

Лекция 2  
Структура сложной  
командной строки



ЯЗЫК ПРОЦЕССОВ  
SHELL



# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки  
Место лекции в структуре курса

- Введение в ОС UNIX/Linux и Shell
- Структура сложной командной строки
- Управление файлами и процессами
- Утилиты обработки текста
- Создание сценариев Shell
- Управляющие конструкции Shell
- Использование sed и awk
- Средства сборки GNU

# ЯЗЫК ПРОЦЕССОВ SHELL

## Средства командной строки Принципы и возможности

- Простые команды — основа командной строки
- Сложные строки — простые команды, сгруппированные с помощью средств языка, обеспечивающих обмен данными и/или организацию выполнения в определенной последовательности
  - Конвейеры
  - Списки
  - Составные команды

# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки

Простая команда shell

- Простая команда — последовательность *слов*, разделенных *пробельными разделителями*, ограниченная *управляющим оператором*

```
student@server:~> cal -m 1 2009
```

- Слово — последовательность символов, воспринимаемое как целое, но не оператор
- Пробельный разделитель — пробел или таб
- Управляющий оператор — символ перевода строки или один из `||` `&&` `&` `;` `::` `|` `(` `)`

# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки

Простая команда shell

- Обычно первое слово — имя команды
  - Встроенная команда shell (cd, pwd, ...)
  - Внешняя программа (which ls — /bin/lS)
- Прочие слова — аргументы — данные для обработки и управление поведением
  - Неопциональные параметры  
`touch myprog.c`
  - Опции  
`ls -l --all -T 4 --timestyle=full-iso /tmp`

# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки

Простая команда shell

- Код возврата (статус выхода) — целое число, возвращаемое вызванной командой в shell
  - Нулевой код возврата соответствует успеху
  - Любое другое значение — неудачное завершение
    - 126 — команда не является исполняемой
    - 127 — команда не найдена
    - 128 + N — завершение команды по сигналу N
- Код возврата используется конструкциями shell (условное выполнение, списки команд) или сценариями пользователя (\$?)

# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки

Организация конвейера

- Конвейер (pipeline) — последовательность команд, в которой вывод предшествующей команды перенаправляется на ввод последующей

```
[time [-p]] [!] команда1 [| команда2 ...]
```

- Примеры

```
who | wc -l
```

```
who | cut -f 1 -d ' '
```

```
cat myfile | sort | uniq
```

# ЯЗЫК ПРОЦЕССОВ SHELL

## Средства командной строки Особенности конвейера

- Применение `time` обеспечивает сбор статистики о времени выполнения конвейера
  - Реальное время
  - Пользовательское время
  - Системное время
- Код возврата конвейера — код возврата последней команды. Символ `!` обеспечивает инвертирование кода возврата



# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки

Список команд

- Последовательность конвейеров, разделенных операторами `;` `&` `&&` `||` и ограниченных `;` `&` или переводом строки
- Последовательное (синхронное выполнение)  
`cd ~ ; ls`  
`./configure ; make ; make install`
- Параллельное (асинхронное выполнение)  
`find / -size +1G & cat myprog.c`  
`paste names phones > adressbook &`

# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки

Список команд

- Операторы условного выполнения `&&` `||` имеют приоритет перед остальными
- Выполнение команды по успеху предыдущей  
`ls -d test && echo 'test exists'`
- Выполнение команды по неудаче предыдущей  
`cmp f1 f2 || echo 'cannot compare files'`
- Допустима общая конструкция вида
  - `команда1 && команда2 || команда3`

# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки

Подстановка команды

- Подстановка обеспечивает возможность использования вывода одной команды в параметрах другой
- Подстановка в обратных кавычках  
`echo There are `who | wc -l` sessions now`
- Возможность вложения подстановок обеспечивается использованием `$(команда)`

# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки

Составные команды

- Группировка команд в круглых скобках ( )
- Условные конструкции (в другой лекции)
  - if ... then ... elif ... else ... fi
  - case ... in ... esac
  - select ... in ... ; do ... ; done
- Циклические конструкции (в другой лекции)
  - while ... ; do ... ; done
  - until ... ; do ... ; done
  - for ... in ... ; do ... ; done

# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки

find с ключом -exec

- Выполняет указанную команду для каждого найденного файла
  - `find / -name kernel.h -exec cat {} \;`
- Файл как параметр в команде `{}`
- Точка с запятой служит признаком завершения записи команды
- Подавляйте вывод ошибок!

# ЯЗЫК ПРОЦЕССОВ SHELL

Средства командной строки  
Небольшой набор команд

- cmp
- :
- comm
- cut
- diff
- id
- join
- mail
- mesg
- more
- nl
- od
- paste
- pr
- printf
- sort
- tee
- time
- uname
- uniq
- write