

Тема:

Настройка и знакомство с интерфейсом командной строки.



План занятия:

1. Знакомство с интерфейсом командной строки. Настройка.
2. Навигация по файловой системе и основные операции с файлами и каталогами.
3. Знакомство с текстовыми редакторами.

1. Знакомство с интерфейсом командной строки. Настройка.

Интерфейс командной строки (Command-Line Interface, CLI) - это текстовый способ взаимодействия с операционной системой или программами с использованием командной строки.

CLI предоставляет пользователю возможность вводить команды в текстовой форме, а операционная система или программа реагирует на эти команды.

Командная строка также известна как терминал или консоль.

Основные элементы интерфейса командной строки:

Приглашение команды (Prompt). Это текстовая строка, которая приглашает пользователя ввести команду. Обычно содержит информацию о текущем пользователе, хосте, пути и/или другие детали.

```
user@hostname:~$
```

Команда. Это текстовая строка, вводимая пользователем для выполнения определенной операции или задачи. Команда может включать имя программы, опции и аргументы.

```
ls -l /path/to/directory
```

Основные элементы интерфейса командной строки:

Опции: Это флаги или параметры, которые изменяют поведение команды. Опции обычно представляют собой одиночные символы (короткие опции), предваренные дефисом, или слова (длинные опции), предваренные двумя дефисами.

```
ls -l
```

```
ls --all
```

Аргументы: Это данные или параметры, передаваемые команде. Они предоставляют дополнительную информацию, необходимую для выполнения задачи.

```
cp file.txt destination_directory/
```

Настройка интерфейса командной строки:

- Персонализация приглашения командной строки: можно изменить символы, отображающиеся перед вводом команды, чтобы сделать их более удобными или информативными.
- Настройка цветовой схемы: можно изменить цвета выводимого текста, чтобы выделить различные типы файлов или подчеркнуть важные сообщения.
- Создание и настройка командных алиасов: алиасы позволяют создавать сокращенные формы команд или сочетания команд для удобства использования.
- Изменение переменных окружения: переменные окружения определяют параметры работы командной строки, такие как пути поиска исполняемых файлов или настройки поведения команд.

Чтобы изменить символы, отображаемые перед вводом команды в командной строке (приглашение), вы можете настроить переменную PS1 (Prompt String 1). Вот несколько примеров того, как это можно сделать:

Изменение символа приглашения на символ доллара (\$):

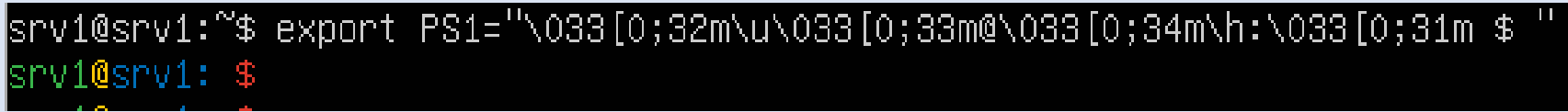
```
export PS1="$ "
```

Добавление имени пользователя и имени хоста в приглашение:

```
export PS1="\u@\h $ "
```

Использование цветов для приглашения:

```
srv1@srv1:~$ export PS1="\033[0;32m\u\033[0;33m@\033[0;34m\h:\033[0;31m $ "
```



\u: Это специальная последовательность символов, которая будет заменена на имя текущего пользователя.

\h: Это специальная последовательность символов, которая будет заменена на имя текущего хоста

\w: Полный путь к текущему каталогу.

\W: Только имя текущего каталога.

"\[\033[01;32m\]: Это начало последовательности ANSI Escape-кода, который устанавливает зеленый цвет для символов, следующих за ним.

Некоторые основные ANSI Escape-коды для установки цвета текста:

\033[0;30m: Черный цвет.

\033[0;31m: Красный цвет.

\033[0;32m: Зеленый цвет.

\033[0;33m: Желтый цвет.

\033[0;34m: Синий цвет.

