

Вывод календаря cal

Смена текущего каталога cd

Создание каталога mkdir

Вывод имени текущего каталога pwd

Поиск в документации

по ключевому слову apropos

Вывод информации

о драйвере клавиатуры dumpkeys, kdbmode

Список запущенных клиентов xlsclients

Создание командной строки xargs

Запуск команды в указанное время at

Вывод конца файла tail

Смена корневого каталога chroot

Кодирование файла uuencode

Компиляция С программ gcc

Компиляция C++ программ gcc, g++

Компиляция программ на языке Fortran g77

Копирование файлов cat,cp

Монтирование накопителя

на магнитной ленте mt

Получение и отправка файлов

при помощи модема rx, sx,rb,sb,rz,sz

Монтирование диска mount

Монтирование гибкого диска fdmount

Монтирование накопителя на ленте mt

Создание мультимедийной

электронной почты metasend

Вывод начала файла head

Отправка новостей Usenet Pnews, postnews

Вывод списка групп новостей Usenet getlist

Чтение новостей Usenet trn

Вывод информации об окне xwininfo

Объединение файлов cat, paste

Объединение отсортированных файлов join

Очистка экрана clear

Вывод количества свободной памяти free

Установка параметров Х сервера xset

Смена пароля passwd

Установка значений

переменных окружения env

Вывод значений

переменных окружения printenv

Подготовка файла к печати РГ

Вывод информации о пользователе who, rwho

Вывод списка работающих

в локальной сети пользователей rusers

Вывод списка работающих

в системе пользователей users

Получение прав другого пользователя su

Запуск последовательности команд batch

Форматирование электронной почты formail

Кодирование электронной почты mimencode,

mmencode

Отправка и чтение электронной почты elm, mail, metasend,

pine

Отправка мультимедийной

электронной почты metasend

Отправка электронной почты

в формате MIME splitmail

Печать электронной почты printmail

Глава 2. Краткая справка по командам Linux

Получение электронной почты

по протоколу РОРЗ popclient

Сервер электронной почты imapd.sendmail

Уведомление о поступлении

новой электронной почты biff, xbiff, wnewmail

Передача файлов ftp.tftp

Перемещение файлов mv

Перерисовка экрана xrefresh

Печать фа ила Ipr

Печать электронной почты printmail

Подсчет количества слов we

Поиск файлов find

Поиск в файлах egrep, fgrep, zgrep

Преобразование рисунков atobm, bmtoa

Просмотр файлов cat, less, more

Просмотр заархивированных файлов zmore

Протоколирование сеанса работы script

Проверка правописания ispell

Запуск команд с пониженным

приоритетом nice

Изменение приоритета процесса snice

Вывод номера процесса pidof

Завершение работы процесса kill.killall.xkill

Вывод списка процессов PS

Вывод списка процессов, отсортирован

ного по процессорному времени,

занимаемому процессом top

Получение электронной почты

по протоколу РОРЗ popclient

Изменение раскладки клавиатуры xmodmap

Вывод расписания команды at atq

Задание расписания для выполнения

задач cron

Создание разделов на жестком диске fdisk

Редактирование текстового файла elvis.emacs.vi,

vim.xedit

Вывод реквизитов Х окна xprop

Установка Х ресурсов xrdb

Вы вод/ресурсов appres, listres, viewres

Смена режима доступа chmod

Вывод растрового рисунка sxpm

Преобразование рисунков atobm, bmtoa

Просмотр и редактирование рисунков bitmap

Редактирование и просмотр рисунков XV

Создание самораспаковывающихся

архивов shar

Просмотр производительности Х сервера x11perf

Сравнение производительности

Х серверов x11perfcomp

Управление доступом к серверу xhost

Вывод списка пользователей, подключен

ных к машинам локальной сети rusers

Вывод информации о системе host

Вывод системной статистики systat

Трассировка системных вызовов strace

Подсчет количества слов we

Просмотр информации о событиях xev

Установка соединения с удаленной

системой minicom, seyon

Вывод списка файлов в каталоге dir.ls

Сравнение файлов cmp,diff,diff3,sdiff,

zcmp, zdiff

Вывод системного идентификатора

пользователя logname

Сортировка файлов sort

Сравнение отсортированных файлов comm

Сравнение текстовых файлов diff,diff3,sdiff

Сравнение трех файлов diff3

Глава 2. Краткая справка по командам Linux

Сравнение заархивированных файлов xcmp, xdiff

Создание ссылок In

Создание столбца column

Удаление столбца cut

Редактирование текстового файла elvis, emacs, vi, vim, xedit

Поиск текстовых строк в бинарном файле strings

Изменение настроек терминала stty

Эмулятор терминала rxvt, xterm

Определение типа файла file

Телекоммуникационный пакет minicom, seyon

Трассировка strace

Удаление файлов rm

Удаление каталогов rmdir

Проверка файлов cksum

Установка свойств фона xsetroot

Установка фонового изображения xpmroot

Просмотр файла в формате PostScript ghostview

Генерация фракталов xfractint

форматирование файла fmt, groff

Форматирование гибкого диска fdformat

Вывод часов oclock, xclock

Установка ширины страницы fold

Вывод списка шрифтов fslsfonts, showfont, xfd, xfontsel, xlsfonts

Сервер шрифтов xfs

Вывод информации о сервере шрифтов fsinfo

Очистка экрана clear

Перерисовка экрана xrefresh

Увеличение части экрана xmag

Форматирование электронной почты formail

Кодирование электронной почты mimencode, mmencode

Отправка и чтение электронной почты elm, mail, metasend,

pine

Отправка электронной почты

в формате MIME splitmail

Создание мультимедийной

электронной почты metasend

Уведомление о приходе

электронной почты biff, xbiff, wnewmail

Запуск сервера электронной почты imapd.sendmail

Электронная таблица pname, spread

Назовите 15 команд, которые вы наиболее часто используете?

