

# Операционные системы (6 семестр)

Лекция 1.2. Работа с командной оболочкой.

Ассистент, к.т.н. Митричев Иван Игоревич

Москва 2018

# План лекции

- Командная оболочка
- Состав команды.
- Шаблоны подстановки.
- Получение помощи.
- Командная подстановка.
- Вычисление арифметических выражений
- Встроенные и системные команды.
- Редактирование и выполнение команд.
- Переменные оболочки и окружения.
- Псевдонимы команд.
- История команд.

# Командные оболочки

**Командная оболочка (shell)** – программа, взаимодействующая с пользователем с помощью текстового интерфейса. Командная оболочка является *интерпретатором*.

**Command.com** - оболочка MS-DOS;

**cmd.exe** — Windows NT, XP

**PowerShell** — Windows XP (последние версии) и более новые версии Windows

**sh** – **Bourne shell**, классическая оболочка Unix;

**bash** – **Bourne Again SHell**, самая распространенная оболочка для ОС семейства Linux;

**dash** — POSIX-совместимая оболочка для Unix очень компактного размера;

**csh** – оболочка C, где используется синтаксис языка C;

**tcsh** – версия оболочки C с возможностью интерактивного редактирования командной строки.

**fish** — дружественная оболочка, имеющая множество подсказок для освоения;

**zsh** — **Z-shell** — расширенная оболочка с большим числом команд;

**ksh** – оболочка **Korn**, есть возможность редактирования команд;

Список установленных оболочек можно получить в файле `/etc/shells`.

Функции оболочек:

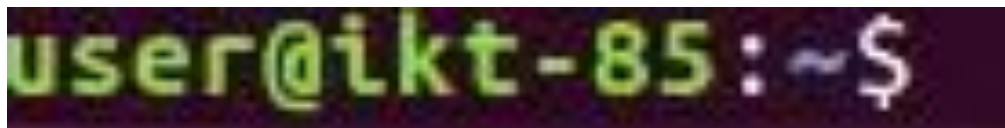
- Интерпретация ввода командной строки. Доступ к командам и результатам их выполнения.
- Поддержка переменных, специальных символов и зарезервированных слов.
- Обработка файлов, операций стандартного ввода и вывода.
- Реализация специального языка программирования оболочки.

# Приглашение ввода команд

В Linux после запуска **shell** на экран выводится приглашение к вводу команд. Начинается с:

**\$** - используется учетная запись пользователя

**#** - используется учетная запись суперпользователя (root).



```
user@ikt-85:~$
```

**user** — один из пользователей (при этом пользователь может иметь административные привилегии!)

Приглашение хранится в переменной окружения **PS1** (ее содержимое: `echo $PS1`) в виде

**u@h:w\$**

Можно определять другой формат, используя следующие спецификаторы:

**d** – текущее время;

**h** – имя хоста; **n** –

символ новой строки;

**A** – текущее время (ЧЧ:ММ);

**u** – имя

пользователя;

**w** – (в нижнем регистре) весь путь к рабочему каталогу;

**W** – (в верхнем регистре) только рабочий каталог

Вы можете сменить стандартную цветовую гамму при помощи переменной **PS1**: в тексте `[e[31m]` измените **31** на другое число.

# Ввод команд

Команда Linux – строка символов из имени команды и аргументов, разделенных пробелами. Аргументы предоставляют команде дополнительные параметры. Аргументы включают также **опции** и **имена** файлов и каталогов.

**ls -l -a file1 file2**            или    **ls -la file1 file2**    (команды идентичны)

- имя команды ls, две опции l и a, два файла file1 и file2

```
#!/bin/sh -e
#
# rc.local
exit 0
```

Команды, являющиеся частью оболочки, называются **встроенными**.