\033[0;35m: Фиолетовый цвет.

\033[0;36m: Голубой цвет.

\033[0;37m: Серый цвет.

После перезагрузки настройки переменной PS1, установленные в текущей сессии командной строки, не сохраняются по умолчанию.

При перезагрузке, командная строка будет использовать значения из файла конфигурации оболочки (например, ~/.bashrc или ~/.bash_profile) для установки переменной PS1 и других настроек окружения.

Чтобы сделать изменения переменной PS1 постоянными и сохранить их после перезагрузки, нужно добавить соответствующую команду в файл конфигурации оболочки. Вот пример того, как это можно сделать:

Откройте файл конфигурации оболочки в текстовом редакторе. Например, для Bash обычно используется файл ~/.bashrc.

Добавьте строку, содержащую команду `export PS1=...` с желаемым значением приглашения.

2. Навигация по файловой системе и основные операции с файлами и каталогами.

Файловая система.

Файловая система - это способ организации и хранения файлов на компьютере или другом устройстве. Она определяет структуру, доступ и управление данными, а также правила и форматы именования файлов.

В операционной системе Linux файловая система организована в виде иерархической структуры директорий, начиная с корневой директории ("/"). В этой иерархии каждая директория может содержать поддиректории и файлы. Файловая система Linux подчиняется стандарту Файловой Системы иерархии Linux (Filesystem Hierarchy Standard - FHS), который определяет общие правила и структуру файловой системы.

Файловая система.

Основные понятия в файловой системе Linux:

1. Корневая директория ("/"): Это верхний уровень иерархии файловой системы. Все другие директории и файлы находятся внутри корневой директории.

2. Путь: Путь - это уникальный адрес, указывающий на расположение файла или директории в файловой системе. Абсолютный путь начинается с корневой директории и указывает полный путь к файлу или директории ("/home/user/documents"). Относительный путь указывает путь относительно текущей рабочей директории ("../documents").

3. Типы файлов: В Linux есть различные типы файлов, включая обычные файлы (текстовые, исполняемые и т.д.), директории (которые содержат другие файлы и директории), символические ссылки (указывающие на другие файлы или директории), устройства (файлы, представляющие аппаратные устройства) и другие специальные файлы.

Файловая система.

Debian и Red Hat (включая CentOS) – два популярных дистрибутива Linux, которые используют различные файловые системы по умолчанию. Вот информация о файлах системах, используемых в Debian и Red Hat.

Debian:

По умолчанию Debian использует файловую систему ext4. Ext4 является стандартной файловой системой для большинства дистрибутивов Linux, включая Debian. Она обеспечивает хорошую производительность, надежность и совместимость с предыдущими версиями ext. Ext4 поддерживает большие размеры файлов и разделов, а также журналирование для обеспечения целостности данных.

Red Hat (включая CentOS):

По умолчанию Red Hat и CentOS используют файловую систему XFS. XFS была разработана компанией SGI и предназначена для обработки больших объемов данных и обеспечения высокой производительности. XFS обладает высокой масштабируемостью, эффективным использованием многопроцессорных систем и быстрым восстановлением после сбоев. Она широко используется в серверных и хранилищных системах, где требуется обработка больших объемов данных.

В обоих дистрибутивах Linux, Debian и Red Hat, вы также можете выбрать другие файловые системы при установке операционной системы или в процессе настройки дискового пространства.

Значение файловой системы для операционной системы Linux.

Организация и хранение данных: Файловая система определяет способ, которым данные организованы и хранятся на диске. Она определяет структуру каталогов, файлов и метаданных, таких как права доступа, временные метки и атрибуты файлов.

Управление доступом и безопасностью: Файловая система поддерживает механизмы управления доступом, позволяющие контролировать, кто может читать, записывать или выполнять файлы и директории. Она также обеспечивает механизмы безопасности, такие как права доступа, аудит и шифрование, чтобы защитить данные от несанкционированного доступа.

