

Использование Unix Shell

Занятие 1

ОСНОВЫ синтаксиса

Иртегов Д.В.
Новосибирский гос. Университет
2016

Командная строка bash

prompt команда [параметры]

\$ date

\$ ls -l

Команда заканчивается

- Переводом строки
- Точкой с запятой
- Амперсандом
- Есть еще несколько вариантов

\$ firefox & ps

Команды

- Встроенные команды bash (полный список можно получить командой `help`)
- Определенные пользователем встроенные команды (функции и `alias`)
- Исполняемые файлы, расположенные в `$PATH`
- Любой исполняемый файл, если набрать полный путь к нему

Минимально необходимые команды

- `pwd` — текущий каталог
- `cd [каталог]` — сменить текущий каталог
 - `cd` — без параметров попадает в `$HOME`
- `ls` — список файлов
- `cp` — копировать файлы
- `mv` — переименовать файлы или переместить в другую ФС
- `echo` — вывести аргументы
- `cat` — вывести файлы в `stdout`
- `more less` — умный просмотр файлов с

Редактирование командной строки

- Стрелка вверх — предыдущая строка
- Стрелки влево-вправо, Backspace (^H), VT100 Delete — как в CUA редакторах
- Ctrl-A — начало строки (VT100 Home может не работать)
- Ctrl-E — конец строки

Стирание и вставка текста

- Backspace (Ctrl-h) — стереть предыдущий СИМВОЛ
- Ctrl-d, VT100 Delete — стереть текущий СИМВОЛ
- Ctrl-t — поменять местами соседние СИМВОЛЫ
- Ctrl-k — стереть до конца строки (и скопировать в буфер)
- Ctrl-w — стереть до начала предыдущего слова (и скопировать в буфер)
- Ctrl-y — вставить буфер перед курсором

Автодополнение

- Tab — дополнить текст
 - aTab файл, начинающийся с a
 - \$aTab команда, начинающаяся с a
 - \$VTab имя переменной, начинающейся с V
 - ~aTab имя пользователя, начинающееся с a
- Alt-? - все возможные дополнения (показать, но не вставлять)
- Alt-* - вставить все возможные дополнения

Еще полезные фенечки

- ~ - ваш домашний каталог
- ~user — домашний каталог пользователя user
- ~- - предыдущий каталог
- !\$ - последний аргумент предыдущей команды
- history — все запомненные предыдущие команды (по умолчанию ~1000)
- !434 — ранее введенная команда с номером 434 в истории

Экранирование символов

- Пробел — специальный символ для shell (разделяет аргументы)
- Что делать, если нужен аргумент с пробелом внутри? (имя файла или каталога с пробелом)
- \ - экранирует любой следующий символ
 - _ - пробел
 - \\ - обратный слэш
 - \& - амперсанд
- "Рабочий стол" — двойные кавычки. Лишают специального значения почти все

Стандартная ошибка молодого разработчика

- У многих команд, аргументы начинающиеся с - имеют специальное значение (опции)
- `ls -l`; `rm -rf`; `cp -r`
- Символ `\` - не является специальным символом для `shell`, поэтому `\`- работать не будет
- Задача: как переименовать или удалить файл `-h`
- У некоторых команд есть правило, что сначала должны идти опции, а только потом имена файлов

Историческая справка

- Кто знает, почему в Windows для разделения каталогов используется \, а не /, как в Unix?
- Командный процессор DOS (command.com) пытался обеспечить совместимость с DCL (DEC Command Language, происходит от Tenex, использовался в RT-11, RSX-11, VMS)
- В DCL, опции начинаются с /
- Разработчики DOS решили, что совместимость с DCL важнее совместимости с Unix.
- Встроенные команды CMD.EXE и многие

Упражнения

- Создайте командой `mkdir` каталог `-h`
- Перейдите в этот каталог (сделайте его текущим)
- Создайте командой `touch` пустой файл с пробелом в имени
- Вернитесь в родительский каталог (`cd ..`)
- Удалите файл в каталоге `-h`, но не сам каталог

Переменные

- `VAR=value` — присвоить
- `$VAR` — подставить
- `unset VAR` — удалить
- `VAR=` — присвоить пустое значение
- `$VAR` для несуществующей и пустой переменной заменяется на пустую строку
- Несуществующая переменная и переменная с пустым значением — разные вещи
- `VAR=value cmd` — запустить команду `cmd` с указанным значением переменной `VAR` (при этом переменная `VAR` не изменится)

Способ выстрелить себе в ногу

- `rm -rf $BUILD_DIRECTORY/`
- Что будет, если переменная `$BUILD_DIRECTORY` не определена?

Упражнения

- Посмотрите значения переменных среды `USER`, `HOME`, `PATH`
- Добавьте в `PATH` каталог `/usr/sfw/bin`, если его там нету
 - Убедитесь, что вам стали доступны команды `gcc` и `gmake`
- Заставьте команду `date` вывести время не в текущем часовом поясе, а в Калифорнии

Исполняемые файлы

- Исполняемый файл — это файл формата, понятного ядру системы, к которому вы имеете право исполнения (`chmod +x`)
- Форматы, понятные ядру системы:
 - ELF — бинарные файлы, создаются компиляторами, например языков C/C++ или Fortran
 - Текстовые файлы, начинающиеся с магической строки
`#!/path/cmd`
 - Ядро может поддерживать другие форматы, например .class-файлы Java

Скрипты shell

- Если текстовый файл начинается со строки `#!/bin/bash`
- И содержит последовательность команд `bash`
- И вы имеете на него право исполнения
- Вы можете исполнить такой файл как команду.
 - Если у файла нету магической строки или он не исполняемый, его все равно можно исполнить командой `bash file.sh`

Упражнение

- Создайте файл `hello.sh`
- Добавьте в него строки

```
#!/bin/bash
```

```
echo Hello world
```

- Сделайте его исполняемым командой
`chmod +x hello.sh`
- Запустите его

Грабли

- Если вы выполняете шелловский скрипт как команду или командой `bash script.sh`, он выполняется в отдельном процессе `bash`
- Все установленные этим скриптом переменные будут потеряны после завершения скрипта
- Если вы хотите настроить из скрипта переменные вашего процесса `shell`, необходимо запускать скрипт командами `source` или `.` (не путать с `./` !)

.-файлы шелла

- Исполняются при каждом* запуске шелла
 - * не все и не при каждом, см. далее
- /etc/profile – глобальный файл для всех bourne shell (sh, ksh, bash)
- /etc/bashrc – глобальный файл для bash
- ~/.profile – личный файл для всех bourne shell
- ~/.bashrc – личный файл для bash
- ~/.bash_profile – login bash
- ~/.bash_history – история всех команд bash
- ~/.bash_logout – исполняется при завершении login bash

Упражнение

- Добавьте в конец файла `.bashrc` команду, которая добавляет каталог `/usr/sfw/bin` в конец переменной `PATH`
- Упражнение для продвинутых
 - Сделайте так, чтобы команда исполнялась только если каталог `/usr/sfw/bin` существует
 - Подсказки: `help if`; `man test`