



Введение в Linux

Урок 5

Программирование bash

Синтаксис. Команды. Переменные.
Управляющие конструкции. Некоторые
простые скрипты. Выполнение задач по
расписанию с помощью cron.

Вопросы по практической работе



План урока

1. Простейшие подстановки bash, переменные.
2. Простейшие скрипты (практическое занятие).

К концу урока мы изучим основные конструкции bash.

Сделаем первые шаги к написанию скриптов.



Инструменты, которые
понадобятся



Инструменты

- Установленная Ubuntu в VirtualBox или VMWare Player.
- PuTTY для удаленного доступа (по желанию).
- SFTP-клиент (чтобы залить Windows-файл в Ubuntu).





Синтаксис



Подстановки bash

```
oga@ubuntu:~$ echo *
core Desktop Documents Downloads examples.desktop Music mydoc Pictures Public Te
mplates ttt Videos vmware-tools-distrib
oga@ubuntu:~$ echo D*
Desktop Documents Downloads
oga@ubuntu:~$ echo {1..12}
1 2 3 4 5 6 7 8 9 10 11 12
oga@ubuntu:~$ echo {01..12}
01 02 03 04 05 06 07 08 09 10 11 12
oga@ubuntu:~$ echo {12..1}
12 11 10 9 8 7 6 5 4 3 2 1
oga@ubuntu:~$ echo {12..01}
12 11 10 09 08 07 06 05 04 03 02 01
oga@ubuntu:~$ echo {a..f}
a b c d e f
oga@ubuntu:~$ echo {M-T,D}
[M-T,D]
oga@ubuntu:~$ echo [M-T,D]*
Desktop Documents Downloads Music Pictures Public Templates ttt
oga@ubuntu:~$ echo [^D]*
core examples.desktop Music mydoc Pictures Public Templates ttt Videos vmware-to
ols-distrib
oga@ubuntu:~$ █
```



Подстановки bash

```
oga@ubuntu:~$ echo {a,c,e}
a c e
oga@ubuntu:~$ echo {a..e}
a b c d e
oga@ubuntu:~$ echo {a..e,z}
a..e z
oga@ubuntu:~$ echo {{a..e},z}
a b c d e z
oga@ubuntu:~$ echo *
core Desktop Documents Downloads examples.desktop Music mydoc Pictures Public Te
mplates ttt Videos vmware-tools-distrib
oga@ubuntu:~$ mkdir empty
oga@ubuntu:~$ cd empty
oga@ubuntu:~/empty$ echo *
*
oga@ubuntu:~/empty$ echo ~
/home/oga
oga@ubuntu:~/empty$ echo ~+
/home/oga/empty
oga@ubuntu:~/empty$ echo ~-
/home/oga
oga@ubuntu:~/empty$ cd ~
oga@ubuntu:~$ echo ~-
/home/oga/empty
oga@ubuntu:~$ █
```



Переменные bash

```
oga@ubuntu:~$ MyVar=Goods
oga@ubuntu:~$ echo MyVar
MyVar
oga@ubuntu:~$ echo $MyVar
Goods
oga@ubuntu:~$ echo $myVarTheBest


oga@ubuntu:~$ echo ${myVar}TheBest
TheBest
oga@ubuntu:~$ echo $myVar TheBest
TheBest
oga@ubuntu:~$ echo $MyVarTheBest

oga@ubuntu:~$ echo ${MyVar}TheBest
GoodsTheBest
oga@ubuntu:~$ echo $MyVar TheBest
Goods TheBest
oga@ubuntu:~$ MyGoods=Goods the best
The program 'the' is currently not installed. You can install it by typing:
sudo apt-get install the
oga@ubuntu:~$ echo $MyGoods

oga@ubuntu:~$ MyGoods="Goods the best"
oga@ubuntu:~$ echo $MyGoods
Goods the best
oga@ubuntu:~$ █
```



