Министерство науки и высшего образования Российской Федерации

Рязанский государственный радиотехнический университет им. В.Ф. Уткина

Поиск файлов в Linux

Методические указания к практическим занятиям

Рязань 2020

УДК 681.3.06

Поиск файлов в Linux: методические указания к практическим занятиям / Рязан. гос. радиотехн. ун-т.; сост. А.А. Митрошин, В.Г. Псоянц. – Рязань, 2020. – 16 с.

Содержат описание практического занятия, используемого в курсе «Операционная система Linux». Могут использоваться при изучении других курсов, связанных с операционной системой Linux.

Предназначены для студентов очной, заочной и очно-заочной форм обучения направления подготовки «Информатика и вычислительная техника». Могут использоваться для студентов других направлений подготовки.

Могут использоваться как методические указания к лабораторным работам в курсах, связанных с изучением операционной системы Linux и свободно распространяемого программного обеспечения.

Ил. --. Библиогр.: -- назв.

Операционная система Linux, команда find, команда xargs, команда grep

Печатается по решению редакционно-издательского совета Рязанского государственного радиотехнического университета.

Рецензент: кафедра САПР вычислительных средств Рязанского государственного радиотехнического университета (зав. кафедрой засл. деят. науки и техники РФ В.П.Корячко)

Поиск файлов в Linux

Составители: Митрошин Александр Александрович

Псоянц Владимир Грикорович

Редактор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Корректор \_\_\_\_\_\_\_\_\_\_\_\_\_

Подписано в печать \_\_\_\_\_\_\_\_. Формат бумаги 60×84 1/16.

Бумага газетная. Печать трафаретная. Усл. печ. л. 1,0.

Уч-изд. л. 1,0. Тираж 50 экз. Заказ

Рязанский государственный радиотехнический университет.

390005, Рязань, ул. Гагарина, 59/1.

Редакционно-издательский центр РГРТУ.

***Утилита find***

Утилита find является одной из самых полезных утилит Linux. Она находит файлы, удовлетворяющие заданному набору параметров, таких как имя файла, дата его модификации, режима доступа, владельца и т.д.

Общий формат команды:

**find путь –опции**

где путь – имя каталога, с которого необходимо начать поиск (символ «.» служит для обозначения текущего каталога, «/» - корневого каталога, «~» - домашнего каталога текущего пользователя), а опции сообщают команде, какие файлы интересуют пользователя.

Основные опции утилиты find приведены в табл. 1.

Таблица 1. Основные опции утилиты find

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| -name имя\_файла | Поиск файлов, имена которых соответствуют заданному шаблону |
| -print | Вывод полных имен заданных файлов в стандартный поток вывода |
| -perm режим | Поиск файлов, для которых установлен указанный режим доступа |
| -prune | Применяется для того, чтобы find не выполняла рекурсивного поиска. Если указана опция –depth, то опция –prune ингорируется |
| -user владелец | Поиск файлов, принадлежащих указанному пользователю |
| -group группа | Поиск файлов, принадлежащих указанной группе |
| -mtime –n +n | Поиск файлов, содержание которых модифицировалось менее чем (-) или более чем (+) n дней назад. Имеются также опции –atime и –ctime, которые позволяют осуществлять поиск файлов по дате последнего чтения и дате последнего изменения файла |
| -nogroup | Поиск файлов, принадлежащих несуществующей группе |
| -nouser | Поиск файлов, принадлежащих несуществующему пользователю |
| -newer файл | Поиск файлов, которые созданы позднее, чем указанный файл |
| -type с | Поиск файлов определенного типа, а именно:  b – специальный блочный файл;  d – каталог;  c – специальный символьный файл;  p – именованный канал;  l – символическая ссылка;  s – сокет;  f – обычный файл. |
| -size n | Поиск файлов, размер которых составляет n единиц. Возможны следующие варианты единиц измерения:  b – блок размером 512 байт (установка по умолчанию);  c – байт;  k – килобайт (1024 байт);  w – двухбайтовое слово. |
| -depth | При поиске файлов сначала просматривается содержимое существующего каталога и лишь затем проверяется запись, соответствующая самому каталогу |
| -fstype | Поиск файлов, которые находятся в файловой системе определенного типа. Обычно соответствующие сведения хранятся в файле /etc/fstab, который содержит сведения о файловых системах, используемых на компьютере |
| -mount | Поиск файлов только в текущей файловой системе. Аналогом этой опции является опция –xdev. |
| -exec команда | Выполнение команды интерпретатора shell для всех обнаруженных файлов. Выполняемые команды имеют формат  команда {} \;  (между символами «{}» и «\;» - пробел) |
| -ok команда | Аналогична опции –exec, но перед выполнением команды отображается запрос. Эта опция полезна для тестирования команды find. |

Часто возникает необходимость в поиске файлов, отвечающих сразу нескольким критериям. Поэтому необходим механизм объединения сразу нескольких параметров. Приемы объединения параметров приведены в табл. 2.

Таблица 2. Объединение нескольких параметров поиска

|  |  |
| --- | --- |
| оператор\_1 –a оператор\_2 | Поиск файлов, соответствующих как оператор\_1, так и оператор\_2. |
| оператор\_1 –o оператор\_2 | Поиск файлов, соответствующих *или* оператор\_1 *или* оператор\_2 |
| ! оператор | Поиск всех файлов, не соответствующих заданному оператору. Символ «!» означает логическое НЕ. |
| \(выражение)\ | Изменение порядка выполнения операторов. В сложных выражениях, прежде всего определяется значение выражения. |

Достаточно часто про вызове find забывают указать опцию –print. Без этой опции утилита может не отображать имен файлов.

***Поиск файлов по шаблону имени***

На рис. 1 приведены примера использования find для поиска файлов по шаблону имени. Для поиска по шаблону имени используется опция –name (name - имя).