Производительность и оптимизация: Различные файловые системы имеют разную производительность в зависимости от типа работы и характеристик хранения данных. Некоторые файловые системы могут обеспечивать высокую скорость чтения/записи, поддержку больших файлов или эффективное использование дискового пространства.

Восстановление после сбоев: Файловые системы обычно используют журналирование. Журналирование позволяет системе восстановить файловую систему до последнего состояния до сбоя, минимизируя потерю данных и ускоряя процесс восстановления.

Расширяемость и функциональность: Некоторые файловые системы предлагают дополнительные функции, такие как снимки (snapshots), сжатие данных, шифрование и RAID (распределенный массив недорогих дисков).

Выбор подходящей файловой системы зависит от конкретных требований и сценариев использования. Различные файловые системы имеют свои преимущества и ограничения, поэтому важно учитывать факторы, такие как производительность, масштабируемость, безопасность, надежность и функциональность при выборе файловой системы для конкретного сервера или компьютера.

Корневая директория и иерархия файловой системы.

Корневая директория и иерархия файловой системы являются важными концепциями в операционной системе Linux.

Корневая директория обозначается символом `/`, и она является верхним уровнем иерархии файловой системы в Linux.

Все файлы и директории на компьютере находятся внутри корневой директории или ее поддиректорий.

В Linux нет дисковых букв, как в Windows (например, `C:`, `D:`) – вместо этого используется одно древовидное пространство, начинающееся с корневой директории.

Корневая директория и иерархия файловой системы.

Иерархия файловой системы (Filesystem Hierarchy) определяет структуру и организацию файлов и директорий в Linux. Она представляет собой стандарт, который определяет, где должны размещаться различные типы файлов и директорий в системе. Основные директории в иерархии файловой системы обычно включают:

`/bin`: Содержит исполняемые файлы (программы), доступные для всех пользователей.

`/boot`: Содержит файлы, связанные с загрузчиком и ядром операционной системы.

`/dev`: Содержит файлы, представляющие устройства (например, жесткие диски, принтеры, клавиатуру и т. д.).

`/etc`: Содержит конфигурационные файлы системы и приложений.

`/home`: Каталоги пользователей, где хранятся их личные файлы и настройки.

`/lib`: Библиотеки, используемые исполняемыми файлами в `/bin` и `/sbin`.

`/mnt`: Точки монтирования для временного монтирования внешних устройств или сетевых ресурсов.

`/opt`: Дополнительные приложения (optional), которые не являются частью базовой системы.

`/proc`: Виртуальная файловая система, предоставляющая информацию о работающих процессах и системе.

`/root`: Домашняя директория пользователя root (суперпользователя).

`/sbin`: Системные исполняемые файлы, доступные только администратору.

`/tmp`: Временная директория для хранения временных файлов.

`/usr`: Содержит различные файлы, включая исполняемые файлы, библиотеки, заголовочные файлы и документацию.

`/var`: Файлы с переменными данными, такими как журналы, кэши, временные файлы и т. д.

Понятие пути и абсолютных/относительных путей.

В контексте файловых систем путь - это строка, которая определяет местоположение файла или директории в иерархии файловой системы. Путь указывает на то, как добраться от корневой директории до нужного файла или директории.

Абсолютный путь:

Абсолютный путь полностью определяет местоположение файла или директории относительно корневой директории. Он начинается с символа "/", который представляет корневую директорию, и затем указывает каждую последующую директорию в пути. Например, абсолютный путь к файлу "example.txt" может выглядеть так: `"/home/user/documents/example.txt"`. Абсолютный путь всегда уникален и не зависит от текущего рабочего каталога.

Относительный путь:

Относительный путь указывает на местоположение файла или директории относительно текущего рабочего каталога. Текущий рабочий каталог - это директория, в которой находится пользователь или процесс в данный момент. Относительный путь не начинается с символа "/", а указывает только последовательность директорий относительно текущего рабочего каталога. Например, если текущий рабочий каталог `"/home/user"`, то относительный путь к файлу "example.txt" в подкаталоге "documents" будет выглядеть так: `"documents/example.txt"`. Относительные пути могут быть относительными к текущему рабочему каталогу или относительно других директорий.

