

Командный интерфейс GNU/Linux

Васильев Андрей

ЯрГУ, Россия

02.06.2012

Операционная система

Функции операционной системы

- Управление физическими ресурсами вычислительной системы
 - Предоставление API для доступа к ресурсам
 - Распределение ресурсов между процессами
- Обеспечение выполнения различного рода приложений
 - Системных
 - Прикладных

Взаимодействие с ОС

Графический интерфейс

- Найти ярлык или исполняемый файл
- Двойным щелчком запустить приложение

Продвинутое использование - горячие клавиши

Командный интерфейс

- Найти/запустить эмулятор терминала
- Написать строку
- Нажать enter

Запуск эмулятора терминала

Терминалы вне графической среды

- 6 эмуляторов терминала, доступных по `Ctrl+Alt+F1`, `Ctrl+Alt+F2`, ...
- требуют ввести имя пользователя и пароль

Приложения - эмуляторы терминала

Достаточно просто запустить

- `gnome-terminal`
- `konsole`
- ...

Что говорит терминал?

```
a@fruct-laptop: ~$
```

- "a" имя пользователя
- "fruct-laptop" имя компьютера
- "~" рабочая директория
- "\$" тип пользователя

Первые команды

- abc

Первые команды

- abc
- `whoami` - имя текущего пользователя

Первые команды

- `abc`
- `whoami` - имя текущего пользователя

Рабочий каталог

В любой момент времени вы находитесь в какой-либо директории

- `pwd` - отображение рабочего каталога
- `ls` - отображение содержимого каталога
- `ls <каталог>` - содержимое указанного каталога, "`ls /`"

Как работать с командным интерфейсом

- После строки вводится строка
- Строку можно свободно редактировать
 - Свободно перемещать курсор ввода
 - Попросить помощи, нажав на <tab>
- Можно посмотреть историю ввода
 - Стрелки вверх/вниз - передвижение по истории
 - Ctrl+R - поиск по истории
- Можно заглянуть в скрывшийся вывод:
Shift+Page Up, Shift+Page Down

Отображение содержимого директории

- `ls` выводит содержимое каталога
- `ls -a` отображение всех файлов

Файлы, начинающиеся с точки

- Скрытые
- `"."` - текущая директория
 - `ls .`
 - `ls /home/monster`
 - `ls`
- `".."` - родительский каталог

Перемещение в дереве каталогов

Для перемещения между каталогами используется команда "cd"

- `cd .` - переход в текущую директорию
- `cd ..` - переход в родительский каталог
- `cd /usr/share` - переход в каталог `"/usr/share"`
- `cd` без аргументов перемещает в домашний каталог `"~"`

Модифицирование дерева каталогов

- `mkdir <ИМЯ>` - создание нового каталога
- `rmdir <ИМЯ>` - удаление пустого каталога

Задача: создайте похожее дерево каталогов

```
+base
|+make
||it
|+love
||it
|+not-war
||hidden weapon storage
```

Как работает командный интерфейс



- Введённую строку разбирает командный интерпретатор и пытается её выполнить
- Существует больше число интерпретаторов
- Вопрос со строками
 - Заключить строку в кавычки, 'abc def'
 - Экранировать пробелы abc\ def
 - Не использовать пробелы abc-def

Задача: создайте несколько каталогов

Операции над файлами

- touch <имя> - создать пустой файл
- cp <имя файла> <назначение>
скопировать файл
 - либо указывается директория для копирования
 - либо указывается полное имя файла
- mv <имя> <назначение> переместить файл/директорию
- rm <имя> удалить указанный файл

Задача: создайте пустые файлы в каждом из созданных каталогов

Просмотр и редактирование файлов

- `cat <имя>` - вывести содержимое файла на консоль
 - `cat /var/log/Xorg.0.log`
- `less <имя>` - открыть программу для просмотра содержимого
 - `less /var/log/Xorg.0.log`
 - Используйте стрелки и Page Up/Page Down для навигации
 - Введите q для закрытия приложения
- `nano <имя файла>` - редактировать файл в текстовом редакторе
 - `emacs`, `vim` - классические текстовые редакторы
 - `gedit`, `kwrite` - графические редакторы

Запуск приложений

Введённая пользователем строка разбирается
BASH

- Внутренняя команда `bash`
- Полный путь к исполняемому файлу
- Неизвестная команда
- Bash пытается найти исполняемые файлы по путям, указанным в переменной `$PATH`
 - `echo $PATH` - вывести содержимое переменной
 - воспользуйтесь `ls` для просмотра списка исполняемых файлов