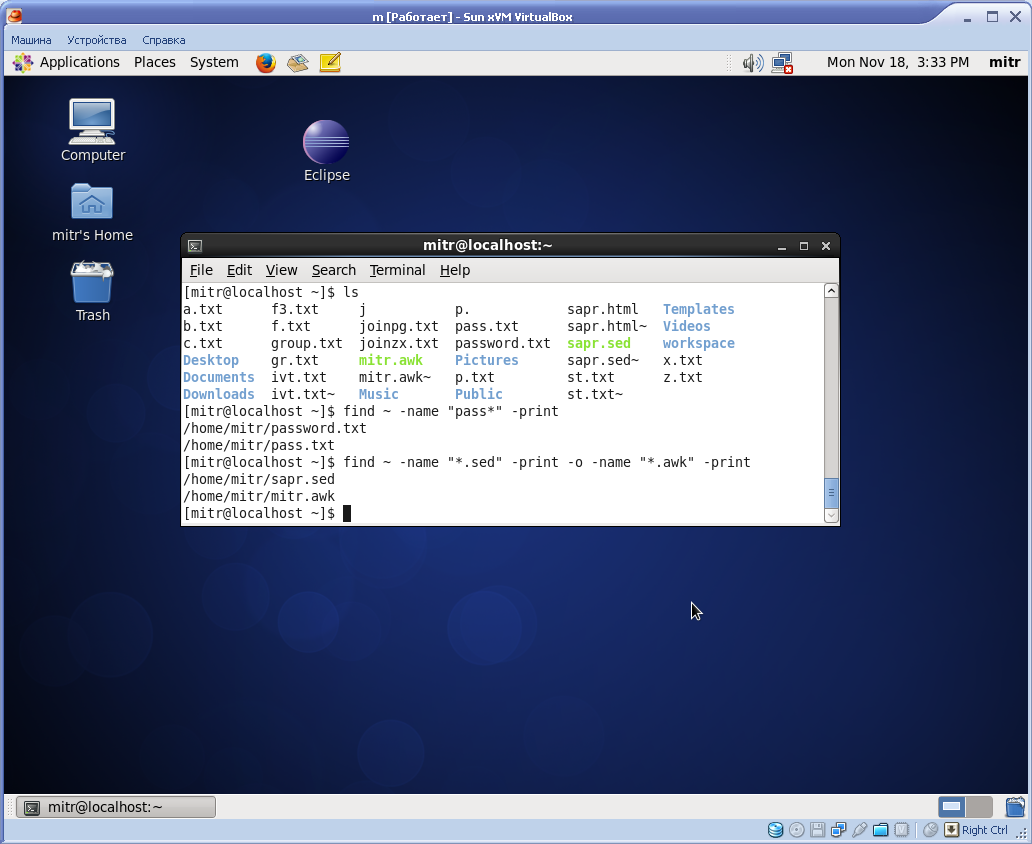


Рисунок 1. Поиск файлов по шаблону имени

Сначала с помощью команды **ls** просматривается содержимое текущего каталога, который в данном случае совпадает с домашним каталогом текущего пользователя.

Затем с помощью команды

**find ~ -name “\*pass” –print**

выводятся полные имена всех файлов, имена которых начинаются на «pass». Поиск производится начиная с каталога ~, то есть с домашнего каталога текущего пользователя.

Далее с помощью команды

**find ~ -name “\*.sed” –print –o -name “\*.awk” –print**

выводятся полные имена файлов, имена которых заканчиваются на «.sed» или (опция -o) на «.awk».

***Поиск файлов с заданным режимом доступа***

Для поиска с помощью find файлов с заданным режимом доступа используется опция –perm (permission - разрешение).

Найдем все файлы с режимом доступа 666, как это показано на рис. 2.

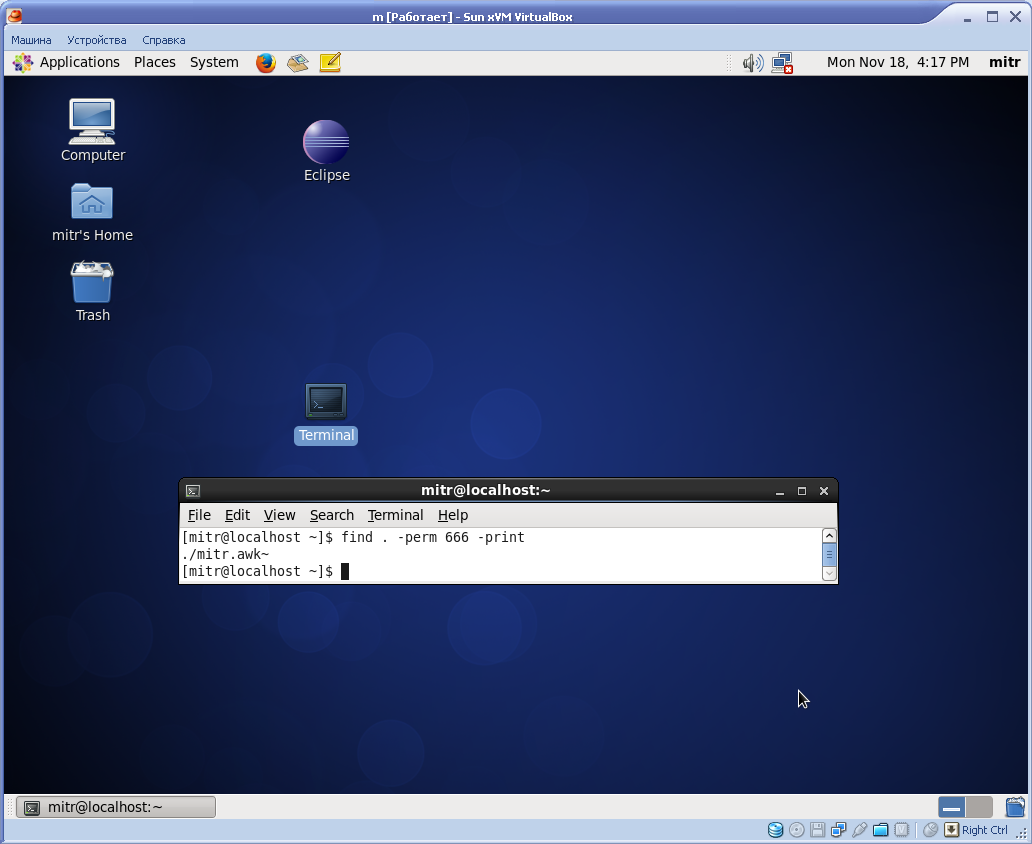


Рисунок 2. Вывод полных имен всех файлов текущего каталога с режимом доступа 666

Если перед значением режима поставить дефис «-» то, будет произведен поиск файлов, для которых установлены все указанные биты разрешений, остальные биты при этом игнорируются.

Если перед значением режима поставить знак «+», осуществляется поиск файлов, для которых установлен хотя бы один из указанных битов разрешений, при этом остальные биты разрешений игнорируются.

Пусть необходимо найти файлы с правом на запись для группы пользователя. Для этого нужно найти соответствие шаблону режима ....w..... Существует много комбинаций, которые подходят под этот шаблон, например ----w-rwx или r---w-rwx. Конечно, можно попытаться перечислить все возможные комбинации, но можно просто поставить перед восьмеричным значением шаблона знак «-».

Бит права записи в файл для группы пользователя соответствует восьмеричному числу 20 (000 010 000 = 24 = 1610 = 208), поэтому команда

**find ~ –perm -20 –print**

найдет файлы с правами доступа, перечисленными ниже:

rwxrwxrwx

rwxrwxr-x

rw-rw-rw-

rw-rw-r--

rw-rw----

Чтобы найти все файлы, которые можно исполнять (сценарии или программы) остальным пользователям можно задать шаблон режимов доступа ..x...... (26=6410=1008) с помощью команды

**find ~ –perm -100 –print**

Если же необходимо найти файлы, которые могут исполнять *или* владелец *или* группа владельца *или* остальные, то шаблон режима будет ..x..x..x , что соответствует восьмеричному (20+23+26 = 1+8+64 = 7310=1118). Тогда для поиска таких файлов можно использовать команду

**find ~ –perm +111 –print**

***Поиск файлов по имени владельца или группы***

Для поиска файлов по имени владельца используется опция –user. Пример поиска файлов по имени владельца:

**find ~ -user “mitr” –print**

Для поиска файлов по имени группы используется опция –group. Пример поиска файлов по имени группы:

**find ~ -group “mitr” –print**

Часто, когда учетная запись пользователя удаляется администратором, файлы пользователя еще остаются в системе. Чтобы найти такие файлы используется опция –nouser. Пример поиска файлов, собственник которых удален из системы:

**find ~ -nouser –print**

Опция –nogroup позволяет искать файлы несуществующих групп.