Некоторые примеры (В зависимости от рабочей платформы):

mkdir - Для создания папок (используйте -р для создания нескольких папок за раз)

ls - список директорий / файлов

top - мониторит статус ресурсов системы системы (свободная память, процессор, запущенные приложения и т.д)

lsof - Проверка что происходит на сервере, и какие файлы открываются какими процессами.

netstat -tcp - Дает полную картину о деталях подключения к сети.

vnstat - Статистика ширины полосы (bandwidth)

sh - Запускает shell скрипты

history - История команд, набранных пользователями

cd - Изменения директории

vi - Базовый текстовый редактор, стандартный для всех линуксов.

chmod - Изменение прав доступа к папкам и файлам.

mount - Монтирование файловых систем.

service - Запуск / перезапуск / остановка сервиса.

chkconfig - Перманентное включение / выключение сервиса.

fdisk -l - Список всех разделов вашей файловой системы

Это мой собственный список, у вас, конечно же может быть свой.

2) Назовите несколько команд для управления пользователями.

last, chage, chsh, lsof, chown, chmod, useradd, userdel, newusers.

3) Как делать проверку синтаксиса команд для следующих сервисов:

Какие команды делают возможным доступ без использования паролей к другим компьютерам.

expect и ssh-keygen

5) Назовите команды мониторинга безопасности.

lsof, netstat, top, ps -ef, tail, last, tcpdump, sestatus, history, w.

6) В чем разница между командами man, info, whatis и -help?

Команда whatis выдает вам ответ в одной строке.

Команда -help выдает вам одну строку ответа на каждую поддерживаемою командой опцию.

Команда man выдает всю информацию о команде и её опциях.

Команда info наиболее полную, из всех вышеназванных команд информацию о заданной команде.

В современных ядрах при подключении оборудования модули подключаются автоматически, а это событие обрабатывается демоном udev, который создает соответствующий файл устройства в каталоге "/dev". Все это выполняется в том случае, если соответствующий модуль корректно установлен в дерево модулей. В случае с файловыми системами ситуация та же: при попытке монтирования файловой системы ядро подгружает необходимый модуль автоматически, и выполняет монтирование.

Если необходимость в модуле не на столько очевидна, ядро его не загружает самостоятельно. Например, для поддержки функции шифрования на loop устройстве нужно вручную подгрузить модуль «cryptoloop», а для непосредственного шифрования — модуль алгоритма шифрования, например «blowfish».

Модули хранятся в каталоге "/lib/modules/<версия ядра>" в виде файлов с расширением «ko». Для получения списка всех модулей из дерева можно выполнить команду поиска всех файлов с расширением «ko» в каталоге с модулями текущего ядра:

find /lib/modules/`uname -r` -name ‘\*.ko’

Загрузить модуль в ядро можно при помощи двух команд: «insmod» и «modprobe», отличающихся друг от друга возможностью просчета и удовлетворения зависимостей. Команда «insmod» загружает конкретный файл с расширением «ko», при этом, если модуль зависит от других модулей, еще не загруженных в ядро, команда выдаст ошибку, и не загрузит модуль. Команда «modprobe» работает только с деревом модулей, и возможна загрузка только оттуда по имени модуля, а не по имени файла. Отсюда следует область применения этих команд: при помощи «insmod» подгружается файл модуля из произвольного места файловой системы (например, пользователь скомпилировал модули и перед переносом в дерево ядра решил проверить его работоспособность), а «modprobe» — для подгрузки уже готовых модулей, включенных в дерево модулей текущей версии ядра. Например, для загрузки модуля ядра «rt73usb» из дерева ядра, включая все зависимости, и отключив аппаратное шифрование, нужно выполнить команду:

# modprobe rt73usb nohwcrypt=0

Загрузка этого модуля командой «insmod» произойдет следующим образом:

# insmod /lib/modules/2.6.38-gentoo-r1/kernel/drivers/net/wireless/rt2x00/rt73usb.ko nohwcrypt=0

Но нужно помнить, что при использовании «insmod» все зависимости придется подгружать вручную. Поэтому эта команда постепенно вытесняется командой «modprobe».

После загрузки модуля можно проверить его наличие в списке загруженных в ядро модулей при помощи команды «lsmod»:

# lsmod | grep rt73usb

Чтобы его выгрузить, можно воспользоваться командой «rmmod» или той же командой «modprobe» с ключем "-r". В качестве параметра обоим командам нужно передать только имя модуля. Если модуль не используется, то он будет выгружен, а если используется — будет выдана ошибка, и придется выгружать все модули, которые от него зависят:

# rmmod rt2x00usb

ERROR: Module rt2x00usb is in use by rt73usb

# rmmod rt73usb

# rmmod rt2x00usb

После выгрузки модуля все возможности, которые он предоставлял, будут удалены из таблицы ядра.

Для автоматической загрузки модулей в разных дистрибутивах предусмотрены разные механизмы. Я не буду вдаваться здесь в подробности, они для каждого дистрибутива свои, но один метод загрузки всегда действенен и удобен: при помощи стартовых скриптов. В тех же RedHat системах можно записать команды загрузки модуля прямо в "/etc/rc.d/rc.local" со всеми опциями.

Файлы конфигурация модулей находится в каталоге "/etc/modprobe.d/" и имеют расширение «conf». В этих файлах преимущественно перечисляются альтернативные имена модулей, их параметры, применяемые при их загрузке, а так же черные списки, запрещенные для загрузки. Например, чтобы вышеупомянутый модуль сразу загружался с опцией «nohwcrypt=1» нужно создать файл, в котором записать строку:

options rt73usb nohwcrypt=1

Черный список модулей хранится преимущественно в файле "/etc/modules.d/blacklist.conf" в формате «blacklist <имя модуля>». Используется эта функция для запрета загрузки глючных или конфликтных модулей.

