

Занятие 2. Bourne Shell aka POSIX sh.

Влад 'mend0za' Шахов
Linux & Embedded Team Leader

Linux & Embedded Department



Что такое Unix shell?

Что такое Unix shell? (Назойливый повтор)

- Обычная программа, запускающаяся после входа в систему
- Интерактивный командный интерпретатор
- Язык программирования
- Платформа интеграции (для утилит)
- Сотни разных реализаций (bash, ksh, zsh, tcsh, ...)
- Масса различных диалектов



Shell. Ключевые понятия - 1

Определения

- Приглашение командной строки (CMD PROMPT):



Shell. Ключевые понятия - 1

Определения

- Приглашение командной строки (CMD PROMPT):
\$, #, user@host:~\$



Shell. Ключевые понятия - 1

Определения

- Приглашение командной строки (CMD PROMPT):
\$, #, user@host:~\$
- Команда:
whoami; top; exit



Shell. Ключевые понятия - 1

Определения

- Приглашение командной строки (CMD PROMPT):

\$, #, user@host:~\$

- Команда:

whoami; top; exit

- Параметр:

man bash; who am i



Shell. Ключевые понятия - 1

Определения

- Приглашение командной строки (CMD PROMPT):

\$, #, user@host:~\$

- Команда:

whoami; top; exit

- Параметр:

man bash; who am i

- Ключ (1 символ):

ls -a; ls -al; ls -a -l /tmp/



Shell. Ключевые понятия - 1

Определения

- **Приглашение командной строки (CMD PROMPT):**
\$, #, user@host:~\$
- **Команда:**
whoami; top; exit
- **Параметр:**
man bash; who am i
- **Ключ (1 символ):**
ls -a; ls -al; ls -a -l /tmp/
- **Длинный ключ (GNU-style):**
ls --version



Shell. Ключевые понятия - 2

Картинка для закрепления

```
.fetchmailrc.sample
mend0za@ak112:/home/mend0za/tmp$ ls -l etc/
mend0za@ak112:/home/mend0za/tmp$ cd tmp
mend0za@ak112:/home/mend0za/tmp$ ls -la
..
..
mutt-ak112-1000-1479-223877113186584578
1034x1200-dsc06692.jpg mutt.html
1600x1404-dsc06703.jpg openvpn.tgz
build
disqus.js ppp.tgz
etc sankercup2012-final-protocol-signed.pdf
template.sh
.fetchmailrc.sample
mend0za@ak112:/home/mend0za/tmp$ ls -la etc
total 16
drwxr-xr-x 4 mend0za mend0za 4096 Окт 19 09:25 .
drwxr-xr-x 5 mend0za mend0za 4096 Ян8 8 16:07 ..
drwxr-xr-x 2 mend0za mend0za 4096 Июн 6 2012 openvpn
drwxr-xr-x 3 mend0za mend0za 4096 Окт 19 09:25 ppp
mend0za@ak112:/home/mend0za/tmp> su
Password:
su: Authentication failure
mend0za@ak112:/home/mend0za/tmp$ su
Password:
root@ak112:/home/mend0za/tmp# export PS1="# %
# pwd
/home/mend0za/tmp
# whoami
root
#
```

команды / ключи
/ параметры .

приглашение
командной
строки
(\$ # >)

работа с
переменными



Приёмы эффективной работы

Как в Shell работать быстро?



Приёмы эффективной работы

Как в Shell работать быстро?

- 1 автодополнение путей и команд
- 2 история команд
- 3 редактирование командной строки



Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB



¹Только у BASH и ZSH (если настроены)

Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды

¹Только у BASH и ZSH (если настроены)



Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды
Пример: `mys[TAB]_co[TAB]`

¹Только у BASH и ZSH (если настроены)



Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды

Пример: `mys[TAB]_co[TAB]`

Результат: `mysql_convert_table_format`

8 vs 26



¹Только у BASH и ZSH (если настроены)

Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды
Пример: `mys[TAB]_co[TAB]`
Результат: `mysql_convert_table_format`
8 vs 26
- Пути и имена файлов

¹Только у BASH и ZSH (если настроены)



Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды

Пример: `mys[TAB]_co[TAB]`

Результат: `mysql_convert_table_format`

8 vs 26

- Пути и имена файлов

Пример: `ls /u[TAB]lo[TAB]sh[TAB]/ca[TAB]`



¹Только у BASH и ZSH (если настроены)

Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды

Пример: `mys[TAB]_co[TAB]`

Результат: `mysql_convert_table_format`

8 vs 26

- Пути и имена файлов

Пример: `ls /u[TAB]lo[TAB]sh[TAB]/ca[TAB]`

Результат: `ls /usr/local/share/ca-certificates/`

16 vs 36

¹Только у BASH и ZSH (если настроены)



Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды

Пример: `mys[TAB]_co[TAB]`

Результат: `mysql_convert_table_format`

8 vs 26

- Пути и имена файлов

Пример: `ls /u[TAB]lo[TAB]sh[TAB]/ca[TAB]`

Результат: `ls /usr/local/share/ca-certificates/`

16 vs 36

- Параметры и ключи ¹

¹Только у BASH и ZSH (если настроены)



Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды

Пример: `mys[TAB]_co[TAB]`

Результат: `mysql_convert_table_format`

8 vs 26

- Пути и имена файлов

Пример: `ls /u[TAB]lo[TAB]sh[TAB]/ca[TAB]`

Результат: `ls /usr/local/share/ca-certificates/`

16 vs 36

- Параметры и ключи ¹

Пример: `apti[TAB]--a[TAB]sh[TAB]core[TAB][ENTER]`

¹Только у BASH и ZSH (если настроены)



Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды

Пример: `mys[TAB]_co[TAB]`

Результат: `mysql_convert_table_format`

8 vs 26

- Пути и имена файлов

Пример: `ls /u[TAB]lo[TAB]sh[TAB]/ca[TAB]`

Результат: `ls /usr/local/share/ca-certificates/`

16 vs 36

- Параметры и ключи ¹

Пример: `apti[TAB]--a[TAB]sh[TAB]core[TAB][ENTER]`

Результат: `aptitude --assume-yes show coreutils`

16 vs 37

¹Только у BASH и ZSH (если настроены)



Приёмы эффективной работы

Автодополнение путей и команд - 1

Волшебная кнопка - TAB

- Имя команды

Пример: `mys[TAB]_co[TAB]`

Результат: `mysql_convert_table_format`

8 vs 26

- Пути и имена файлов

Пример: `ls /u[TAB]lo[TAB]sh[TAB]/ca[TAB]`

Результат: `ls /usr/local/share/ca-certificates/`

16 vs 36

- Параметры и ключи ¹

Пример: `apti[TAB]--a[TAB]sh[TAB]core[TAB][ENTER]`

Результат: `aptitude --assume-yes show coreutils`

16 vs 37

¹Только у BASH и ZSH (если настроены)



Приёмы эффективной работы Автодополнение путей и команд - 2

Единственный вариант подстановки:
TAB дополняет сразу



Приёмы эффективной работы Автодополнение путей и команд - 2

Единственный вариант подстановки:

TAB дополняет сразу

Несколько вариантов подстановки?

Ещё больше волшебства - 2 кнопки TAB!

2xTAB - список вариантов подстановки



Приёмы эффективной работы

Автодополнение путей и команд - 3

Примеры

- `apt[TAB][TAB]`
- `aptitude --[TAB][TAB]2`
- `ls /[TAB][TAB]3`

²Только для BASH и ZSH

³Можно использовать вместо команды "ls"



Приёмы эффективной работы История команд

Просмотр истории

- “Up” и “Down” - вперёд-назад



Приёмы эффективной работы История команд

Просмотр истории

- “Up” и “Down” - вперёд-назад
- “Ctrl+R” - интерактивный поиск в истории



Приёмы эффективной работы История команд

Просмотр истории

- “Up” и “Down” - вперёд-назад
- “Ctrl+R” - интерактивный поиск в истории
- повторно “Ctrl+R” - искать дальше