***Поиск файлов по типу***

Если необходимо найти только файлы определенного типа, то для их поиска используется опция –type тип. Тип файла может быть следующим:

b – специальный блочный файл;

d – каталог;

c – специальный символьный файл;

p – именованный канал;

l – символическая ссылка;

s – сокет;

f – обычный файл.

Для того, чтобы найти все каталоги, начиная с домашнего каталога, можно воспользоваться командой

**find ~ -type d –print**

***Поиск файлов по размеру***

Для поиска файлов по размеру используется опция –size n, где n – количество единиц в некоторых единицах измерения. Возможны следующие варианты единиц измерения:

b – блок размером 512 байт (установка по умолчанию);

c – байт;

k – килобайт (1024 байт);

w – двухбайтовое слово.

При поиске файлов, размеры которых лежат в определенном диапазоне, перед числом можно поставить знаки «-» или «+». Знак «-» означает «меньше чем», а знак «+» - «больше чем». Команда

**find ~ -size -100c –size +10c –print > size.txt**

сохраняет в файле size.txt полные имена всех файлов, пути которых начинаются с домашнего каталога работающего пользователя и размер которых менее 100 байт, но более 10 байт.

***Опция -depth***

Опция –depth позволяет организовать поиск таким образом, что сначала проверяются все файлы текущего каталога и рекурсивно все его подкаталоги и только в конце – сам каталог. Опция широко применяется при создании списка файлов, помещаемых в архив с помощью команд tar или cpio, так как в этом случае сначала записывается образ каталога и лишь после этого задаются права доступа к нему. Это позволяет архивировать те каталоги, для которых у пользователя нет разрешения на запись.

***Опция -prune***

Опция –prune прекращает поиск, выполняемый командой find, на текущем путевом имени. Если текущее путевое имя принадлежит каталогу, то find не будет в него заходить для дальнейшего поиска.

***Поиск файлов по времени доступа***

Опция –mtime применяется при поиске файлов, доступ к которым осуществлялся x дней назад. Если аргумент опции снабдить знаком «-», то будут найдены файлы, к которым не было доступа в течении x дней. При использовании аргумента со знаком «+» производится поиск файлов, доступ к которым осуществлялся на протяжении последних x дней.

Команда

**find ~ -mtime +2 –print**

выведет все файлы, доступ к которым осуществлялся в течение последних 2 дней.

Опция –newer позволяет найти файлы, доступ к которым осуществлялся в промежутке времени между обновлением двух заданных файлов. Общий формат команды find с опцией –newer следующий:

**find –newer старый\_файл ! –newer новый\_файл**

Символ «!» является оператором логического отрицания. Он означает, что надо найти файлы новее, чем старый\_файл, но старше, чем новый\_файл.

***Выполнение действий с найденными файлами***

Для выполнения действий с найденными файлами используются опции –exec и –ok.

После опции –exec следует указать выполняемую команду, после чего ввести фигурные скобки, пробел, слеш и точку с запятой:

**find ~ -type f –exec ls –l {} \;**

Пример использования команды приведен на рис. 3. Здесь выполняется поиск обычных файлов, список которых отображается с помощью команды ls -l.

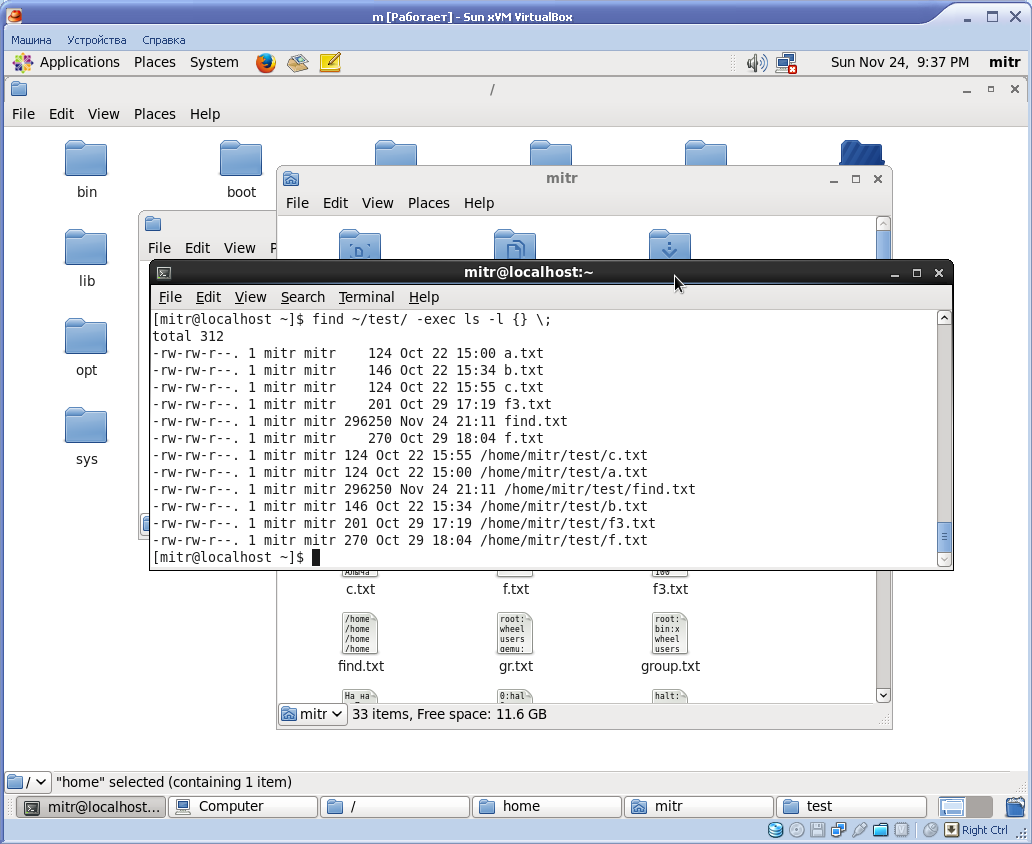


Рисунок 3. Пример использования команды find с опцией –exec

Чтобы удалить в этом каталоге файлы, которые не обновлялись в этом каталоге в течении 3 дней, можно выполнить команду, представленную на рис. 4.