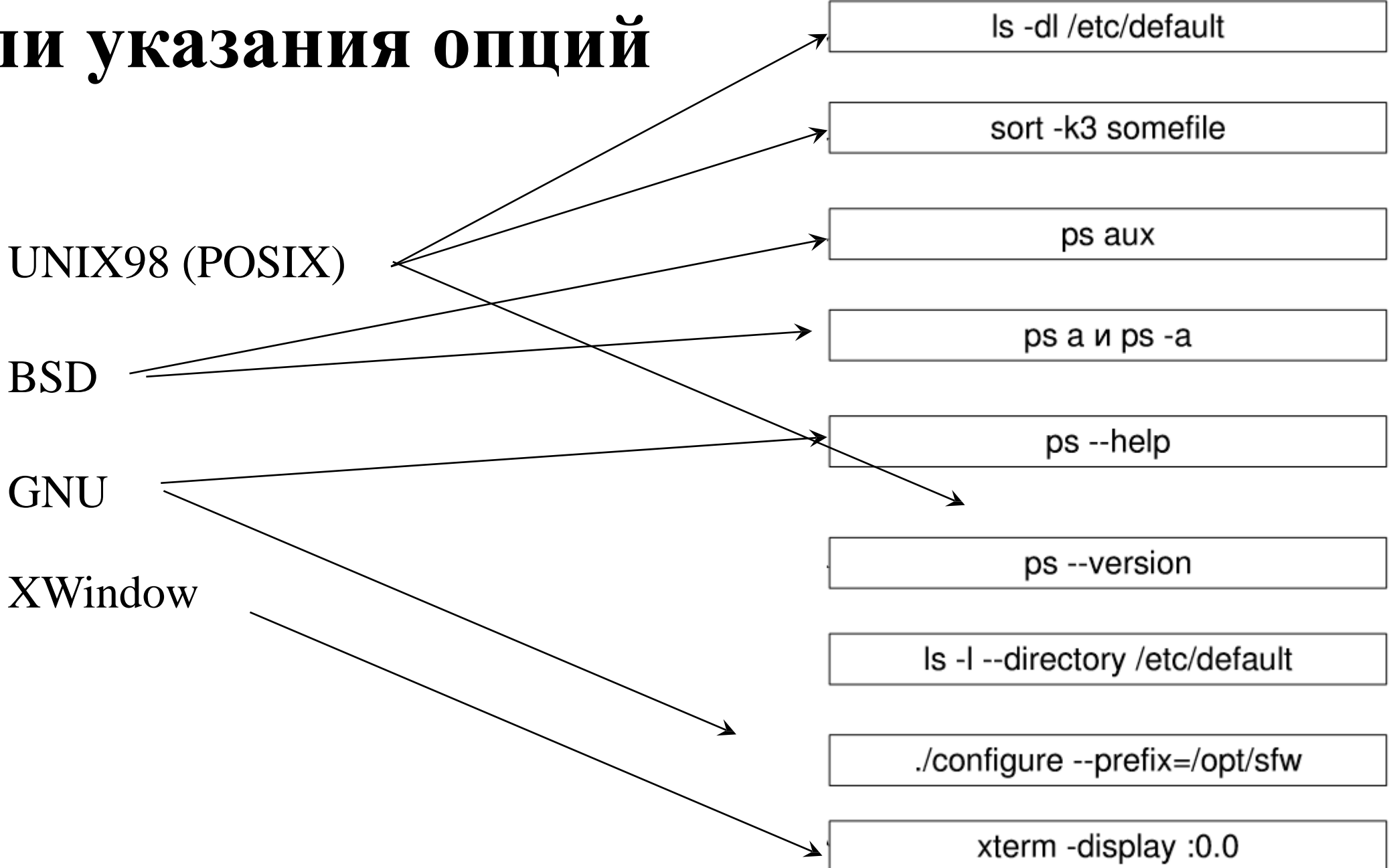
Пример: pwd, cd, exes. Они *могут выполняться различно* для различных оболочек.

Можно запускать исполняемые файлы и **скрипты (сценарии)**.

Первая строка скрипта для командной оболочки, как правило, содержит «she-bang» (#!) и название командной оболочки, в которой скрипт будет исполняться: **#!/bin/bash** – для оболочки **bash**.

**# код писал Вася** - комментарий (не обрабатывается интерпретатором команд).

# Стили указания опций



# Полезные советы

Копирование/вставка – Ctrl+Insert / Shift+Insert (вставка — также Ctrl-Shift-V)

Закрывать командную оболочку - Ctrl+D или exit

**Автодополнение вводимых команд, переменных и т.п. – Tab (крайне полезно!)** Если строка начинается с символа \$ дополняется имя переменной оболочки, ~ - дополняется имя пользователя, @ - дополняется имя хоста.

Как переключиться на виртуальный терминал? (tty1-tty7 = Ctrl+Alt+F1/ Ctrl+Alt+F7).

**Как перейти в режим root?** 1) su (заметьте: в приглашении появляется #)

2) su - копирует переменные окружения из сеанса, где вызывается команда, в сеанс суперпользователя

3) sudo -i повышение прав, пользователь должен иметь на это права (определены в /etc/sudoers)

◆ Самостоятельная работа: выясните, пароль какого пользователя требуется вводить в случаях 1, 2, 3  
(подсказка - в Debian-based дистрибутивах, напр. Ubuntu, по умолчанию пользователь root не имеет пароля)

Кто может повысить свои права?