Приёмы эффективной работы Редактирование командной строки

Emacs editing mode ⁴

- “Left” и “Right” - вперёд-назад по текущей строке

⁴Только KSH-совместимые: bash, zsh, pdksh, mksh, etc



Приёмы эффективной работы Редактирование командной строки

Emacs editing mode ⁴

- “Left” и “Right” - вперёд-назад по текущей строке
- “Ctrl+a” и “Ctrl+e” - перейти в начало и конец строки

⁴Только KSH-совместимые: bash, zsh, pdksh, mksh, etc



Приёмы эффективной работы Редактирование командной строки

Emacs editing mode⁴

- “Left” и “Right” - вперёд-назад по текущей строке
- “Ctrl+a” и “Ctrl+e” - перейти в начало и конец строки
- “Ctrl+u” - удалить от курсора до начала строки

⁴Только KSH-совместимые: bash, zsh, pdksh, mksh, etc



Приёмы эффективной работы Редактирование командной строки

Emacs editing mode⁴

- “Left” и “Right” - вперёд-назад по текущей строке
- “Ctrl+a” и “Ctrl+e” - перейти в начало и конец строки
- “Ctrl+u” - удалить от курсора до начала строки
- “Ctrl+w” - удалить слово (от курсора до разделителя, влево)

⁴Только KSH-совместимые: bash, zsh, pdksh, mksh, etc



Условное выполнение команд

Код возврата (RETURN CODE):
результат выполнения у любой команды Shell

Shell return code:

- 0 - выполненъ успешно
- не 0 - ошибка



Условное выполнение команд

Код возврата (RETURN CODE):
результат выполнения у любой команды Shell

Shell return code:

- 0 - выполненъ успешно
- не 0 - ошибка

Операции над кодом возврата:

- “&&” - логическое И
- “||” - логическое ИЛИ



Условное выполнение команд

Код возврата (RETURN CODE):
результат выполнения у любой команды Shell

Shell return code:

- 0 - выполненъ успешно
- не 0 - ошибка

Операции над кодом возврата:

- “&&” - логическое И
- “||” - логическое ИЛИ

Примеры:

- `cat /proc/1/envron || echo fail`
- `find /usr/share/doc -name “*.txt” && echo ok`



Скрипты

Shell Script, определение

Последовательность команд Shell.

Разделитель: перевод строки, “;”



Скрипты

Shell Script, определение

Последовательность команд Shell.

Разделитель: перевод строки, “;”

shebang

`#!/something` или чем мы запускаем скрипт.

По умолчанию : `#!/bin/sh`

Всегда первая строка скрипта.

Фактически: `/bin/sh scriptname`



Скрипты

Shell Script, определение

Последовательность команд Shell.

Разделитель: перевод строки, “;”

shebang

`#!/something` или чем мы запускаем скрипт.

По умолчанию : `#!/bin/sh`

Всегда первая строка скрипта.

Фактически: `/bin/sh scriptname`

Парадоксальные примеры

`#!/bin/rm`

`#!/bin/awk -f`

`#!/bin/less`



Запуск скриптов

- ❶ sh scriptname
- ❷ chmod +x script
./script
- ❸ из каталогов в переменной PATH
echo \$PATH
~/bin (если есть)
/usr/local/bin
- ❹ в текущей копии shell⁵
./script
source script⁶

⁵Остальные способы - запускают новый shell

⁶Несовместимо с POSIX. Происходит из ksh. Добавляет текущий каталог к списку путей



Потоки ввода-вывода

Особенности архитектуры⁷:

У каждой запущенной программы 3 потока I/O:

- 0 ввода
- 1 вывода
- 2 ошибок

Связаны с экраном и клавиатурой терминала.

⁷См документацию языка программирования Си



Потоки ввода-вывода

Особенности архитектуры⁷:

У каждой запущенной программы 3 потока I/O:

- 0 ввода
- 1 вывода
- 2 ошибок

Связаны с экраном и клавиатурой терминала.