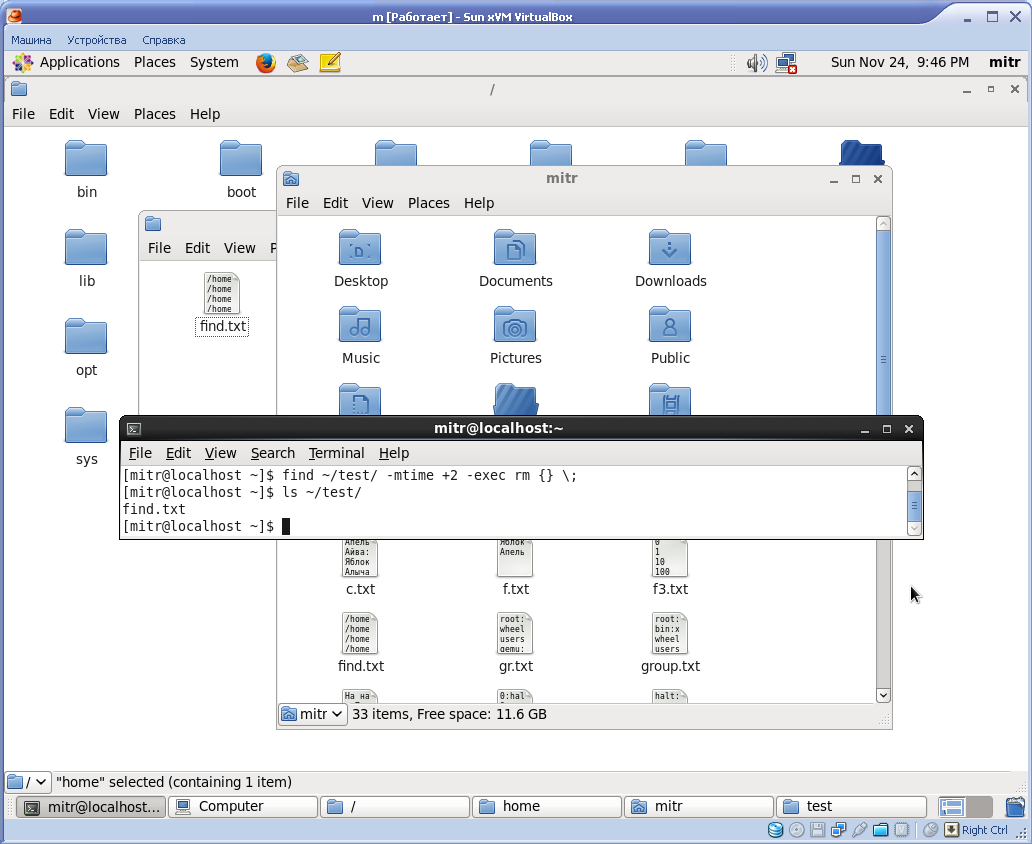


Рисунок 4. Удаление файлов с использованием опции –exec

При перемещении и удалении файлов, найденных с помощью команды find желательно быть осторожным. Лучше вместо опции -exec использовать опцию –ok, которая отличается от –exec тем, что просит подтвердить каждую выполняемую операцию. Пример использования опции –ok приведен на рис. 5.

Для удаления предлагаемого в результате поиска файла необходимо ввести y, для предотвращения удаления – n.

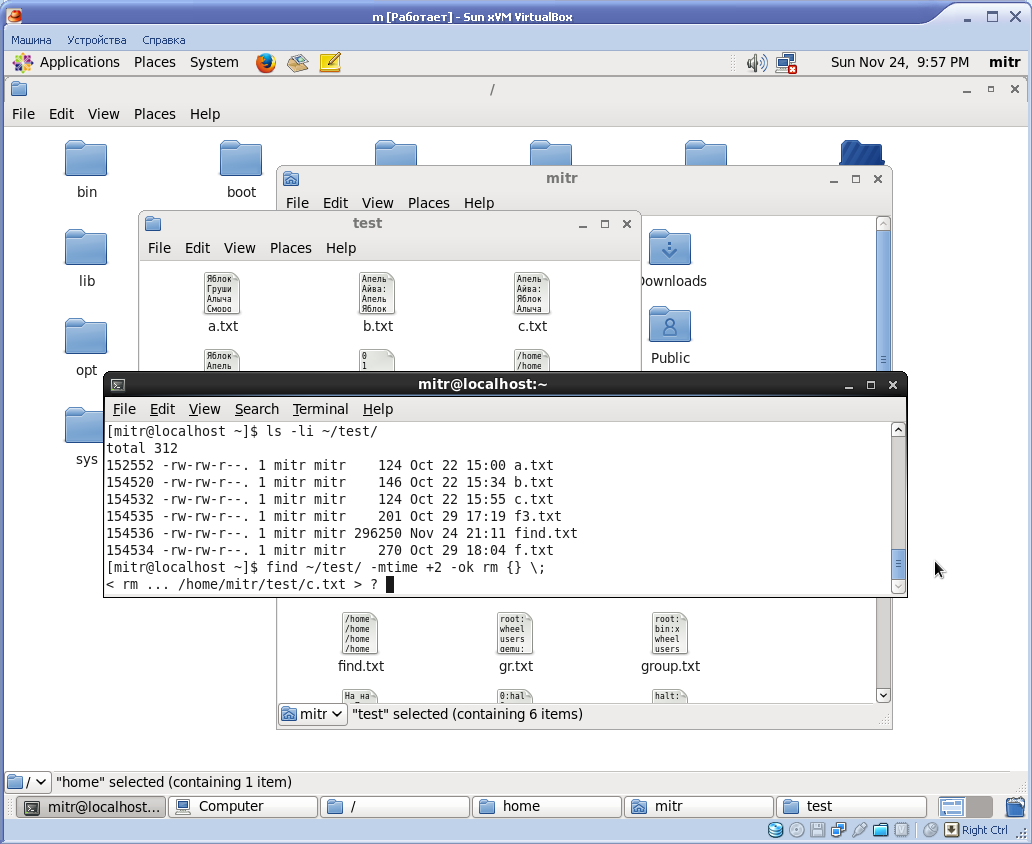


Рисунок 5. Пример использования команды find с опцией -ok

***Команда xargs***

При наличии опции –exec команда find передает указанной команде все найденные файлы. Если в системе длина командной строки ограничена, то передаче большого количества имен файлов может появляться сообщение об ошибке. В этом случае можно использовать команду xargs, которая будет передавать имена файлов не все сразу, а порциями.

Например, следующая команда возвращает список всех файлов, начиная с текущего каталога, а команда xargs выполняет для них команду file, определяющую тип каждого файла:

**find . –type f –print | xargs file**

Подробнее с командой xargs можно познакомиться, используя справочную систему.

***Команда grep***

Основной функцией команды grep (global regular expession print – печать глобальных регулярных выражений) является поиск и вывод строк, соответствующих регулярному выражению.