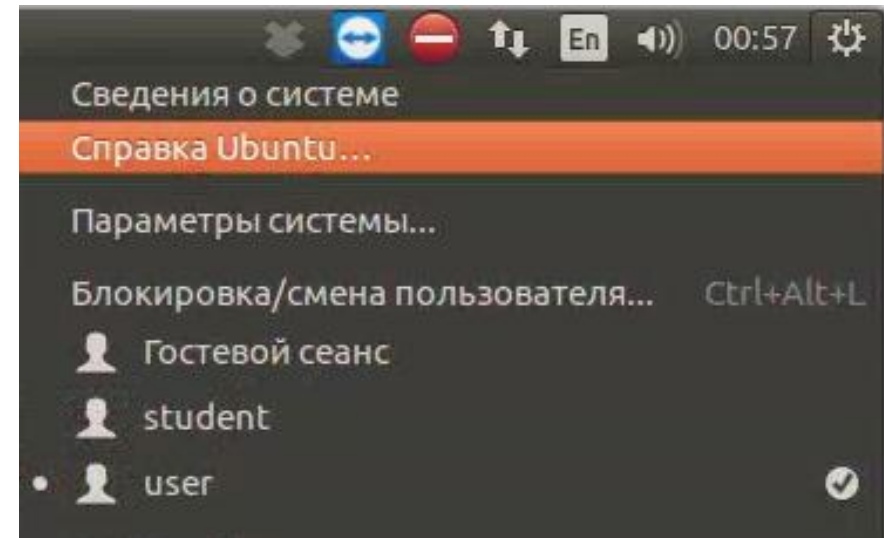
1) знающий пароль root

2) указанный в файле /etc/sudoers пользователь или член группы, указанной там

(по умолчанию — группа администраторов названа wheel)

Linux – многозадачная и многопользовательская система.

Завершение сеанса пользователя не приводит к выключению компьютера.



# Еще полезные советы

**Использование истории введенных команд** – чтобы просмотреть введенные ранее команды, используйте клавиши со стрелками **[Вверх]** и **[Вниз]**, сочетание клавиш **[Ctrl]-[R]** позволяет по введенным начальным буквам ранее использовавшейся команды вывести последнюю введенную.

**[Ctrl]-[Z]** – временно прервать исполнение команды.

**.... &** - запуск команды в фоновом режиме (при закрытии терминала команда перестанет работать!)

**nohup ... &** - «отвязать» команду от терминала — при закрытии терминала команда не перестанет работать.

Можно установить программу `screen` и выполнять команды внутри нее с аналогичным `nohup` эффектом.

**[CTRL]-[a]-[d]** — выйти из `screen`.

**screen -list** — просмотреть список запущенных экземпляров `screen`.

**screen -r номер\_процесса\_screen** — присоединиться к старой сессии `screen` заново. для создания терминала

**[Ctrl]-[C]** — остановка выполнения текущей команды (используйте при зависании).

**clear** — очистка экрана (сдвиг введенных команд вверх)



# Шаблоны подстановки и перечисление

\* – любое количество любых символов, или их отсутствие, кроме имен файлов, начинающихся на . (скрытые файлы)

echo \*

? заменяет один символ в имени файла [bcd] – один из символов перечисления

Пример: echo .[bcd]\*

```
.bash_history .bash_logout .cache .cinnamon .config .dbus .dircache .dmrc
```

[!bcd] – кроме символов b, c, d

echo .bash{rc,\_profile} – перечисление (набор вариантов); echo {,/usr}/{,s}bin

echo {,/usr{/local}}/{,s}bin – вложенный перебор.

# Получение помощи

- `help` – встроенная помощь оболочки;
- `ls --help` – отобразить справку для команды `ls`
- **man** – система помощи в любой Unix системе;
- `Info` – иерархическая система Gnu TexInfo;

Перечень разделов **man**-страниц:

- 1.– команды пользовательского уровня и приложения
- 2.– системные вызовы и коды ошибок ядра
- 3.– библиотечные функции
- 4.– информация о файлах устройств и других специальных файлах
- 5.– форматы конфигурационных файлов
- 6.– помощь по играм
- 7.– макросы, кодировки, интерфейсы
- 8.– команды системного администрирования
- 9.– внутренние интерфейсы и спецификации ядра

# Система помощи man

```
man <номер раздела> <имя страницы>
```

**man-страницы** – комплект документации в Linux.

Программа man находит страницу документации (хранятся в виде архивов), расшифровывает и открывает программу просмотра текста less:

PgUp, PgDn – перемещение по тексту;

<пробел> – следующая страница;

</> строка – поиск подстроки вниз;

<?> строка – поиск подстроки вверх;

<n> следующее вхождение искомой строки; <q> выход.

<!> - вызов командной строки для тестирования команд

Примеры:

```
man passwd
```

```
man -k clock
```

```
man -a
```

← Получить все страницы, а не только первую найденную.

```
man -k <строка> или apropos <строка>
```

← Поиск по подстроке в имени страницы.

```
man -f <строка> или whatis <строка>
```

← Поиск по полному имени страницы.