Массивы bash



```
oga@ubuntu:~$ MyVar[1]=ab
oga@ubuntu:~$ MyVar[2]=cd
oga@ubuntu:~$ MyVar[4]=fe
oga@ubuntu:~$ echo ${myVar[1]}

oga@ubuntu:~$ echo ${MyVar[1]}
ab
oga@ubuntu:~$ echo ${MyVar[2]}
cd
oga@ubuntu:~$ echo ${MyVar[3]}

oga@ubuntu:~$ echo ${MyVar[4]}
fe
oga@ubuntu:~$ echo ${#MyVar}
5
oga@ubuntu:~$ echo ${#MyVar[0]}
5
oga@ubuntu:~$ echo ${MyVar[0]}
Goods
oga@ubuntu:~$ █
```

The image shows a terminal window titled 'oga@ubuntu: ~'. It contains a series of commands and their outputs demonstrating the use of bash arrays. The array 'MyVar' is initialized with values 'ab', 'cd', and 'fe' at indices 1, 2, and 4 respectively. Subsequent commands use various syntaxes to access and display these values, including lowercase 'myVar' and uppercase 'MyVar' with and without curly braces. It also shows how to get the number of elements in the array using '#MyVar' and how to access the first element 'Goods' at index 0.



Переменные окружения

- PATH
- LANG
- TERM
- PS1
- PS2



Переменные окружения

- Присвоить значение:
 - `vpath=/tmp`
- Использовать значение:
 - `echo $vpath`
- Удалить:
 - `unset vpath`



Экспорт переменных в дочерние процессы

- `export SomeVar=SomeText`
- `SomeVar=SomeText bash`



Присвоение переменной вывода другой программы

- `result=`echo sometext``

или

- `result=$(echo sometext)`



Арифметика

- `let "c = a + b"`
 - или
- `c = `expr $a + $b``
 - или
- `c=$(($a + $b))`



Виды команд

- Алиасы.
 - Ключевые слова интерпретатора.
 - Функции.
 - Внутренние команды.
 - Внешние команды.
-
- Чтобы проверить тип команды, используйте `type`.



type

- type [опции] команда:?
 - -a
 - выведет все варианты команд, а не только тот, который будет вызываться.
 - -p
 - выведет значения команд, которые находятся во внутреннем кэше оболочки.
 - -t
 - выведет, чем является команда: псевдоним, ключевое слово, встроенная функция или файл.