Общий формат команды grep:

**grep [параметры] регулярное\_выражение [файлы]**

В качестве регулярного выражения может быть указана обычная строка. Если файл не задан, то текст берется из стандартного входного потока.

Строку, которая задана в качестве регулярного выражения и состоит из нескольких слов, необходимо заключать в двойные кавычки, в противном случае первое слово будет восприниматься как образец для поиска, а оставшиеся рассматриваться как имена файлов. Если в шаблоне используется имя системной переменной, то оно также заключается в двойные кавычки. Если в состав шаблона входят метасимволы, то шаблон заключается в одинарные кавычки.

Основные опции команды grep представлены в табл. 3.

Таблица 3. Основные опции команды grep

|  |  |
| --- | --- |
| **Опция** | **Описание** |
| -c | Задает отображение только числового значения, указывающего, сколько строк соответствует шаблону |
| -i | Игнорировать регистр символов |
| -h | Подавляет вывод имен файлов, включающих найденные строки (по умолчанию в выводе grep каждой строке предшествует имя файла, в котором она содержится) |
| -l | Отображает только имена файлов, содержащих найденные строки |
| -n | Задает нумерацию выводимых строк |
| -s | Подавляет вывод сообщений о несуществующих или нетекстовых файлах |
| -v | Задает отображение строк, не соответствующих шаблону |
| -r | Определяет рекурсивный обход всего дерева указанного каталога. |

Если не указано имени файла, то команда обрабатывает стандартный ввод, например строки, набранные на клавиатуре, как это показано на рис. 6. При нажатии клавиши Enter для перехода на новую строку, grep выводит строки, содержащие шаблон «Маша», отсюда и удвоение этих строк.

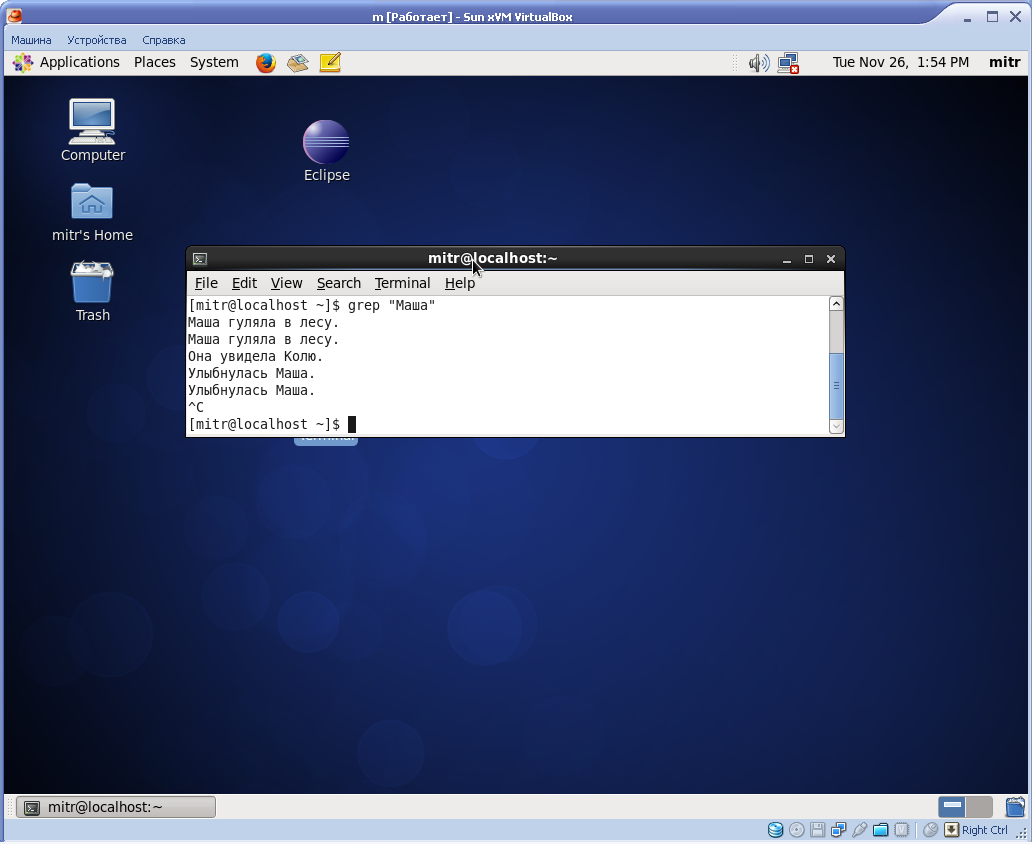


Рисунок 6. Обработка grep стандартного входного потока

Если во всех файлах текущего каталога необходимо найти фразу «sapr vs», то это можно сделать с помощью команды

**grep “sapr vs” \***

Опция –c позволяет определить, сколько строк соответствует заданному шаблону.

На рис. 7 приведены примеры использования grep без опции –c и с опцией -c.

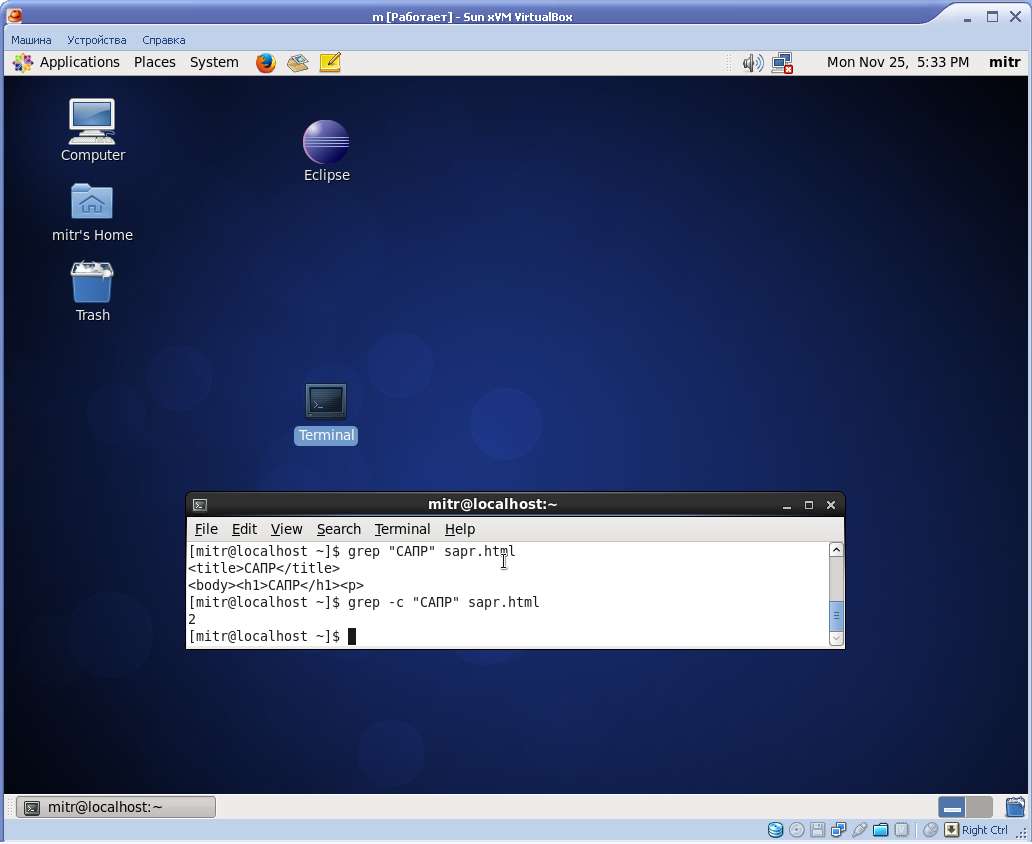


Рисунок 7. Вывод команды grep без опции –c и с опцией –c

С помощью опции –n выводимые строки можно перенумеровать, как это показано на рис. 8. Теперь можно установить, в строках с какими номерами произошло совпадение с шаблоном.