Типы файлов.

В файловых системах Linux существует несколько типов файлов, каждый из которых имеет свою специфическую функцию. Некоторые из наиболее распространенных типов файлов включают:

Обычные файлы содержат данные, такие как текстовые документы, изображения, видеофайлы или исполняемые программы. Они являются наиболее распространенным типом файлов и хранят информацию в бинарном или текстовом формате.

Директории представляют собой контейнеры, в которых можно хранить другие файлы и директории. Они используются для организации файловой системы в иерархическую структуру, позволяющую легко найти и управлять файлами.

Символические ссылки, также известные как "символические ссылки" или "симлинки", являются специальными файлами, которые содержат путь к другому файлу или директории. Они создаются для создания альтернативного пути к файлам или директориям, что позволяет ссылаться на них по другому имени или расположению.

Типы файлов.

Специальные файлы представляют собой файлы, используемые для взаимодействия с устройствами в системе. Некоторые из типов специальных файлов включают:

Блочные устройства (Block devices): Используются для обработки блоков данных, например, жестких дисков или флэш-накопителей.

Символьные устройства (Character devices): Используются для обработки потоков символов, например, клавиатуры, мыши или принтера.

FIFO (Named pipes): Используются для обмена данными между процессами через файловую систему.

Сокеты (Sockets): Используются для обмена данными между процессами по сети.

Исполняемые файлы содержат машинный код и предназначены для запуска программ. Они могут быть скомпилированными исполняемыми файлами (.exe в Windows) или сценариями на интерпретируемых языках программирования (например, сценарии на языке Shell или Python).

Распространенные файловые системы .

Существует несколько распространенных файловых систем, которые широко используются в различных операционных системах. Некоторые из наиболее популярных файловых систем включают:

1.ext4 (Fourth Extended Filesystem):

ext4 является наиболее распространенной файловой системой в операционной системе Linux. Она является преемницей ext3 и предоставляет улучшенные возможности, такие как поддержка больших файлов и разделов, улучшенная производительность и надежность.

2.XFS (XFS Filesystem):

XFS разработана для операционной системы Unix и предоставляет высокую производительность и масштабируемость. Она поддерживает большие файлы и разделы, обеспечивает хорошую производительность в условиях высоких нагрузок на ввод-вывод и обладает надежностью и целостностью данных.

3.Btrfs (B-tree File System):

Btrfs является современной файловой системой для Linux, предназначенной для обеспечения высокой производительности, надежности и гибкости. Она поддерживает функции копирования на запись (Copy-on-Write), снимки (Snapshots), сжатие данных и дедупликацию (Data deduplication).

4.NTFS (New Technology File System):

NTFS является файловой системой, используемой в операционных системах семейства Windows, начиная с Windows NT. Она поддерживает различные функции, такие как разграничение прав доступа, журналирование для восстановления после сбоев и поддержку файлов большого размера.

5.APFS (Apple File System):

APFS является файловой системой, разработанной компанией Apple для использования в операционных системах macOS, iOS, watchOS и tvOS. Она обеспечивает высокую производительность, поддержку шифрования, сжатие данных, снимки и другие функции.

6.FAT32 (File Allocation Table):

FAT32 является простой файловой системой, используемой во многих операционных системах. Она поддерживает широкую совместимость и может быть прочитана и записана на различных платформах, включая Windows, macOS и Linux. Однако она ограничена по размеру файлов и разделов.

Особенности и преимущества файловых систем.

ext4: Особенности: поддержка файлов и разделов большого размера (до 1 экзбайта и 16 терабайт соответственно), журналирование для восстановления после сбоев, поддержка различных атрибутов файлов и директорий.