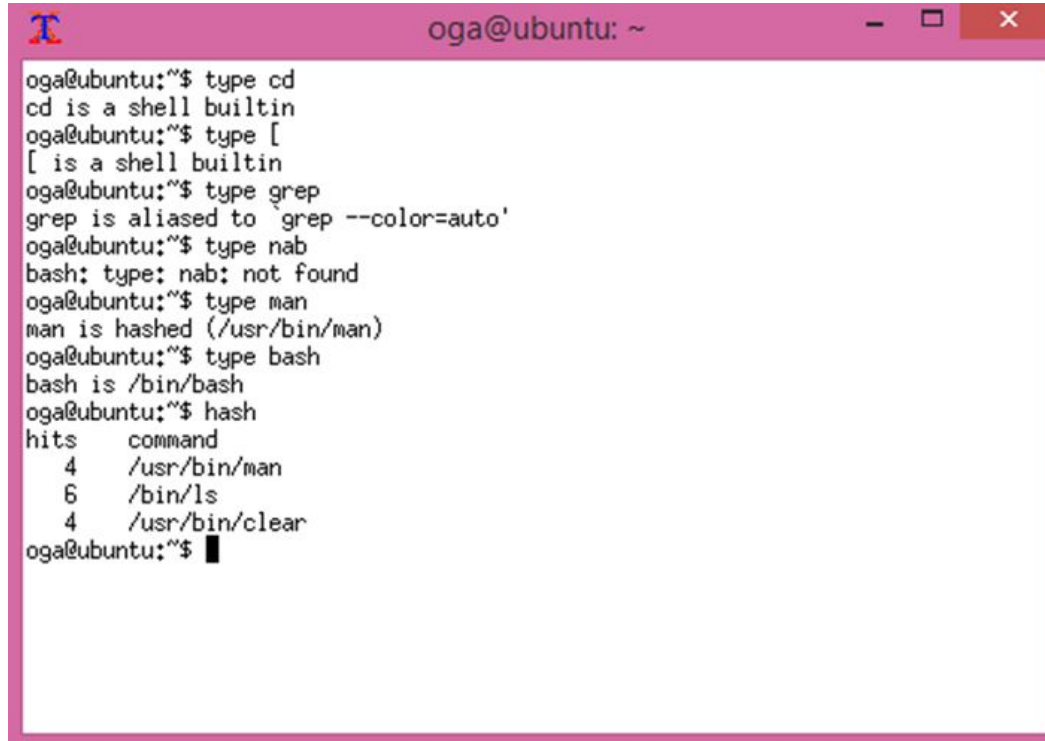


Алиас

```
oga@ubuntu: ~  
oga@ubuntu:~$ la  
.bash_history Documents mydoc Videos  
.bash_logout Downloads Pictures vmware-tools-distrib  
.bashrc examples.desktop .profile .Xauthority  
.cache .gconf Public .xsession-errors  
.config .ICEauthority .rnd .xsession-errors.old  
core .local .selected_editor  
Desktop .mozilla Templates  
.dmrc Music ttt  
oga@ubuntu:~$ ls  
core Downloads mydoc Templates vmware-tools-distrib  
Desktop examples.desktop Pictures ttt  
Documents Music Public Videos  
oga@ubuntu:~$ \ls  
core Downloads mydoc Templates vmware-tools-distrib  
Desktop examples.desktop Pictures ttt  
Documents Music Public Videos  
oga@ubuntu:~$ type la  
la is aliased to `ls -A'  
oga@ubuntu:~$ type ls  
ls is aliased to `ls --color=auto'  
oga@ubuntu:~$
```



Встроенные, внешние, хешированные команды



```
oga@ubuntu: ~  
oga@ubuntu:~$ type cd  
cd is a shell builtin  
oga@ubuntu:~$ type [  
[ is a shell builtin  
oga@ubuntu:~$ type grep  
grep is aliased to `grep --color=auto`  
oga@ubuntu:~$ type nab  
bash: type: nab: not found  
oga@ubuntu:~$ type man  
man is hashed (/usr/bin/man)  
oga@ubuntu:~$ type bash  
bash is /bin/bash  
oga@ubuntu:~$ hash  
hits      command  
   4      /usr/bin/man  
   6      /bin/ls  
   4      /usr/bin/clear  
oga@ubuntu:~$
```