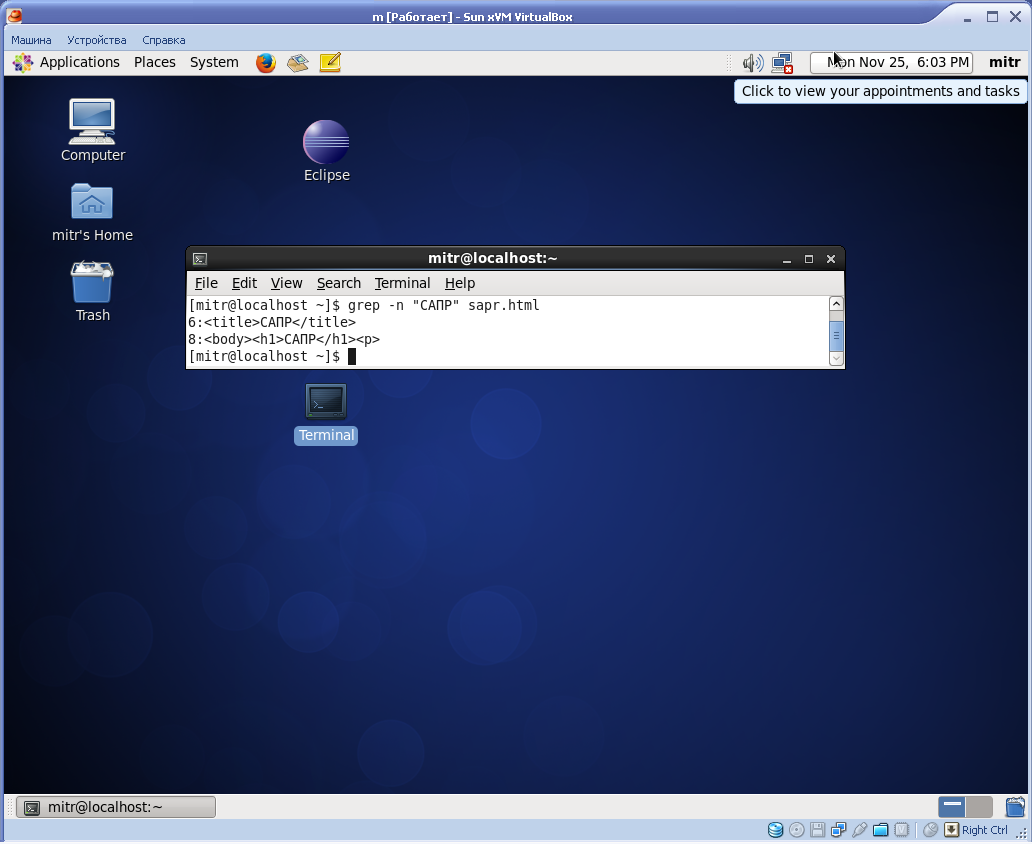


Рисунок 8. Вывод команды grep с опцией –n

Опция –v используется в тех случаях, когда нужно отобрать строки, не соответствующие шаблону (рис. 9).

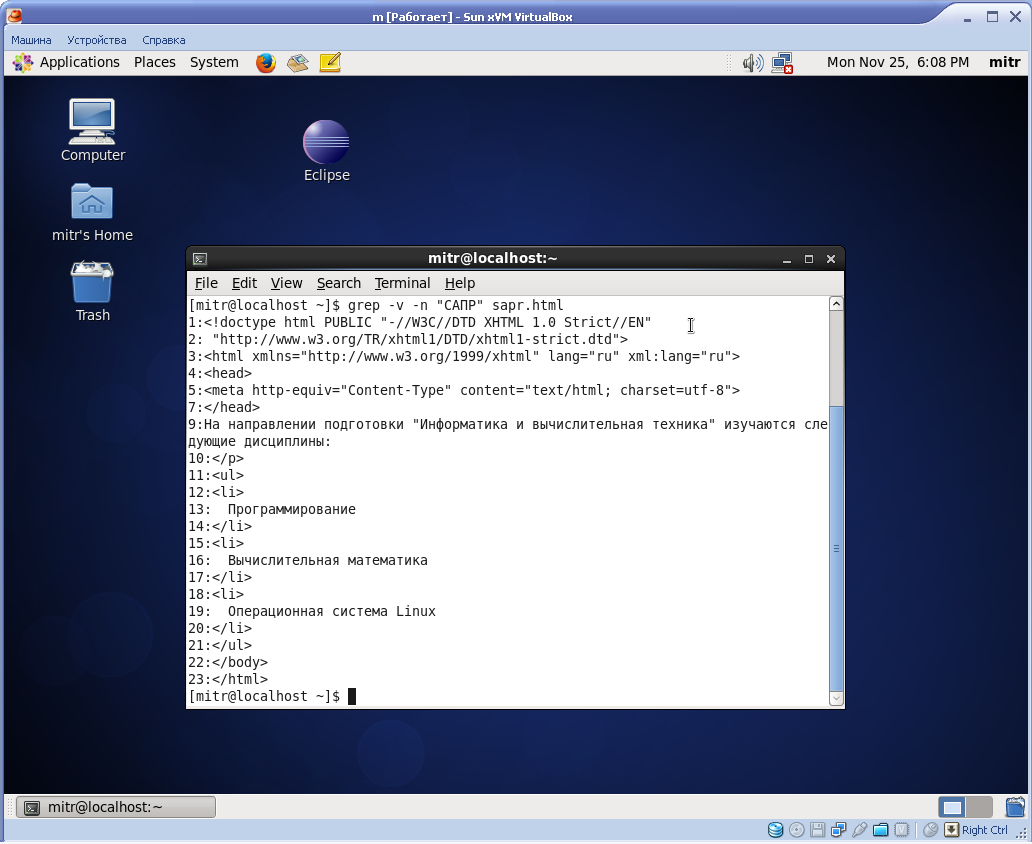


Рисунок 9. Строки, не соответствующие шаблону

Опция –r позволяет рекурсивно обходить дерево каталогов, начиная с текущего каталога, как это показано на рис. 10.