При появлении малейших неполадок с модулем, нужно смотреть сообщения ядра. Они выводятся по команде «dmesg» и, в зависимости от настроек syslog, в файл "/var/log/messages". Сообщения ядра могут быть информативными или отладочными, что поможет определить проблему в процессе работы модуля, а могут сообщать об ошибке работы с модулем, например недостаточности символов и зависимостей, некорректных переданных параметрах. Например, выше рассмотренный модуль «rt73usb» требует параметр типа bool, что говорит о том, что параметр может принимать либо «0», либо «1». Если попробовать передать «2», то система выдаст ошибку:

# modprobe rt73usb nohwcrypt=2

FATAL: Error inserting rt73usb (/lib/modules/2.6.38-gentoo-r1/kernel/drivers/net/wireless/rt2x00/rt73usb.ko): Invalid argument

Ошибка «Invalid argument» может говорить о чем угодно, саму ошибку ядро на консоль написать не может, только при помощи функции «printk» записать в системный лог. Посмотрев логи можно уже узнать в чем ошибка:

# dmesg | tail -n1

rt73usb: `2' invalid for parameter `nohwcrypt'

В этом примере выведена только последняя строка с ошибкой, чтобы не загромаждать статью. Модуль может написать и несколько строк, поэтому лучше выводить полный лог, или хотя бы последние строк десять.

Ошибку уже легко найти: значение «2» неприемлемо для параметра «nohwcrypt». После исправления, модуль корректно загрузится в ядро.

Просмотр общей информации о ядре (версии, имени ОС, аппаратная платформа и т.п.) производится с помощью команды uname.

Просмотреть список подключенных модулей в данный момент возможно с помощью команды lsmod:

Помимо имени модуля, команда lsmod показывает также размер, число пользователей модуля и имена пользователей.

Команда modinfo выдает информацию об одном или нескольких модулях.

Удаление модуля ядра происходит командой rmmod module\_name.

Удаленный модуль может понадобиться в процессе работы, для загрузки модуля необходимо выполнить команду: insmod /path/to/module.ko

Существует так же и другая команда для управления модулями: modprobe. Особенность данной команды в том, что она удаляет/добавляет модули с учетом зависимостей между модулями (зависимости между модулями прописаны в файле /lib/modules/версия/modules.dep). Пример использования:

Управление правами доступа происходит с помощью команды chmod, управление владельцем файла происходит с помощью команды chown.

Синтаксис команд следующий:

chmod [к\_какой\_группе\_прав][что\_сделать\_с\_правами][какие\_права] над\_каким\_объектом

или

chmod [права] над\_чем

где:

к\_какой\_группе\_прав

может быть u (от user) - владелец-пользователь, g (от group) - владелец-группа, o (от other) - остальные пользователи, a (от all) - все вышеперечисленные группы вместе

что\_сделать\_с\_правами

может быть + - добавить, - -убрать, = - присвоить указанное

какие\_права

может быть: r- чтение, w - запись, x - выполнение

над\_каким\_объектом

соответственно - имя или путь к файлу

права

числовое обозначение прав доступа (755, 644 и т.п.)

пример:

[Print-server]$ chmod a=rw file

[Print-server]$ ls -l file

-rw-rw-rw- 1 user users 78 Nov 20 file

это будет аналогично:

[Print-server]$ chmod ugo=rw file1

[Print-server]$ ls -l file1

-rw-rw-rw- 1 user users 78 Nov 20 file1

а так же:

[Print-server]$ chmod 666 file2

[Print-server]$ ls -l file2

-rw-rw-rw- 1 user users 78 Nov 20 file2

Использование команды chown выглядит следующим образом:

chown user:group file

сменить владельцев файла file на user:group.

[user]$ ls -l /bin/ls

-rwxr-xr-x 1 root root 49940 Sep 12 1999 /bin/ls

Вы видите, что в данном случае владельцем файла является пользователь root и группа root. Но нас сейчас в выводе этой команды больше интересует первое поле, определяющее тип файла и права доступа к файлу. Это поле в приведенном примере представлено цепочкой символов -rwxr-xr-x. Эти символы можно условно разделить на 4 группы.

Первая группа, состоящая из единственного символа, определяет тип файла. Этот символ в соответствии с возможными типами файлов, рассмотренными в предыдущем разделе, может принимать такие значения:

- = — обычный файл;

d = — каталог;

b = — файл блочного устройства;

c = — файл символьного устройства;

s = — доменное гнездо (socket);

p = — именованный канал (pipe);

l = — символическая ссылка (link).

[user]$ chmod wXp имя-файла

где вместо символа w подставляется

либо символ u (т. е. пользователь, который является владельцем);

либо g (группа);

либо o (все пользователи, не входящие в группу, которой принадлежит данный файл);

либо a (все пользователи системы, т. е. и владелец, и группа, и все остальные).

Вместо X ставится:

либо + (предоставляем право);

либо – (лишаем соответствующего права);

либо = (установить указанные права вместо имеющихся),

Вместо p — символ, обозначающий соответствующее право:

r (чтение);

w (запись);

x (выполнение).

Вот несколько примеров использования команды chmod:

[user]$ chmod a+x file\_name

предоставляет всем пользователям системы право на выполнение данного файла.

[user]$ chmod go-rw file\_name

удаляет право на чтение и запись для всех, кроме владельца файла.

[user]$ chmod ugo+rwx file\_name

дает всем права на чтение, запись и выполнение.

Если опустить указание на то, кому предоставляется данное право, то подразумевается, что речь идет вообще обо всех пользователях, т. е. вместо [user]$ chmod a+x file\_name

можно записать просто

[user]$ chmod +x file\_name

Второй вариант задания команды chmod (он используется чаще) основан на цифровом представлении прав. Для этого мы кодируем символ r цифрой 4, символ w — цифрой 2, а символ x — цифрой 1. Для того, чтобы предоставить пользователям какой-то набор прав, надо сложить соответствующие цифры. Получив, таким образом, нужные цифровые значения для владельца файла, для группы файла и для всех остальных пользователей, задаем эти три цифры в качестве аргумента команды chmod (ставим эти цифры после имени команды перед вторым аргументом, который задает имя файла). Например, если надо дать все права владельцу (4+2+1=7), право на чтение и запись — группе (4+2=6), и не давать никаких прав остальным, то следует дать такую команду:

[user]$ chmod 760 file\_name

Если вы знакомы с двоичным кодированием восьмеричных цифр, то вы поймете, что цифры после имени команды в этой форме ее представления есть не что иное, как восьмеричная запись тех самых 9 бит, которые задают права для владельца файла, группы файла и для всех пользователей.