Связаны с терминалом только по умолчанию

shell позволяет переопределить весь ввод и вывод программы



⁷См документацию языка программирования Си

Базовый синтаксис перенаправления

- Ввод “<”

```
sort <.bash_history
```

⁸Файл затрёт новым содержанием, если он существовал ранее



Базовый синтаксис перенаправления

- **Ввод** “<”
sort <.bash_history
- **Вывод** “>”⁸
find /usr/share/doc -name “*.txt” >txt-docs
- **Вывод** “1>”
find /usr/share/doc -name “*.txt” 1>txt-docs

⁸Файл затрёт новым содержанием, если он существовал ранее



Базовый синтаксис перенаправления

- **Ввод** “<”
sort <.bash_history
- **Вывод** “>”⁸
find /usr/share/doc -name “*.txt” >txt-docs
- **Вывод** “1>”
find /usr/share/doc -name “*.txt” 1>txt-docs
- **Ошибки** “2>”
find /tmp 2>find.errors

⁸Файл затрёт новым содержанием, если он существовал ранее



Базовый синтаксис перенаправления

- **Ввод** “<”
sort <.bash_history
- **Вывод** “>”⁸
find /usr/share/doc -name “*.txt” >txt-docs
- **Вывод** “1>”
find /usr/share/doc -name “*.txt” 1>txt-docs
- **Ошибки** “2>”
find /tmp 2>find.errors
- **Вывод (дописать в конец)** “1>>”
find /usr/share/doc -name “*.txt” >>txt-docs
- **Ошибки (дописать в конец)** “2>>”
find /tmp 2>>find.errors

⁸Файл затрёт новым содержанием, если он существовал ранее



Базовый синтаксис перенаправления

- **Ввод** “<”
sort <.bash_history
- **Вывод** “>”⁸
find /usr/share/doc -name “*.txt” >txt-docs
- **Вывод** “1>”
find /usr/share/doc -name “*.txt” 1>txt-docs
- **Ошибки** “2>”
find /tmp 2>find.errors
- **Вывод (дописать в конец)** “1>>”
find /usr/share/doc -name “*.txt” >>txt-docs
- **Ошибки (дописать в конец)** “2>>”
find /tmp 2>>find.errors

⁸Файл затрёт новым содержанием, если он существовал ранее



Расширенный синтаксис перенаправления

- **Pipe**⁹ “cmd1 | cmd2 ”

Вывод cmd1 направляется на ввод cmd2.

man bash|grep ksh

⁹Классика Unix

¹⁰In Real Life (IRL) используется только в скриптах



Расширенный синтаксис перенаправления

- **Pipe**⁹ “cmd1 | cmd2 ”

Вывод cmd1 направляется на ввод cmd2.

man bash|grep ksh

- **Склеить потоки** “N>&M”

В примере: просмотреть одновременно и вывод и ошибки

find /tmp 2>&1 | less

⁹Классика Unix

¹⁰In Real Life (IRL) используется только в скриптах



Расширенный синтаксис перенаправления

- **Pipe**⁹ “cmd1 | cmd2 ”

Вывод cmd1 направляется на ввод cmd2.

man bash|grep ksh

- **Склеить потоки** “N>&M”

В примере: просмотреть одновременно и вывод и ошибки

find /tmp 2>&1 | less

- **”Ввод здесь”**¹⁰ “<<END_MARKER”

```
sort <<EOF
```

```
oieu
```

```
ak
```

```
zf
```

```
EOF
```

⁹Классика Unix

¹⁰In Real Life (IRL) используется только в скриптах



Переменные

Переменные:

настройки окружения пользователя для процесса ¹¹

¹¹Смотри `environ(7)` о подробностях реализации



Переменные

Переменные:

настройки окружения пользователя для процесса ¹¹

Какие бывают?

- встроенные (в Shell): HOME, PWD, PATH, PS1
- пользовательские

¹¹Смотри `environ(7)` о подробностях реализации



Просмотр и изменение значений

- **set** - просмотр списка



Просмотр и изменение значений

- **set** - просмотр списка
- **\$VAR1** - вывести конкретную переменную



Просмотр и изменение значений

- **set** - просмотр списка
- **\$VAR1** - вывести конкретную переменную

```
echo $USER $HOME $PWD  
echo $PATH
```



Просмотр и изменение значений

- **set** - просмотр списка
- **\$VAR1** - вывести конкретную переменную

```
echo $USER $HOME $PWD  
echo $PATH
```

- **unset** - сброс (обнуление) значения
- **VAR1="значение"** - установить новое значение