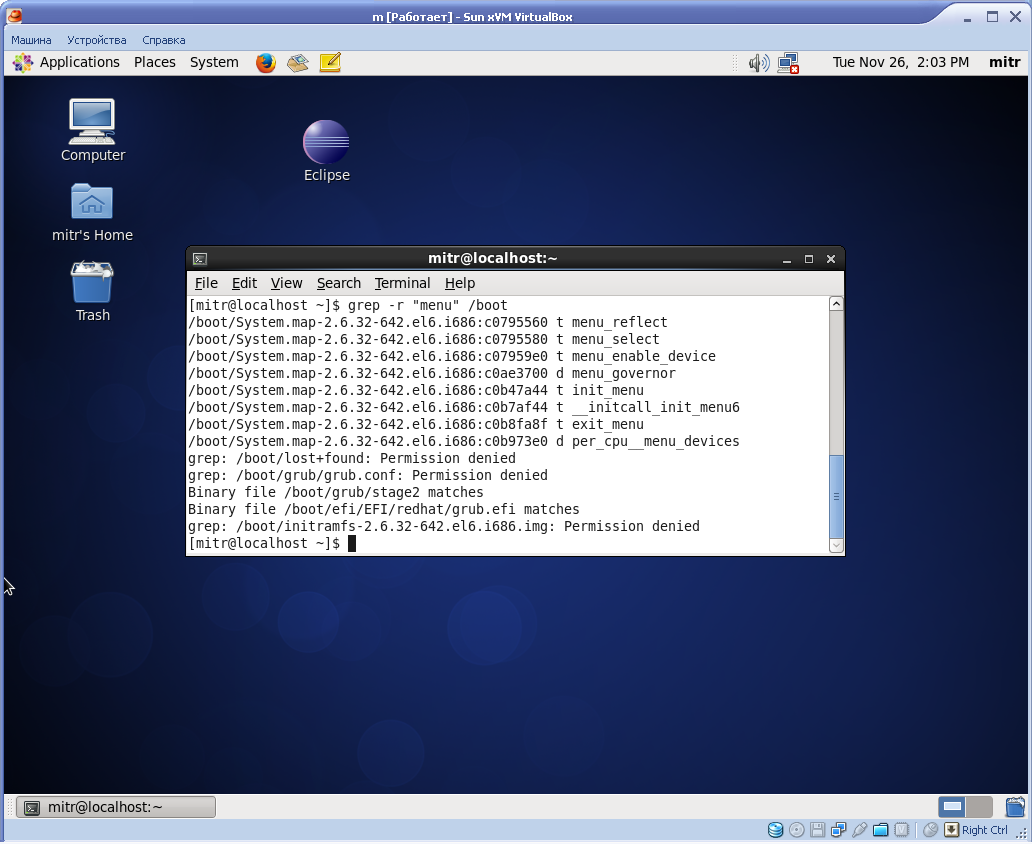


Рисунок 10. Рекурсивный обход дерева каталогов

С помощью регулярных выражений можно определять более сложные критерии поиска строк. При работе с регулярными выражениями шаблон поиска необходимо заключать в одиночные кавычки ' ', что позволит защитить все встречающиеся в нем специальные символы от интерпретатора shell, который в противном случае может перехватить их у grep.

Опция --color позволяет выделить цветом вхождения в выводе grep. Пример такого выделения приведен на рис. 11.

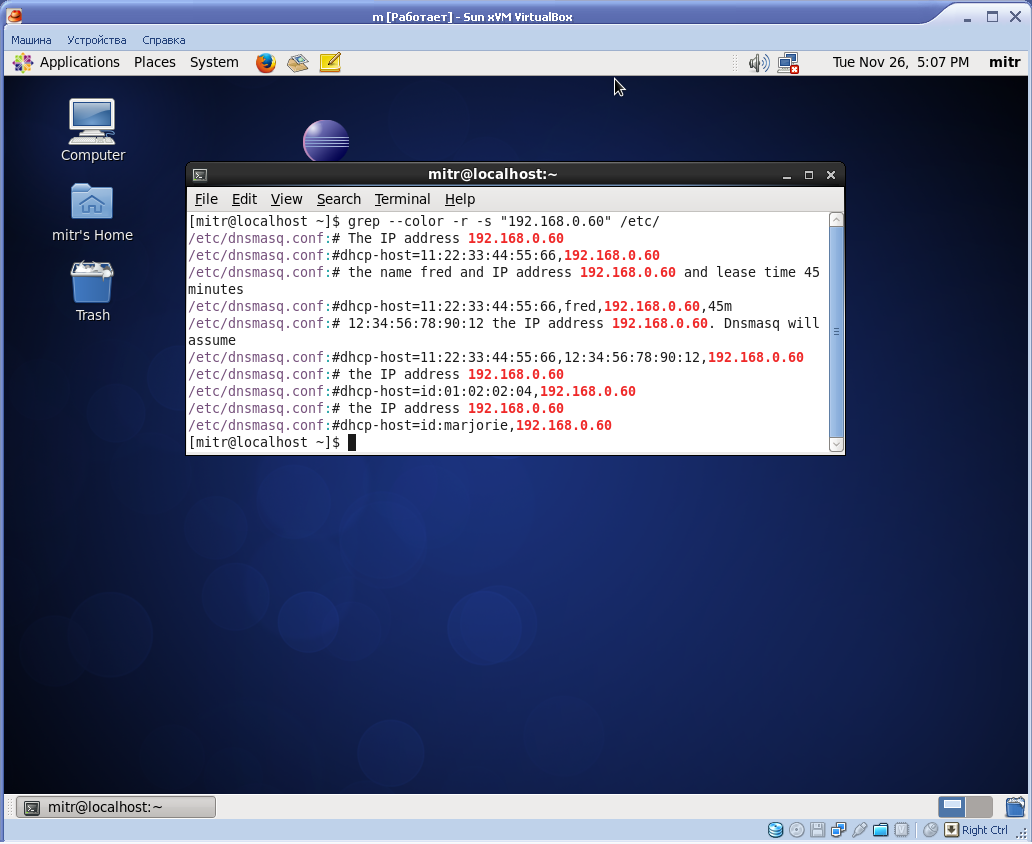


Рисунок 11. Выделение цветом в выводе grep

***Использование grep совместно с другими командами***

Команда grep часто используется как фильтр для обработки выходных данных других программ. Далее приводятся примеры использования grep с другими программами.

Например, если необходимо найти файлы по дате его последней модификации, то можно воспользоваться командой, показанной на рис. 12.