Подробный вывод функций

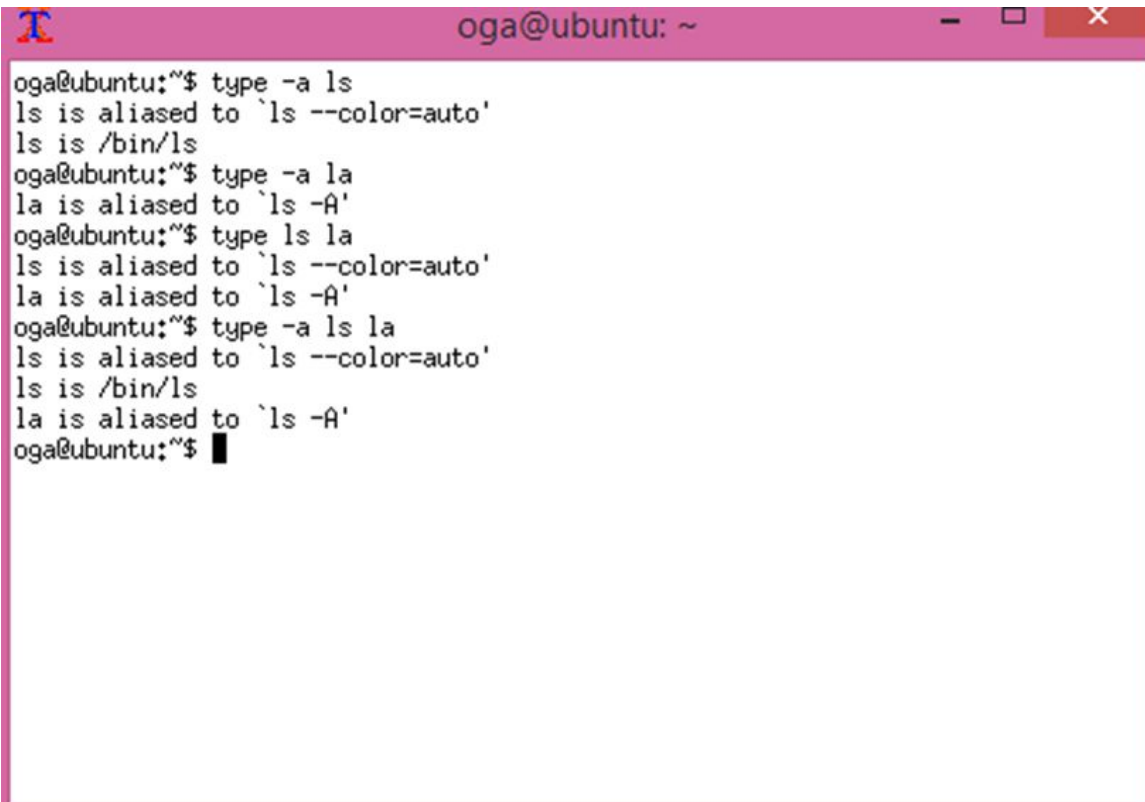
```

oga@ubuntu:~$ type ls
ls is aliased to `ls --color=auto'
oga@ubuntu:~$ type man
man is hashed (/usr/bin/man)
oga@ubuntu:~$ type -a ls
ls is aliased to `ls --color=auto'
ls is /bin/lis
oga@ubuntu:~$ type -a man
man is /usr/bin/man
oga@ubuntu:~$ quote hello; echo
'hello'
oga@ubuntu:~$ type quote
quote is a function
quote ()
{
    local quoted=${1//\'/\'\\\'\'\'\'};
    printf "%s\'" "$quoted"
}
oga@ubuntu:~$ █

```



Еще примеры



```
oga@ubuntu: ~  
oga@ubuntu:~$ type -a ls  
ls is aliased to `ls --color=auto'  
ls is /bin/ls  
oga@ubuntu:~$ type -a la  
la is aliased to `ls -A'  
oga@ubuntu:~$ type ls la  
ls is aliased to `ls --color=auto'  
la is aliased to `ls -A'  
oga@ubuntu:~$ type -a ls la  
ls is aliased to `ls --color=auto'  
ls is /bin/ls  
la is aliased to `ls -A'  
oga@ubuntu:~$
```



Функции

```
oga@ubuntu:~/empty$ function empty()  
> {  
> dirlist=`ls`  
> if [ "$dirlist" = "" ]  
> then return 0  
> else return 1  
> fi  
> }  
oga@ubuntu:~/empty$ cd ..  
oga@ubuntu:~$ mkdir F00  
oga@ubuntu:~$ empty  
oga@ubuntu:~$ empty;echo $?  
1  
oga@ubuntu:~$ >test  
oga@ubuntu:~$ empty;echo $?  
1  
oga@ubuntu:~$ ls  
core          Downloads      F00      mydoc      Templates  ttt  
Desktop       empty          Music    Pictures   test       Videos  
Documents     examples.desktop mydo2    Public     test2      vmware-tools-distrib  
oga@ubuntu:~$ cd F00  
oga@ubuntu:~/F00$ empty;echo $?  
0  
oga@ubuntu:~/F00$ █
```



Цикл for

```
for переменная in диапазон  
do  
    Действие  
done
```



Циклы



```
oga@ubuntu: ~  
oga@ubuntu:~$ for i in {1..12}  
> do  
> echo $i  
> done;  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
oga@ubuntu:~$
```

A terminal window with a pink title bar. The title bar contains a terminal icon, the text 'oga@ubuntu: ~', and standard window controls (minimize, maximize, close). The terminal content shows a 'for' loop that iterates from 1 to 12, printing each number on a new line. The prompt 'oga@ubuntu:~\$' is visible at the end of the output.