Преимущества: высокая производительность, надежность и совместимость с предыдущими версиями ext-файловых систем, широкая поддержка в операционных системах Linux.

XFS: Особенности: поддержка больших файлов и разделов (до 8 экзбайт и 8 экзбайт соответственно), высокая производительность в условиях высоких нагрузок на ввод-вывод, журналирование для восстановления после сбоев.

Преимущества: хорошая масштабируемость, эффективное использование пространства на диске, поддержка снимков (snapshots) для создания точек восстановления.

Btrfs: Особенности: копирование на запись (Copy-on-Write), снимки (Snapshots) для создания состояний файловой системы на определенный момент времени, сжатие данных для экономии места на диске, дедупликация данных для устранения дубликатов.

Преимущества: высокая производительность, гибкость и способность восстановления после сбоев, поддержка восстановления данных с использованием снимков.

Особенности и преимущества файловых систем.

NTFS: Особенности: разграничение прав доступа (ACL), поддержка журналирования, шифрование файлов и папок, поддержка файлов большого размера и разделов.

Преимущества: высокая совместимость с операционными системами Windows, хорошая поддержка безопасности и прав доступа, поддержка сжатия данных.

APFS: Особенности: копирование на запись (Copy-on-Write), поддержка снимков (Snapshots), шифрование файлов и метаданных, сжатие данных для экономии места на диске.

Преимущества: высокая производительность, эффективное использование пространства на диске, поддержка быстрого восстановления после сбоев.

FAT32: Особенности: простая структура, поддержка совместимости с различными операционными системами.

Преимущества: широкая совместимость с разными платформами, возможность чтения и записи на различных операционных системах.

При выборе файловой системы для установки Linux важно учитывать ваши потребности, требования к производительности, масштабируемости, надежности и функциональности. Если вы не уверены, какую файловую систему выбрать, часто рекомендуется использовать ext4, так как она является хорошим общим выбором.

Путь до файла или каталога — это набор символов, показывающий расположение файла или каталога в файловой системе.

Путь может быть **полным (абсолютным)** — это путь, который указывает на одно и то же место в файловой системе, вне зависимости от текущего рабочего каталога. Полный путь всегда начинается с корневого каталога, например `/usr/local/bin/`.

Путь может быть также **относительным** — это путь по отношению к текущему рабочему каталогу пользователя.

Команды

pwd (print working directory) покажет текущий каталог (каталог, в котором мы сейчас находимся) и полный путь. Команда необходима, чтобы понять, в каком месте файловой системы мы находимся.

Команды

cd (change directory) — перемещение между каталогами. Эта команда позволит нам сменить текущую директорию, используя полный или относительный путь:

- используем полный путь: `cd /usr/local/bin;`
- используем относительный путь: `cd Загрузки;`
- быстро вернуться в домашний каталог: `cd ~`

ls -l покажет подробный список содержимого. Сюда будут включены дата изменения, владелец и группа владельца, права и другие свойства файлов или каталогов в директории.

ls -a покажет скрытые файлы и каталоги. В Unix-подобных системах такие файлы и каталоги начинаются с точки. Этот параметр очень часто используют в сочетании с параметром -l, например `ls -al /home/user`.

Команда `cp` (`copy`): `cp file1 file2` — копирование файлов или каталогов. При операции копирования можно использовать как полный, так и относительный путь.

Копирование директорий

Поскольку директория может содержать поддиректории, необходимо использовать параметр **`-r`** (рекурсивно).

Например, **`cp -r /dir1 .`** скопирует каталог `/dir1` в текущую директорию.

```
mv /home/user/file /home/user1/file
```

переместит файл из каталога
/home/user в каталог /home/user1.

Команда **mv**, если её применить к файлу или каталогу в текущей директории, переименует файл или каталог. Например: `mv file1 file2`, `mv dir1 dir2`.

В случае с каталогами операция `mv` не требует параметра `-r`, поскольку никак не воздействует на поддиректории.