Выполнять смену прав доступа к файлу с помощью команды chmod может только сам владелец файла или суперпользователь. Для того, чтобы иметь возможность изменить права группы, владелец должен дополнительно быть членом той группы, которой он хочет дать права на данный файл.

Чтобы завершить рассказ о правах доступа к файлам, надо рассказать еще о трех возможных атрибутах файла, устанавливаемых с помощью той же команды chmod. Это те самые атрибуты для исполняемых файлов, которые в индексном дескрипторе файла в двухбайтовой структуре, определяющей права на файл, занимают позиции 5-7, сразу после кода типа файла.

Первый из этих атрибутов — так называемый "бит смены идентификатора пользователя". Смысл этого бита состоит в следующем.

Обычно, когда пользователь запускает некоторую программу на выполнение, эта программа получает те же права доступа к файлам и каталогам, которые имеет пользователь, запустивший программу. Если же установлен "бит смены идентификатора пользователя", то программа получит права доступа к файлам и каталогам, которые имеет владелец файла программы (таким образом, рассматриваемый атрибут лучше называть "битом смены идентификатора владельца"). Это позволяет решать некоторые задачи, которые иначе было бы трудно выполнить. Самый характерный пример — команда смены пароля passwd. Все пароли пользователей хранятся в файле /etc/passwd, владельцем которого является суперпользователь root. Поэтому программы, запущенные обычными пользователями, в том числе команда passwd, не могут производить запись в этот файл. А, значит, пользователь как бы не может менять свой собственный пароль. Но для файла /usr/bin/passwd установлен "бит смены идентификатора владельца", каковым является пользователь root. Следовательно, программа смены пароля passwd запускается с правами root и получает право записи в файл /etc/passwd (уже средствами самой программы обеспечивается то, что пользователь может изменить только одну строку в этом файле).

Установить "бит смены идентификатора владельца" может суперпользователь с помощью команды

[root]# chmod +s file\_name

Аналогичным образом работает "бит смены идентификатора группы".

Еще один возможный атрибут исполняемого файла — это "бит сохранения задачи" или "sticky bit" (дословно — "бит прилипчивости"). Этот бит указывает системе, что после завершения программы надо сохранить ее в оперативной памяти. Удобно включить этот бит для задач, которые часто вызываются на выполнение, так как в этом случае экономится время на загрузку программы при каждом новом запуске. Этот атрибут был необходим на старых моделях компьютеров. На современных быстродействующих системах он используется редко.

Если используется цифровой вариант задания атрибутов в команде chmod, то цифровое значение этих атрибутов должно предшествовать цифрам, задающим права пользователя:

[root]# chmod 4775 file\_name

При этом веса этих битов для получения нужного суммарного результата задаются следующим образом:

4 — "бит смены идентификатора пользователя",

2 — "бит смены идентификатора группы",

1 — "бит сохранения задачи (sticky bit)".

Если какие-то из этих трех битов установлены в 1, то несколько изменяется вывод команды ls -l в части отображения установленных атрибутов прав доступа. Если установлен в 1 "бит смены идентификатора пользователя", то символ "x" в группе, определяющей права владельца файла, заменяется символом "s". Причем, если владелец имеет право на выполнение файла, то символ "x" заменяется на маленькое "s", а если владелец не имеет права на выполнение файла (например, файл вообще не исполняемый), то вместо "x" ставится "S". Аналогичные замены имеют место при задании "бита смены идентификатора группы", но заменяется символ "x" в группе атрибутов, задающих права группы. Если равен 1 "бит сохранения задачи (sticky bit)", то заменяется символ "x" в группе атрибутов, определяющей права для всех остальных пользователей, причем "x" заменяется символом "t", если все пользователи могут запускать файл на выполнение, и символом "T", если они такого права не имеют.

Таким образом, хотя в выводе команды ls -l не предусмотрено отдельных позиций для отображения значений битов смены идентификаторов и бита сохранения задачи, соответствующая информация выводится. Вот небольшой пример того, как это будет выглядеть:

[root]# ls -l prim1

-rwSrwsrwT 1 kos root 12 Dec 18 23:17 prim1

Операционная система (ОС далее) Linux поддерживает множество файловых систем, в настоящее время наиболее широко используются: ext2, ext3,ext4, reiserfs. Так же, современные ОС Linux совместимы с файловыми системами (ФС далее), используемыми ОС Windows, такими как NTFS и FAT32,

Список поддерживаемых Вашим ядром Linux файловых систем можно увидеть в файле /proc/filesystems.

Посмотреть сколько файл имеет ссылок и инод файла можно командой:

[mc-sim@ASUS /]$ ls -li

193 drwxr-xr-x 1 mc-sim root 368 Mar 30 2008 bin

1 drwxr-xr-x 1 mc-sim root 0 Jan 1 1970 dev

197 lrwxrwxrwx 1 mc-sim root 7 Mar 30 2008 etc -> tmp/etc

....

Файловые системы, доступные для установки системы.

Ext2

Эта файловая система была написана специально для Linux в 1993 году, как замена существовавшей тогда ext (Extended File System, расширенная файловая система). Данная ФС отличается невероятно высокой скоростью, уступая лишь более современной ext4 в тестах чтения. В настоящее время, код оригинальной Ext2 убран из ядра Linux, так как сильно устарел и его трудно поддерживать. Сейчас под Ext2 подразумевается Ext3 с отключённым журналированием. Ext2 часто используется на флеш-картах и некоторых SSD-накопителях, так как отсутствие журналирования является плюсом для таких накопителей, ибо продлевает срок их службы.

Достоинства:

Высокая производительность за счет отсутствия журналирования.

Недостатки:

Отсутствие журналирования. При ее использовании вам придется смириться с длительными проверками файловой системы при любом некорректном завершении работы. А при объемах современных дисков это может затянуться надолго.

Необходимо применение ИБП.

Не вполне рациональное использование дискового пространства.