Циклы

```
oga@ubuntu: ~  
oga@ubuntu:~$ for i in *  
> do  
> echo $i  
> done  
core  
Desktop  
Documents  
Downloads  
empty  
examples.desktop  
Music  
mydo2  
mydoc  
Pictures  
Public  
Templates  
test2  
ttt  
Videos  
vmware-tools-distrib  
oga@ubuntu:~$
```



Более привычный способ

```
#!/bin/bash
for (( c=1; c<=5; c++ ))
do
    echo "Попытка номер $c"
done
```



Сравнение

```
oga@ubuntu: ~  
oga@ubuntu:~$ a=3  
oga@ubuntu:~$ echo $a  
3  
oga@ubuntu:~$ if [ $a = 3 ]  
> then  
> echo Yes  
> fi  
Yes  
oga@ubuntu:~$ if [ $a = 3 ]; then echo Yes; fi  
Yes  
oga@ubuntu:~$ if [ $a = 4 ]; then echo Yes; fi  
oga@ubuntu:~$ if [ $a = 4 ]; then echo Yes; else echo NO;fi  
NO  
oga@ubuntu:~$ a=4  
oga@ubuntu:~$ if [ $a = 4 ]; then echo Yes; else echo NO;fi  
Yes  
oga@ubuntu:~$ [ $a = 4]&&echo YES  
bash: [: missing `'  
oga@ubuntu:~$ [ $a = 4 ]&&echo YES  
YES  
oga@ubuntu:~$ [ $a = 4 ]||echo NO  
oga@ubuntu:~$ [ $a = 3 ]||echo NO  
NO  
oga@ubuntu:~$ █
```



Команда test — она же [

- == — равенство.
- != — неравенство.
- -lt — меньше.
- -gt — больше.
- -lte — меньше или равно.
- -gte — больше или равно.
- -f — файл.
- -d — директория
- и некоторые другие.



Улучшенные сравнения [[

```
if [[ 2 < 3 ]]
then
    echo Yes
fi
```

+ не разбивает строки на слова.



Взаимодействие процессов

- Конвейер:
 - `cmdfirst | cmdsecond`



Взаимодействие процессов

- Последовательный запуск:
 - `cmdfirst; cmdsecond`
- Эквивалентно:
 - `cmdfirst`
 - `cmdsecond`