Команда rm (remove) — удаление файлов или каталогов. Например, `rm file1` удалит файл. Для удаления каталогов необходимо использовать параметр `-rf` (recursive, forced) — удалить со всем содержимым, не спрашивая подтверждения.

Внимание! Операция удаления — необратимое действие. Debian-подобные дистрибутивы не спрашивают подтверждения действия. Ошибочное удаление файлов или каталогов может привести к неработоспособности системы.

1. Команда **touch** создаст пустой файл.
2. Команда **mkdir** создаст каталог. В некоторых дистрибутивах — `md`, `make directory`.
3. Команда **cat** (`catenate`) позволяет быстро прочитать содержимое файла, а также склеить несколько файлов в один.

4. Программы постраничного просмотра текста: **less** и **more**.
5. Команда **tail** позволит вывести на экран заданное количество строк от конца файла или содержимое файла в режиме интерактивного просмотра.

3. Знакомство с текстовыми редакторами.

Текстовые редакторы vi/Vim

Режимы работы редактора

Командный режим:

- навигация по файлу;
- удаление символов или строк;
- копирование/вставка.

Режим редактирования:

ввод и редактирование текста

Режим последней строки:

- сохранение файла;
- выход из редактора;
- поиск по тексту.

Командный режим: навигация по строкам осуществляется либо при помощи клавиш «Вверх», «Вниз», «Влево», «Вправо», либо используя буквы: k — вверх, j — вниз, h — влево, l — вправо.

Удаление символа под курсором — клавиша x. Удаление строки: dd (дважды нажать клавишу d).

Удаление строки: dd (дважды нажать клавишу d).

Пролистывание страниц: клавиши PgUp, PgDn или комбинации Ctrl + F, Ctrl + B.

Режим редактирования

Для перехода в режим редактирования используется одна из команд:

- клавиша **i** начнёт редактирование строки с текущего положения курсора;
- клавиша **a** начнёт редактирование строки со следующего после курсора символа;
- клавиша **o** начнёт редактирование текста со следующей строки.

Режим последней строки: после окончания работы с текстом нажимаем клавишу Esc (выходим из режима редактирования). Далее нажимаем клавишу «:» (двоеточие) и получаем в последней строке редактора приглашение к вводу команд:

- q — выйти из редактора;
- q! — выйти из редактора без сохранения изменений;
- w — сохранить файл;
- wq — сохранить и выйти из редактора.

После ввода нужной команды нажимаем клавишу Enter.

Текстовый редактор nano

Сразу после запуска открытый файл становится доступным для редактирования. В отличие от редактора vim, у nano нет каких-то особых режимов работы.









Ниже области редактирования располагается панель с наиболее популярными комбинациями клавиш управления редактором. ^ означает клавишу Ctrl. Сохранение файла — комбинация клавиш Ctrl + o, выход из редактора — Ctrl + x.

```
##
## This string is VERY resource intensive!!!
# color brightyellow start="\"(\.|\.[^\"])*\\[[[:space:]]*$" end="\"^(\.|\.[^\"])*\""
##
## And we want to have some nice comment highlighting too
# color brightblue "//.*"
# color brightblue start="/\*" end="\*/"

## Here is a short example for HTML
syntax "HTML" "\.html$"
color cyan start="<" end=">"
color red "&[^\n;[[[:space:]]]*;"

# Generic conf/rc/sh file syntax highlighting
#####
syntax "conf/rc/sh" "conf$|rc$|sh$"
color white ".+"
color green "^#.*"

## Here is a short example for TeX files
# syntax "TeX" "\.tex$"
# color green "\.|\.[A-Za-z]*"
# color magenta "[{}]"
```

 Помощь Записать Читать файл Пред Стр Вырезать ТекПозиция Выход Выводить Поиск След Стр Отм Вырез Орфография

MC (Midnight Commander) - это текстовый файловый менеджер для командной строки, который предоставляет удобный и интуитивно понятный интерфейс для навигации по файловой системе и выполнения различных операций с файлами и папками.