Ext4

Современный вариант файловой системы Ext, предлагаемый по умолчанию в большинстве дистрибутивов Linux. Отличается высокой надёжностью, скоростью чтения/записи, поддержкой операций TRIM (необходимы для SSD-накопителей), автоматическая дефрагментация без размонтирования и многое другое. В большинстве случаев рекомендуется использовать эту ФС. Недавно в неё была добавлена поддержка шифрования (на уровне файловой системы). Ext4 - одна из самых быстрых и надёжных файловых систем. Кстати, Ext4 очень любит Google, и применяет её на своих серверах.

Достоинства:

Ext4 поддерживает гигантские объемы дисков и размеры файлов, работает с ними гораздо быстрее (очень заметно при удалении).

Низкий уровень фрагментации файловой системы, и, как следствие, прирост производительности.

Улучшена производительность некоторых баз данных и мультимедийных приложений

Ext4 умеет откладывать выделение места на диске на последний момент, что положительно сказывается на производительности.

Ext3 поддерживает 32 тысячи поддиректорий. В Ext4 ограничения сняты.

Запись контрольных сумм в журнал и, как следствие, высокая надежность.

Онлайн дефрагментация.

Восстановление удаленных файлов.

Недостатки:

Неэкономичность использования дискового пространства.

Ограничение на количество inode, что в случае его превышения (из-за хранения гигантского количества мелких файлов), может привести к невозможности записи данных на диск, даже если есть свободное место.

XFS

Файловая система XFS была разработана в середине 2000-х годов для операционной системы IRIX, которая была стандартом для создания выскопрофессиональной компьютерной графики. В начале 90-х, XFS была выпущена как свободное ПО, и включена в ядро Linux. Основным достоинством XFS, является прекрасная работа с очень большими файлами. К примеру если у вас есть диск, который вы используете для торрентов, то XFS для него будет лучшим выбором. XFS является 64-х битной ФС, поддерживает отложенное выделения дискового пространства (что позволяет оптимально расходовать место на диске и уменьшить фрагментацию), дефрагментацию на лету, а также динамическое выделение inodes, и их высвобождение для записи данных. Связка XFS+LVM+RAID mdadm (программный), весьма популярна в серверной среде.

Достоинства:

Очень эффективна в работе с большими по размеру файлами, хорошо работает также с большими по объему каталогами на чтение и поиск в них.

Рациональное использование дискового пространства.

Высокая производительность.

Онлайн дефрагментация.

Недостатки:

Высокая сложность восстановления удалённых данных.

Не рекомендуется держать на ней корневую файловую систему, так как xfs считает, что содержимое находившихся открытыми на запись файлов при некорректном прерывании работы системы не определено, и она заполняет эти файлы нулями.

В случае сбоя во время записи метаданных в журнал, возможна потеря данных (это характерно для всех ФС с журналированием метаданных). Рекомендуется использование ИБП (Источника Бесперебойного Питания).

JFS

JFS - это 64-х битная журналируемая файловая система от компании IBM, изначально созданная для их же операционной системы AIX. Существует 2 поколения JFS, которые используются в AIX. В Linux используется только второе поколение, JFS2. Отличается хорошей производительностью, лёгкостью восстановления удалённых данных, а также хорошей работой с мелкими файлами. Как и XFS, JFS журналирует только метаданные, потому в случае сбоя, во время записи метаданных в журнал, возможна потеря данных.

Достоинства:

Рациональное использование дискового пространства.

Дает хороший прирост в скорости работы.

Возможно восстановить данные с поврежденного тома, или же стертые данные.

Нет ограничений на количество файлов.

Достаточно производительна, особенно на небольших файлах.

Минимальная нагрузка на процессор.

Недостатки:

Средняя производительность.

Невысокая скорость работы с большими файлами.

Btrfs

Btrfs - это файловая система от компании Oracle, созданная в качестве ответа на файловую систему ZFS, от компании Sun (которую позже купила Oracle). Btrfs позиционируется как современная и многофункциональная замена Ext4, и помимо самой ФС, обладает функционалом менеджера логических томов, менеджера RAID, возможностью создания снимков состояния (а также снимков снимков), и копированием при записи (при копировании данных создаётся только символьная ссылка, а само копирование происходит только при обращении к этим данным для записи. Это позволяет существенно сократить используемое дисковое пространство для снимков состояния (снапшотов)). Файловая система Btrfs имеет структуру B-деревьев. Корнем дерева является сама ФС, а все каталоги и разделы диска - ветвями (так называемыми подтомами). Во время разметки диска, достаточно отдать всё пространство (кроме раздела swap) под Btrfs, а все необходимые разделы реализовывать в виде подтомов, которые будут монтироваться к общему дереву. В результате будет возможно создавать мгновенные снимки состояния разделов, и в случае необходимости - полностью восстанавливать их к исходному состоянию. Пример установки Ubuntu на диск с Btrfs, описан в этой статье.

Достоинства:

Высокая скорость работы.

Возможность создания снимков состояния (и снимков снимков)

Механизм Copy-on-Write (Копирование при записи).

Встроенный функционал менеджера логических томов и менеджера RAID.

Оптимизированна для SSD-накопителей.

Возможность сконвертировать ext4 в Btrfs.

Неограниченное количество inodes

Сжатие данных (значительная экономия дискового пространства).

Недостатки:

Пока нет возможности online-сканирования ФС на ошибки (без размонтирования).

Нет поддержки дедупликации данных.

Так как Btrfs всё ещё в состоянии активной разработки, она не отличается высокой надёжностью, потому не рекомендуется для использования в ответственных системах, таких как сервера.

ZFS

ZFS - это файловая система от компании Sun Microsystems (ныне принадлежит Oracle), которая была создана для их операционной системы Solaris. В 2005 году, Sun открыла код ZFS в рамках проекта OpenSolaris. Это позволило портировать ZFS на другие UNIX-подобные системы. В FreeBSD данная ФС доступна штатно, наряду с UFS. Для Linux доступен "ядерный" модуль ZFS on Linux. Лицензия CDDL, по которой распространяется ZFS, несовместима с GPL, и потому не позволяет включить ZFS в ядро Linux. Однако на официальном сайте доступны готовые установочные модули для большинства дистрибутивов. Данная файловая система отличается поддержкой неограниченных объёмов данных (на самом деле, конечно, ограничения есть, но максимальный размер данных так велик, что можно сказать, что объём данных практически бесконечен), поддержкой снимков состояния, функционалом менеджера логических томов, поддержкой дедупликации данных, механизмом Copy-on-Write и многим другим. Данная ФС рекомендуется для файловых хранилищ, а также серверов, и в отдельных случаях - для десктопов (например если у вас несколько дисков, объединённых в RAID-массив).