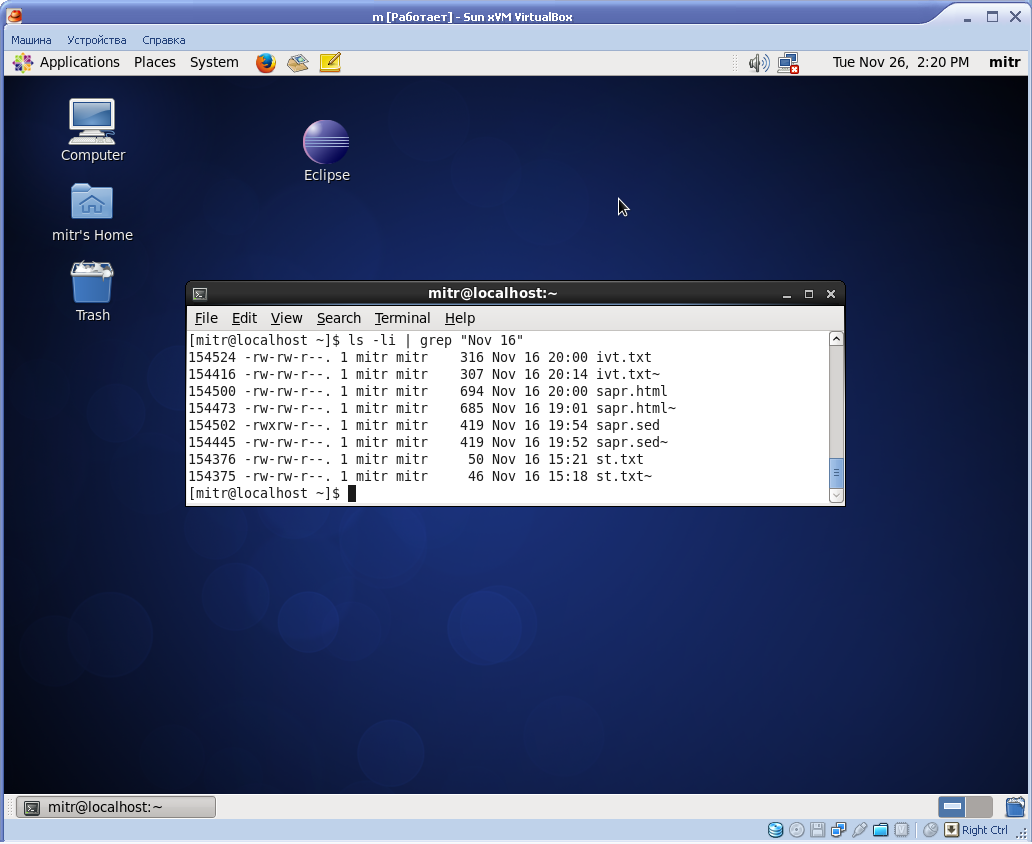


Рисунок 12. Поиск файлов по дате последней модификации

Команда grep очень полезна при просмотре конфигурационных файлов и логов. Частым примером использования команды grep стал программный канал с командой dmesg, которая выводит сообщения ядра во время загрузки компьютера (рисунок 13).

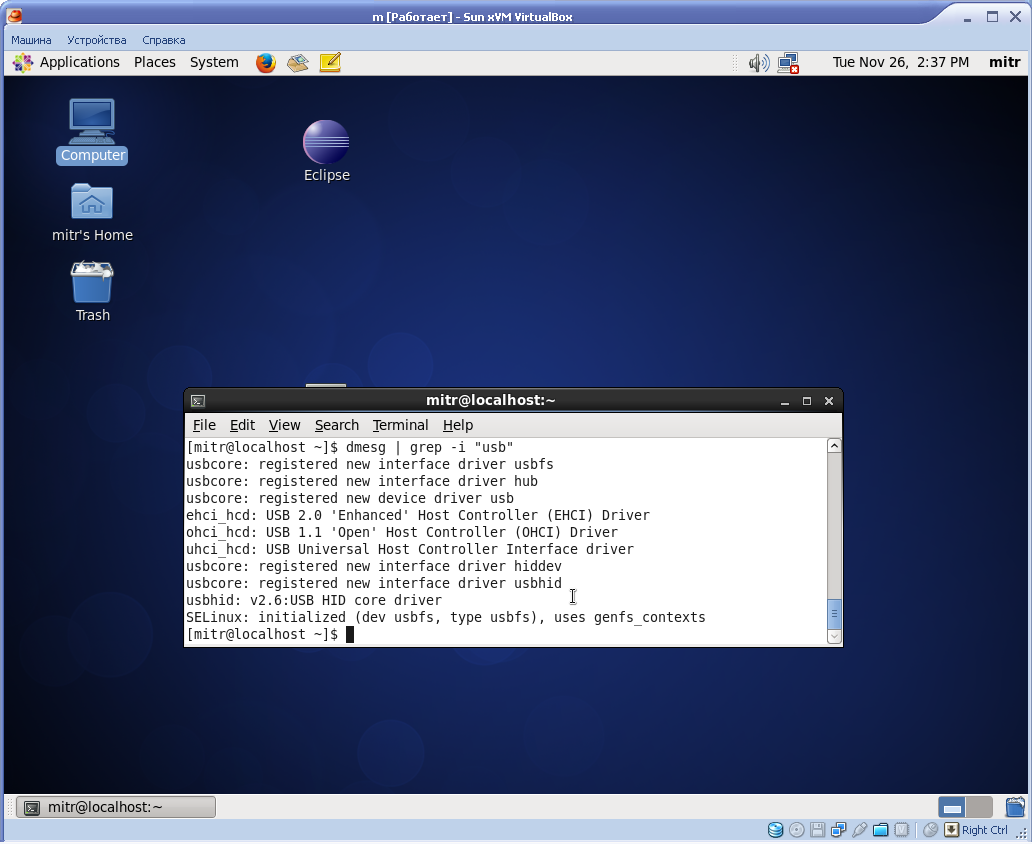


Рисунок 13. Использование grep вместе с dmesg

***Порядок выполнения работы***

1. Изучите теоретический материал.

2. Повторите вызовы команд, представленных в методическом пособии.

3. Выполните практическое задание.

4. Ответьте на контрольные вопросы.

***Практическое задание***

1. Используя справочную систему Linux или ресурсы Internet изучите не описанные опции команд find, xargx и grep.

2. Продемонстрируйте преподавателю примеры использования изученных опций.

3. Продемонстрируйте преподавателю вызовы команд с описанными опциями.

4. Выполните дополнительное задание преподавателя.

***Контрольные вопросы***

1. Для чего предназначена команда find?

2. Опишите основные опции find.

3. Как выполнить действия с файлами, найденными find?

4. Для чего предназначена команда xargs?

5. Для чего предназначена команда grep?

6. Опишите основные опции grep.

7. Как использовать grep совместно с другими утилитами? Приведите примеры.

8. Как использовать grep с регулярными выражениями?

***Библиографический список***

1. Пик Д., О’Райли Т., Лукидис М. UNIX: инструментальные средства. – К.: Издательская группа BHV, 2002.

2. Тейнсли Д. Linux и UNIX: программирование в shell. Руководство разработчика. – К.: Издательская группа BHV,2001.

3. Карлинг М., Деглер Д., Деннис Д. Системное администрирование Linux. – М.: Издательский дом «Вильямс», 2000.