MC предлагает двухпанельный интерфейс, где каждая панель отображает содержимое отдельной директории. Вы можете перемещаться по файловой системе, выделять файлы и папки, копировать, перемещать, переименовывать и удалять их, а также выполнять различные другие операции. MC также предоставляет возможность работы с архивами, просмотра и редактирования файлов, а также выполнения команд во встроенном терминале.

Для запуска MC в командной строке достаточно ввести команду **mc**. После запуска вы увидите двухпанельный интерфейс, где вы можете перемещаться с помощью клавиш со стрелками, используя клавиши Tab и Enter для выбора файлов и папок, а также используя различные функциональные клавиши для выполнения операций.

| Left | File | Command | Options | Right | | |
|------------------------------|-------------|--------------|---------|----------------------|-------------|--------------|
| <- /usr/src/boot-floppies v> | | | | <- /usr/src/linux v> | | |
| Name | Size | MTime | | Name | Size | MTime |
| /.. | 4096 | Nov 25 16:22 | | /.. | 4096 | Nov 25 16:22 |
| /debian | 4096 | Nov 20 03:31 | | /Documentation | 4096 | Nov 18 01:43 |
| /documentation | 8192 | Nov 20 23:15 | | /arch | 4096 | May 12 2000 |
| /i386-specials | 4096 | Nov 19 23:10 | | /drivers | 4096 | Nov 25 16:41 |
| /m68k-specials | 4096 | Nov 19 23:10 | | /fs | 4096 | Nov 25 16:48 |
| /powerpc-specials | 4096 | Nov 19 23:10 | | /include | 4096 | Nov 25 16:42 |
| /scripts | 4096 | Nov 19 23:10 | | /init | 4096 | Nov 25 16:43 |
| /updates | 57344 | Nov 21 14:36 | | /ipc | 4096 | Nov 25 16:50 |
| /utilities | 4096 | Nov 21 14:39 | | /kernel | 4096 | Nov 25 16:43 |
| Maintainers | 1193 | Feb 5 2000 | | /lib | 4096 | Nov 25 16:50 |
| Makefile | 40732 | Nov 21 00:04 | | /m | 4096 | Nov 25 16:46 |
| README | 2470 | Feb 28 2000 | | /net | 4096 | Nov 25 16:50 |
| README-CVS | 3747 | Oct 1 20:44 | | /scripts | 4096 | Nov 25 16:42 |
| README-Overview | 4142 | Apr 17 2000 | | .config | 18436 | Nov 25 16:39 |
| README-T~slators | 3736 | Nov 15 18:49 | | .config.old | 18271 | Nov 25 16:39 |
| /.. | | | | /.. | | |

Hint: There is a Midnight Commander URL: <http://www.gnome.org/mc/>

fr@surimi:/usr/src/linux\$

1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit

Встроенный редактор в Midnight Commander (MC) называется "mcedit". Он предлагает простой и удобный текстовый интерфейс для просмотра и редактирования файлов.

Чтобы открыть файл в редакторе MC, следуйте этим шагам:

Запустите Midnight Commander, введя команду `mc` в командной строке.

Навигируйте к файлу, который вы хотите отредактировать, используя двухпанельный интерфейс MC.

Выделите файл, который хотите открыть, и нажмите клавишу F4 или Enter. Это откроет выбранный файл в редакторе mcedit.

После открытия файла в редакторе mcedit вы можете просматривать и редактировать его содержимое. Вот некоторые основные команды редактора:

Ctrl+S или F2: Сохранить файл.

Ctrl+Q или F10: Выйти из редактора без сохранения.

Ctrl+R: Перезагрузить файл с диска.

Ctrl+F или /: Поиск текста в файле.

Ctrl+G: Показать информацию о файле (размер, права доступа и т. д.).

Ctrl+X Ctrl+M: Показать список доступных макросов.