Достоинства:

128-битная файловая система, что даёт возможность хранения соответствующих объёмов информации

использование продвинутых алгоритмов хэширования для повышения надёжности хранения

поддержка снапшотов (snapshot), благодаря этой и следующей особенности ZFS сочетает в себе возможности файловой системы и системы управления томами (такой как LVM)

поддержка пулов хранения (storage pools)

высокая производительность на ряде задач

endian'о-независимость, что даёт возможность безболезнненого смешения устройств big endian и little endian

отсутствие необходимости в выполнении fsck благодаря использованию другого подхода к организации хранения данных

Недостатки:

Невозможно уменьшить zfs pool.

Невозможно увеличить количество устройств внутри избыточного vdev ( raidz/raidz2 )

Из-за широкого использования Copy-on-write возможна заметная фрагментация при частом изменении файлов. Дефрагментатора пока нет.

Штатно отсутствует в ядре Linux.

Потребляет значительное количество оперативной памяти.

F2FS

F2FS - это файловая система, оптимизированная для Flash и SSD накопителей, разработанная компанией Samsung и включённая в ядро Linux с версии 3.8. Для адаптации F2FS к различным видам флеш-накопителей, отличающихся своими характеристиками в зависимости от внутренней геометрии и схемы управления, предусмотрен широкий спектр параметров для управления структурой распределения данных в разделе и предоставлена возможность выбора различных алгоритмов очистки и выделения блоков. ФС учитывает все технические особенности твердотельных накопителей, такие как неизменное время доступа и ограниченный цикл перезаписи. Хранение структур данных организовано в форме журнала, а при обновлении информации используется механизм копирования при записи (Copy-On-Write, COW), при котором при изменении данные не перезаписываются, но сохраняются в новом месте. Метод работы F2FS позволит существенно продлить жизнь флеш-накопителей, поскольку файловая система учитывает внутреннюю геометрию расположения чипов в носителе и работу контроллера. Для снижения износа флеш-накопителя данные по возможности распределяются равномерно, сводя к минимуму повторную запись в одни и те же блоки.

Достоинства:

Специально оптимизирована для твердотельных накопителей и может существенно продлить их срок службы, а также обеспечить максимальное быстродействие.

Недостатки:

F2FS слишком молода, и пока не может похвастаться высокой надёжностью.

Недоступна непосредственно при установке большинства дистрибутивов (необходимо вручную переносить систему на раздел с F2FS).

NILFS

Это журнально-структурированная файловая система, разработанная специально для Linux. Будучи журнально-структурированной файловой системой (одна из разновидностей технологии «копирования-при-записи»), NILFS записывает данные в специальные журналоподобные файлы, при этом никогда их не перезаписывая, что позволяет минимизировать время поиска информации и избежать возможной потери данных, характерной для других файловых систем. Для примера, такая потеря может произойти на файловой системе ext3 при крахе компьютера в момент, когда производилась запись: после перезагрузки запись в журнале будет отменена и частично записанные данные потеряются.

Достоинства:

Малое время записи и восстановления данных

Минимальные повреждения файловых данных и целостности системы при аппаратных сбоях.

32-битные контрольные суммы (CRC32) для контроля целостности данных и метаданных (для групп блоков, для отдельных сегментов)

Запись данных и метаданных в правильном порядке

Резервные копии суперблока

Блоки файлов и inode-ов управляются B-tree структурой

64-битные внутренние данные

Поддержка больших файлов (8 Эксабайт)

Размер блоков меньше размера страницы (напр. 1 кб. или 2 кб.)

Изменение размера "на лету" (начиная с Linux-3.x и nilfs-utils 2.1)

Недостатки:

Отсутствие инструмента проверки ФС на ошибки (fsck)

Довольно неэкономное расходование дискового пространства.

Необходимость в использовании сборщика мусора nilfs\_cleanerd, в противном случае количество свободного места на диске определяется неправильно.

Недоступна во время установки системы.

Reiser4

Raiser4 - четвёртая версия одной из первых журналируемых файловых систем, разработанная специально для Linux.

включает в себя такие современные технологии как задержка выделения пространства, встроенную возможность кодирования и сжатия данных, транзакции. В настоящее время практически не развивается, так как главный разработчик ФС Ганс Рейзер, находится в тюремном заключении. Поддержка Raiser4 ведётся русским программистом Эдуардом Шишкиным. Raiser4 штатно отсутствует в ядре Linux, поэтому необходимо её добавление посредством патча.

Достоинства:

атомарная структура ФС (atomic filesystem) — позволяет производить операции файловой системы или полностью, или никак, но не наполовину; это повышает надёжность хранения данных;

использует алгоритм танцующего дерева, который пришёл на смену алгоритму B+-дерева, используемого в базах данных; Это позволяет Reiser4 более эффективно использовать пространство, благодаря тому, что маленькие файлы хранятся вместе и более высокой скорости работы, так как дерево балансируется только при записи на диск, достижении предела памяти или завершении транзакции; Благодаря редкой оптимизации появляется возможность достижения более глубокого уровня её проведения.

Архитектура Reiser4 основана на плагинах, что может облегчить добавление новых возможностей в файловую систему. Из значимых плагинов на данный момент существует плагин прозрачной компрессии (архиватор), позволяющий сжимать содержимое файлов перед записью на диск (что также значительно увеличивает скорость работы, уменьшая непосредственно считываемый/записываемый объём данных).

Недостатки:

Raiser4 в настоящее время почти не развивается, ведётся лишь исправление мелких ошибок и адаптация под новые версии ядра Linux, а также небольшие изменения функциональности.

Отсутствует в ядре Linux.

Прочие файловые системы.

