

Материалы к лекции № 2

Основы работы в Linux

Фёдоров Станислав, ст. преп. кафедры ИУС

Сюжетная линия

- Основы командной строки
 - Основные компоненты операционной системы
 - Типичная структура каталогов
 - Основные команды
 - справочные команды
 - команды для работы с директориями
 - команды для работы с файлами
 - команды для управления правами доступа
 - Наглядный интерактивный пример

Сюжетная линия

- Регулярные выражения
 - Основные шаблоны и символы
 - Синтаксис `grep`
 - Наглядный интерактивный пример.
Распознавание IP-адресов

Основные компоненты операционной системы

- Ядро
- Системный загрузчик
- Утилиты поддержки (работа с файловыми системами, сетью, процессами, жестким диском, и т.п.)
- Рабочие средства пользователя (командный процессор, редакторы, базовые утилиты)
- Главная системная библиотека (glibc)

Типичная структура каталогов

- **/** Корневой каталог
- **/bin** Базовый набор утилит общего пользования
- **/boot** Каталог, содержащий ядро системы(хотя часто оно размещается в корневом разделе) и системный загрузчик
- **/dev** Каталог с виртуальными файлами, соответствующими устройствам компьютера
- **/etc** Каталог общесистемных конфигурационных файлов
- **/home** Каталог для домашних каталогов реальных и виртуальных пользователей. Содержимое его не регламентировано. Стандартное представление: `/home/{username1,...,username#}`. В системах с большим количеством пользователей их домашние каталоги могут быть объединены в группы
- **/lib** Системные библиотеки
- **/media** Каталог предназначенный для автоматического монтирования сменных устройств

Типичная структура каталогов

- **/mnt** Каталог, предназначенный для монтирования временно используемых файловых систем
- **/opt** Каталог, предназначенный для размещения коммерческого ПО и крупных программных пакетов
- **/proc** Каталог с виртуальными файлами, соответствующими запущенным процессам
- **/root** Домашний каталог пользователя root
- **/sbin** Набор утилит администрирования системы
- **/sys** Системные файлы
- **/tmp** Аналог каталога **/var**, хранящий временные файлы текущего сеанса работы
- **/usr** Каталог для пользовательских приложений и данных. Ветвь **/usr/local** предназначена для самостоятельно собираемых из исходных текстов (не входящих в дистрибутив) программ
- **/var** Хранилище изменяемых файлов, генерируемых в ходе жизнедеятельности программ – программных кэшей, log-файлов, почтовых ящиков, описаний запущенных процессов, дампов памяти и т.д.

Основные команды:

справочные команды

- **man** Получение справочной информации по команде или утилите, имя которой указано в командной строке. Например, команда “man man” даст подробное руководство по использованию справочной системы man
 - Ex: \$ man command_name
- **info** Аналог man
 - Ex: \$ info commandname

Основные команды: команды для работы с директориями

- **pwd** Вывод пути к текущему каталогу
 - Ex: \$ pwd
- **ls** Вывод списка файлов и каталогов текущего или указанного каталога
 - Ex: \$ ls
- **cd** Смена текущего каталога
 - Ex: \$ cd dirname
- **mkdir** Создание каталога
 - Ex: \$ mkdir dirname
- **rmdir** Удаление каталога
 - Ex: \$ rmdir dirname

Основные команды: команды для работы с файлами

- **cat** Вывод содержимого файла в стандартный вывод. Если имя не указано, тогда производится копирование стандартного ввода в стандартный вывод (в обычном случае — с клавиатуры на экран)
 - Ex: \$ cat filename
- **cp** Копирование файла
 - Ex: \$ cp /path_to_file/filename
- **«.»** - текущий каталог
 - Ex: \$ cp filename1 filename2
- **mv** Переименование файла или перемещение в другой каталог
 - Ex: \$ mv filename1 filename2
- **rm** Удаление файла
 - Ex: \$ rm filename
- **find** Поиск файла в дереве каталогов
 - Ex: \$ find ./ -name filename
 - Ищет файл с название «filename» текущем каталоге
- **whereis** Поиск исполняемого файла программы и его man-страницы
 - Ex: \$ whereis commandname