# Страница описания команды в man

NAME – информация для поиска по ключевому слову;

SYNOPSIS – шаблон вызова, опции и аргументы;

DESCRIPTION – подробное описание программы;

OPTIONS – описание опций;

FILES – файлы, связанные с командой;

AUTHOR – автор и e-mail;

SEEALSO – указатели на другие страницы man;

COPYRIGHT – права и лицензии.

```
LS(1)                               User Commands                               LS(1)
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current directory by default).
Manual page ls(1) line 1 (press h for help or q to quit)
```

# Командная подстановка

**Командная подстановка (command substitution)** – служит для передачи результатов выполнения команды другой команде.

- внешняя\_команда `внутренняя\_команда`
- внешняя\_команда \$(внутренняя\_команда)

Пример:           ls l `which rpm`  
                  ls l \$(cat /etc/shells)

Установка переменной оболочки:  
ID=`id`; echo \$ID

Целочисленная арифметика в bash:

**оператор \$(( ))**

Для арифметики с плавающей запятой  
используйте калькулятор bc.

```
echo $((1+2))
echo $((7%3))
echo $((5*6))
echo $((7/3))
```

# Встроенные и системные команды

Процедуры оболочки

Исполняемые файлы

- Встроенные команды выполняются командной оболочкой, системные являются исполняемыми файлами.
- Встроенные команды могут выполняться по-разному в разных оболочках.
- Встроенные команды выполняются быстрее, чем системные.
- Для многих встроенных команд есть системные аналоги.

- Где находятся системные команды?

- В директориях

/bin     /sbin

/usr/bin /usr/sbin

/usr/local/bin /usr/local/sbin

pwd и /bin/pwd

# Редактирование и исполнение команд

Для ввода длинной команды используется \ (перевод строки) Для ввода нескольких команд в одной строке используется ;

Команда1 && Команда2 – команда 2 выполняется только в случае удачного выполнения команды 1

Команда1 || Команда2 – команда 2 выполняется только в случае неудачного выполнения команды 1

# Переменные оболочки

Perem='double'
echo \$Perem
Perem='double word'
Perem2="triple \$Perem" Perem2=\${Perem2}WHAT
set
unset Perem2

← Задание переменной

← Вывод значения

В именах переменных допустимы латинские буквы, цифры, и знак подчеркивания. Переменные оболочки доступны только в текущей оболочке. Исключение: командная подстановка  $\$( )$  или ``` - созданные переменные копируются в дочернюю оболочку. **Переменные никогда не возвращаются** из дочерних оболочек в родительские!

← Экранирование строк

← Установка переменных

← Список переменных оболочки

← Уничтожение переменной



# Переменные окружения

- Окружение копируется при порождении дочерних процессов.
- Export служит для экспорта переменной оболочки в качестве переменной окружения:

**export VAR1**

**env** – выдает список переменных окружения

## Основные переменные окружения

HOME – путь к домашнему каталогу; LOGNAME и USER – имя пользователя; MAIL – путь к почтовому ящику;

PATH – путь поиска исполняемых файлов; PS1 – вид приглашения оболочки;

PWD – имя текущего каталога;

OLDPWD – имя предыдущего каталога;

SHELL – имя исполняемого файла оболочки; TERM – тип терминала;

HOSTNAME – имя хоста;

SHLVL – номер загруженной оболочки.

```
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
```

# Файлы пользовательских настроек

/etc/profile – в данном файле хранят переменные, устанавливаемые для всех пользователей;

~/.bash\_profile }  
~/.bash\_login } настройки профиля пользователя  
~/.profile }

~/.bashrc – исполняется при запуске оболочки вручную.

Точка «.» - файл является скрытым.

**Файлы пользовательских настроек позволяют сохранить переменные между сеансами работы.**

# Псевдонимы команд

**Псевдоним позволяет использовать короткое имя вместо длинной команды.**

`alias` — вывести список псевдонимов;

`alias ll='ls -ld'` — создать новый псевдоним;

`unalias ll` — удалить псевдоним;

`unalias a` — удалить все псевдонимы.

`~/.bashrc` — файл пользовательский настроек, в котором обычно, объявляют псевдонимы (в Ubuntu здесь уже объявлены псевдонимы для команд `ls`, `grep` с часто используемыми опциями)

# История команд

Переменные

\$HISTFILE – имя файла истории.

По умолчанию: ~/.bash\_history – файл с историей.

\$HISTFILESIZE – количество команд, хранимых в истории.

```
656 cd script_main/старые/  
657 cd ../  
658 ls  
659 cd ~/Linux2018/
```

## history

!номер команды – выполнить команду вновь;

!! последняя выполненная команда;

!<префикс> поиск недавно введенной команды, начинающейся на <префикс>;

!<подстрока> поиск недавно введенной команды по подстроке;

fc <префикс> поиск и редактирование недавней команды;

<Ctrl> + <R> поиск недавней команды по подстроке;

<вверх>, <вниз> последние команды.