NTFS

Linux поддерживает файловую систему NTFS, применяемую в Windows. Её поддержкой занимается драйвер ntfs-3g, который работает через FUSE (File System on Userpace, файловая система в пространстве пользователя). Однако производительность будет ниже, чем при использовании модуля для ядра от компании Paragon (модуль платный). Тем не менее, скорость чтения/записи весьма высока, ntfs-3g работает с любыми NTFS-накопителями, в следствии простой реализации драйвера, высока его надёжность.

HFS+

Linux поддерживает файловую систему HFS+, используемую в Mac OS X в режиме чтения со включённым журналированием, и в режиме чтения/записи с отключённым. Для возможности полноценного использования HFS+ в режиме чтения/записи, также существует платный драйвер от Paragon.

UFS

Linux поддерживает файловую систему UFS (Unix File System), используемую в FreeBSD и других UNIX-системах. Штатно поддерживается монтирование в режиме только чтение, для монтирования в режиме чтение/запись, в ядре должна быть включена поддержка UFS.

FAT/FAT32

Разумеется Linux поддерживает старые (я бы даже сказал, древние) фаловые системы FAT и FAT32

extFAT

Linux поддерживает файловую систему extFAT, разработанную компанией Майкрософт в режиме чтения/записи, посредством FUSE-драйвера. Это современная версия FAT, обладающая следующими достоинствами:

Уменьшение количества перезаписей одного и того же сектора, что важно для флеш-накопителей, у которых ячейки памяти необратимо изнашиваются после определённого количества операций записи (это сильно смягчается выравниванием износа — wear leveling, — встроенным в современные USB-накопители и SD-карточки). Это была основная причина разработки ExFAT.

Теоретический лимит на размер файла 264 байт (16 эксабайт).

Максимальный размер кластера увеличен до 225 байт (32 мегабайта).

Улучшение распределения свободного места за счёт введения бит-карты свободного места, что может уменьшать фрагментацию диска.

Введена поддержка списка прав доступа.

Поддержка транзакций (опциональная возможность, должна поддерживаться устройством).

Из недостатков:

Более сложная структура, чем у FAT, вызывает больший расход вычислительных ресурсов компьютера.

Проприетарная, разрабатывается Майкрсоофт и очень плохо поддерживается в относительно старых системах (нужно устанавливать драйвера).

Конечно, это не весь список файловых систем, которые поддерживает Linux. Есть там ещё малоизвестные и специфичные файловые системы. Линукс обладает, наверное, самой широкой поддержкой разнообразных файловых систем, поэтому какой-бы накопитель вы не подключили, с вероятностью в 98% он будет работать сразу и без нареканий.

Начальный загрузчик

Когда вы включаете компьютер, прошивка BIOS или UEFI вашего компьютера загружает программное обеспечение с загрузочного устройства. Первая программа, которая загружает в любой операционной системе, является начальным загрузчиком. В Linux, это, как правило, начальный загрузчик Grub.

Если у вас установлено несколько операционных систем, то Grub предлагает меню, позволяющее выбирать одну из них - например, если у вас установлен Linux в конфигурации с двойной загрузкой, вы можете, когда происходит загрузка, выбрать либо Linux или Windows.

Если у вас установлена только одна операционная система, то Grub может загрузить систему Linux практически мгновенно, но он все еще продолжает обработку. Grub обрабатывает процесс непосредственной загрузки Linux, задает параметров командной строки и для того, чтобы устранить неполадки, позволяет вам загрузить Linux другими способами. Без начального загрузчика, дистрибутив просто не будет загружаться.

Ядро Linux

Определенная часть программного обеспечения, которое загружается с помощью Grub, является ядром Linux. Это та часть системы, которая на самом деле называется «Linux». Ядро является центральной частью системы. Он управляет вашим процессором, памятью и устройствами ввода/вывода, такими как клавиатуры, мыши и дисплеи. Поскольку ядро обращается непосредственно к оборудованию, многие драйверы оборудования являются частью ядра Linux и запускаются внутри ядра.

Все другое программное обеспечение работает выше ядра. Ядро является частью программного обеспечения, которое лежит на самом нижнем уровне и взаимодействует с аппаратным обеспечением. Он представляет собой слой абстракции над оборудованием, который имеет дело со всеми аппаратными причудами, тогда как остальная часть системы может почти совсем с ними не иметь дело. В Windows используется ядро Windows NT, а в Linux используется ядро Linux.

Демоны

Демоны, по существу, являются фоновыми процессами. Они часто запускаются как часть процесса начальной загрузки, поэтому они являются тем, что загружается после ядра и прежде, чем вы видите свой графический экран входа в систему. В Windows такие процессы называются «сервисами», в то время как в UNIX-подобных системах они называются «демонами» (прим.пер.: более точно - «даемонами» от английского « daemons»).

Например, crond, который управляет выплнением запланированных задач, является демоном - буква d в конце означает «демон». syslogd является еще одним демоном, который традиционно управляет вашим системным журналом. Серверы, такие как сервер sshd, работают как демоны фонового режима. Это гарантирует, что они всегда работают и прослушивают дистанционные подключения.

Демоны, по существу, являются всего лишь фоновыми процессами, но они являются процессами системного уровня, работу которых вы вообще не замечаете.

Командная оболочка

В большинстве систем Linux по умолчанию используется командная оболочка Bash. Командная оболочка обеспечивает интерфейс командного процессора, что позволяет управлять вашим компьютером с помощью ввода команд в текстовом режиме. В командной оболочке можно также запускать скрипты командного процессора, которые представляют собой набор команд и операций, выполняемых в тем порядке, который указан в скрипте.

Даже если вы пользуетесь только графическим рабочим столом, командные оболочки работают и используются в фоновом режиме. Когда вы откройте окно терминала, вы видите приглашение к работе, выдаваемое командной оболочкой.

Утилиты командной оболочки

В командной оболочке предоставляется некоторое количество основных встроенных команд, но большинство из команд командной оболочки, которыми пользуются пользователи Linux, не встроены в командную оболочку. Например, такие важные команды, как команда cp, используемая для копирования файлов, команда ls, используемая для получения списка файлов в каталоге, и команда rm, используемая для удаления файлов, являются частью пакета базовых утилит GNU Core Utilities.