Основные команды: команды для работы с файлами

- **which** Поиск пути до исполняемого файла
 - Ex: \$ which binname
- **locate** Поиск файла содержащего в названии запрашиваемую фразу
 - Ex: \$ locate phrase
- **file** Определение типа файла
 - Ex: \$ file /bin/l
- **grep** Поиск в файлах строк, совпадающих с заданным шаблоном и их вывод
 - Ex: \$ grep «hello» textfile
- **sort** Сортировка строк текстовых файлов
 - Ex: \$ sort filename
- **diff** Поиск различий между двумя файлами
 - Ex: \$ diff filename filename
- **more** Постраничный вывод файла
 - Ex: \$ more textfile
- **less** Постраничный вывод файла с возможностью навигации в обоих направлениях
 - Ex: \$ less textfile

Основные команды: команды для управления правами доступа

- **chmod** Изменение прав доступа к файлу
 - Ex: \$ chmod <permission mode> filename
- **chown** Изменение владельца файла
 - Ex: \$ chgrp newowner filename
- **chgrp** Изменение группы-владельца файла
 - Ex: \$ chgrp newgroup filename

Основные команды: команды управления процессами

- **ps** Вывод списка работающих процессов
 - Ex: \$ ps
- **fg** Продолжение работы отложенного процесса
 - Ex: \$ fg %PID
- **bg** Продолжение работы отложенного процесса в фоне
 - Ex: \$ bg %PID
- **kill** Завершение работы указанного процесса.
 - Ex: \$ kill PID
 - Ex: \$ kill -9 PID – принудительное завершение процесса
- **&** Запуск процесса в фоновом режиме
 - Ex: \$ firefox &

Вопросик

Зачем в POSIX стандартизируется
иерархия директорий?

Наглядный интерактивный пример

- терминал и работа в командной строке
- поиск и получение информации по командам

Регулярные выражения. Распознавание IP-адресов

- `([0-9]{1,3}\.){3}[0-9]{1-3}` - режим расширенных регэкспов
- `\([0-9]\{1,3\}\.\\.\)\{3\}[0-9]\{1-3\}` - режим простых регэкспов
- демонстрация поиска в файлах IP-адресов

Основные шаблоны и символы

Выражение	Описание	Пример	Что означает пример
a	Обычные символы обозначают сами себя	<i>apple</i>	Строка «apple»
[...]	Любой символ, заключенный в []	<i>[02468]</i>	Любая из цифр 0,2,4,6,8
[^...]	Любой символ, не заключенный в []	<i>[^13579]</i>	Любой символ кроме нечетных цифр
[x-x]	Диапазон символов	<i>[A-Z]</i>	Любая большая латинская буква
.	Любой одиночный символ	<i>c.t</i>	cut, cat,c9t и т.д.
^	Начало строки	<i>^[0-9]</i>	Строки, начинающиеся с цифры
\$	Конец строки	<i>/bin/sh\$</i>	Строки кончающиеся bin/sh
*	Ноль или больше предшествующих символов	<i>[a-z]*</i>	Любая последовательность символов в нижнем регистре или ничего
?	Ноль или один предшествующий символ	<i>https?://</i>	http:// и https://
+	Один или более предшествующих символов	<i>T+</i>	T, TT, TTT, TTTT и т.д.
{n}	n повторов предыдущих символов	<i>[0-9]{3}</i>	Последовательность из 3 цифр, например 124, 111, 743
{n,}	n или больше предыдущих символов	<i>0{3,}</i>	000, 0000, 00000 и т.д.
{n,m}	От n до m предыдущих символов	<i>[A-Z]{2,3}</i>	Строки вроде AB, ABC, YY, ZZZ

Синтаксис **grep**

- `grep 'patern' textfile`
- Команда `grep` может работать в двух режимах: простые регулярные выражения и расширенные регулярные выражения. Для использования расширенных регулярных выражений необходимо указывать ключ **-E**.

Вопросик

Какую задачу поиска по шаблону
нельзя решить с помощью регулярных
выражений (пример)?

Спасибо за внимание