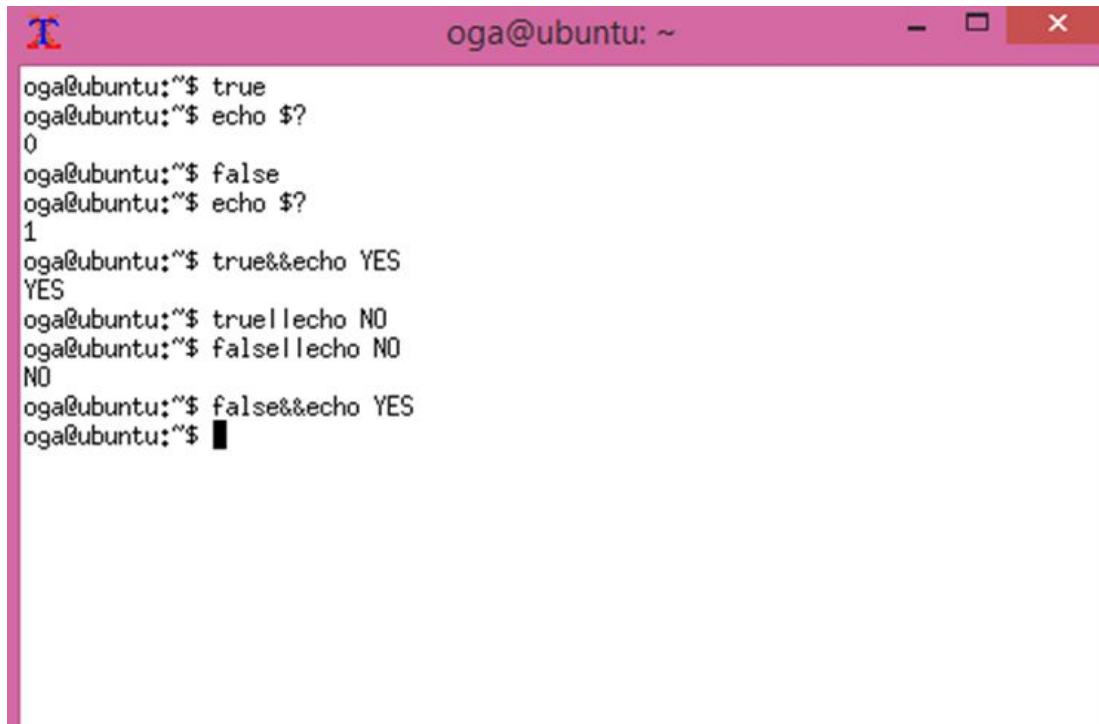


Взаимодействие процессов

- Условие:
 - `cmdfirst && cmdsecond`
 - `cmdfirst || cmdsecond`



true/false в bash



```
oga@ubuntu: ~  
oga@ubuntu:~$ true  
oga@ubuntu:~$ echo $?  
0  
oga@ubuntu:~$ false  
oga@ubuntu:~$ echo $?  
1  
oga@ubuntu:~$ true&&echo YES  
YES  
oga@ubuntu:~$ true||echo NO  
oga@ubuntu:~$ false||echo NO  
NO  
oga@ubuntu:~$ false&&echo YES  
oga@ubuntu:~$
```



Аргументы

$\$ \#$ – число аргументов

$\$0$ – имя программы

$\$1$ – первый аргумент

$\$2$ – второй аргумент

...

$\$9$ – Девятый
аргумент

$\$\{10\}$ – десятый
аргумент

$\$\{11\}$ – и так далее

$\$*$ – все аргументы



Практика



Задачи по расписанию



Формат /etc/crontab

- min hour day month weekday user command
- min (0-59) — минута,
- hour (0-23) — час,
- day (1-31) — день месяца,
- month (1-12) — день месяца,
- weekday (0-6) — день недели, 6 — воскресенье.



Сокращения в crontab

- 1-6 = 1,2,3,4,5,6,
- * - все допустимые значения,
- */3 = 0,3,6,9 ...,
- Допустимы символьные значения: sun, mon..., jun, feb..., @reboot, @yearly, @annually, @montly, @weekley, @daily, @midnight, @hourly.



Обратите внимание!

- crontab — не shell.
- Куда будут отправляться сообщения в stdout?
- Какое значение имеет PATH в crontab?



Утилита crontab

Утилита crontab позволяет управлять файлами расписаний в директории /var/spool/cron/.

Запускается с ключами:

- crontab имя_файла_расписания — добавить файл расписания.
- crontab -l — вывести содержимое текущего файла расписания.
- crontab -r — удаление текущего файла расписания.
- crontab -e — редактирование текущего файла расписания (при первом запуске будет выведен список поддерживаемых текстовых редакторов).
- sudo crontab -u username — запустить crontab от лица другого пользователя.



Разовые задачи



Примеры команды at

\$ at f mycrontest.sh 10pm tomorrow job 14 at Sun Jul 8 22:00:00 2007

\$ at f mycrontest.sh 2:00 tuesday job 15 at Tue Jul 10 02:00:00 2007

\$ at f mycrontest.sh 2:00 july 11 job 16 at Wed Jul 11 02:00:00 2007

\$ at f mycrontest.sh 2:00 next week job 17 at Sat Jul 14 02:00:00 2007



Вопросы участников