Вот почему много споров относительно того, должен ли Linux действительно назваться «Linux» или «GNU/Linux». Критики названия «Linux» правильно отмечают, что большая часть программного обеспечения идет в составе типичных систем Linux, что часто не признается. Критики названия «GNU/Linux» правильно отмечают, что в типичную систему Linux также входит другое важное программное обеспечение, которое нельзя охватить названием «GNU/Linux».

Не все из утилит командной оболочки и программ, работающих из командной строки, разрабатываются в рамках проекта GNU. Некоторые команды и терминальные программы имеют свои собственные проекты, предназначенных специально для них.

Графический сервер X.org

Часть Linux, представляющая собой графический рабочий стол, не является частью ядра Linux. Она реализуется с помощью пакете специального вида, известного как «Сервер X», поскольку он реализует «Оконную систему X», которая возникла много лет назад.

В настоящее время наиболее популярным сервером X или графическим сервером является сервер X.org. Когда вы видите окно графического входа или появляется рабочий стол, то это свое волшебство проявляет сервер X.org. Работа всей графической системы осуществляется под управлением сервера X.org, который взаимодействует с вашей видеокарты, монитором, мышью и другими устройствами. X.org не предоставляет полную окружение рабочего стола, это просто графическая система, поверх которой могут быть созданы окружение и инструментарий рабочего стола.

Среда рабочего стола

То, что вы действительно используете на рабочем столе Linux, представляет собой рабочую среду. Например, в состав системы Ubuntu входит среда рабочего стола Unity, в состав системы Fedora входит GNOME, в Kubuntu входит KDE, в в состав системы Mint обычно входит Cinnamon or MATE. Эти окружения рабочего стола обеспечивают все, что вы видите - фон рабочего стола, панели, заголовки и границы окон.

Они также обычно включают в себя свои собственные утилиты, которые встроены для того, чтобы окружение рабочего стола выглядело как единое целое. Например, в GNOME и Unity входит файловый менеджер Nautilus, разработанный в рамках GNOME, а KDE включает в себя файловый менеджер Dolphin, разработанный в рамках проекта KDE.

Программы рабочего стола

Не каждая программа, работающая на рабочем столе, является частью среды рабочего стола. Например, программы Firefox и Chrome работают независимо от среды рабочего стола. Они просто являются программами, которые могут нормально работать поверх любой среды рабочего стола. Либо пакет OpenOffice.org, который является еще одним набор программ, которые не привязаны к конкретной среде рабочего стола.

Вы можете запускать любую программу Linux, предназначенную для рабочего стола, в любой графической среде, но те программы, которые предназначены для определенной рабочей среды, могут выглядеть не к месту или могут мешать другим процессам. Например, если вы пытаетесь запустить файловый менеджер Nautilus, предназначенный для GNOME, в среде KDE, он будет выглядеть не к месту, потребует от установки разнообразных библиотек GNOME, и, вероятно, когда вы откроете его, будет в фоновом режиме запускать процессы рабочего стола GNOME. Но он будет работать и его можно будет использовать.

Последний шаг остается за дистрибутивами Linux. Они берут все это программное обеспечение, объединяют его так, чтобы оно хорошо работало совместно, и добавляют свои собственные необходимые утилиты. Например, для дистрибутивов создаются их собственные инсталляторы операционной системы с тем, чтобы вы действительно смоли установить Linux, а также менеджеры пакетов для установки дополнительного программного обеспечения и поддержки установленного программного обеспечения в обновленном виде.

Первое что мы сделаем - это посмотрим текущие параметры сети, для этого в консоли от суперпользователя выполним команду:

sudo ifconfig

Далее мы научимся останавливать, запускать, перезапускать сетевые интерфейсы. И первой командой будет команда которая остановит сетевой интерфес eth0:

sudo ifconfig eth0 down

Раз мы с вами его остановили, то надо научиться его и обратно подключать:

sudo ifconfig eth0 up

Следующая команда остановит все, без исключения сетевые интерфейсы на вашем ПК:

sudo /etc/init.d/networking stop

Следующая опять же запустит, только теперь все сразу:

sudo /etc/init.d/networking start

Ну и осталось только научиться перезапускать их:

sudo /etc/init.d/networking restart

И первое с чего мы начнем настройку сети - это настройка адреса, маски и шлюза, если конечно у вас не настроен DHCP, для этого открываем файл /etc/network/interfaces:

sudo nano /etc/network/interfaces

Для ручной настройки ip адреса, маски и шлюза, этот файл примерно должен иметь такой вид:

auto lo

iface lo inet loopback

auto eth0

iface eth0 inet static

address 10.54.89.86

netmask 255.255.255.0

gateway 10.54.89.65

Первые две строчки отвечают за настройку локальной петли, а так как она настроена, то нам ее трогать не надо. Перейдем к строке auto eth0, которая поясняет, что этот интерфейс должен автоматом запустится вместе с операционной системой. Строка iface eth0 inet static - поясняет, что используются ручные настройки сети. address 10.54.89.86 - производит установку ip адреса для сетевой карты. netmask 255.255.255.0 - соответственно устанавливает маску и наконец gateway 10.54.89.65 - установка ip адреса шлюза.

Если у вас на роутере настроена автоматическая раздача адресов (DHCP), то конфигурационный фал будет таким:

auto lo

iface lo inet loopback

auto eth0

iface eth0 inet dhcp

Тут я ничего пояснять не буду, так как здесь и так все понятно.

Вот в принципе и вся настройка сети, осталась только настроить адрес dns сервера, для этого открываем файл /etc/resolv.conf:

# Generated by NetworkManager

nameserver 10.0.1.1

Где nameserver 8.8.8.8 - это и есть как раз адрес вашего dns сервера. По секрету вам расскажу одну фишку, если у вас скорость отклика интернета довольна низкая, то поставьте у себя в качестве dns сервера, сервер от Google, он имеет адрес 8.8.8.8. Скорость отклика интернета должна хоть немного возрасти. Как видите, чтобы linux настройка сети прошла удачно, никаких сверхзнаний и сверхумений не потребовалось. На этом все! Всем удачи.