RedHat Enterprise Linux Essential

Unit 6: The bash shell

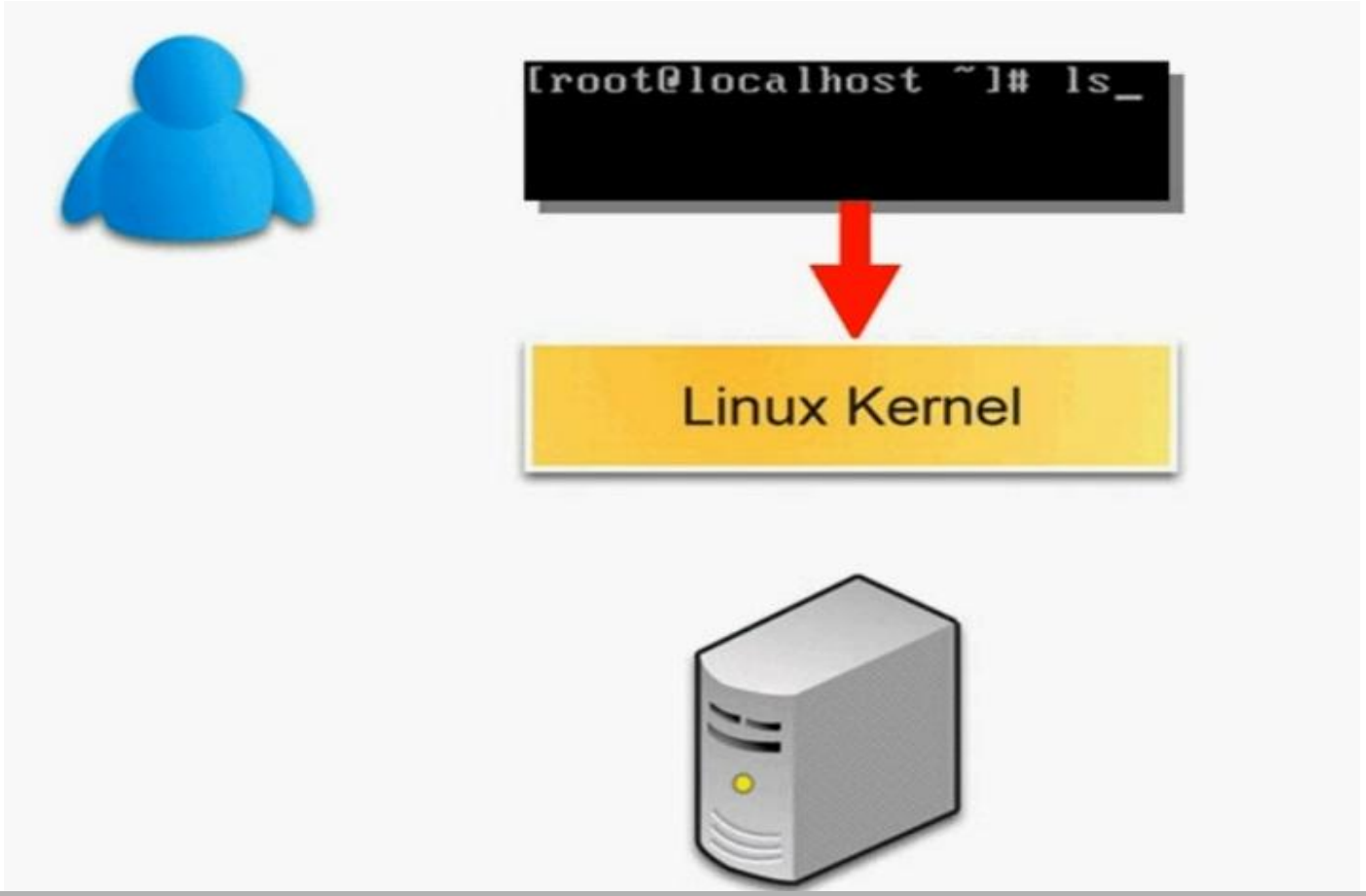
Objectives

Upon completion of this unit, you should be able to:

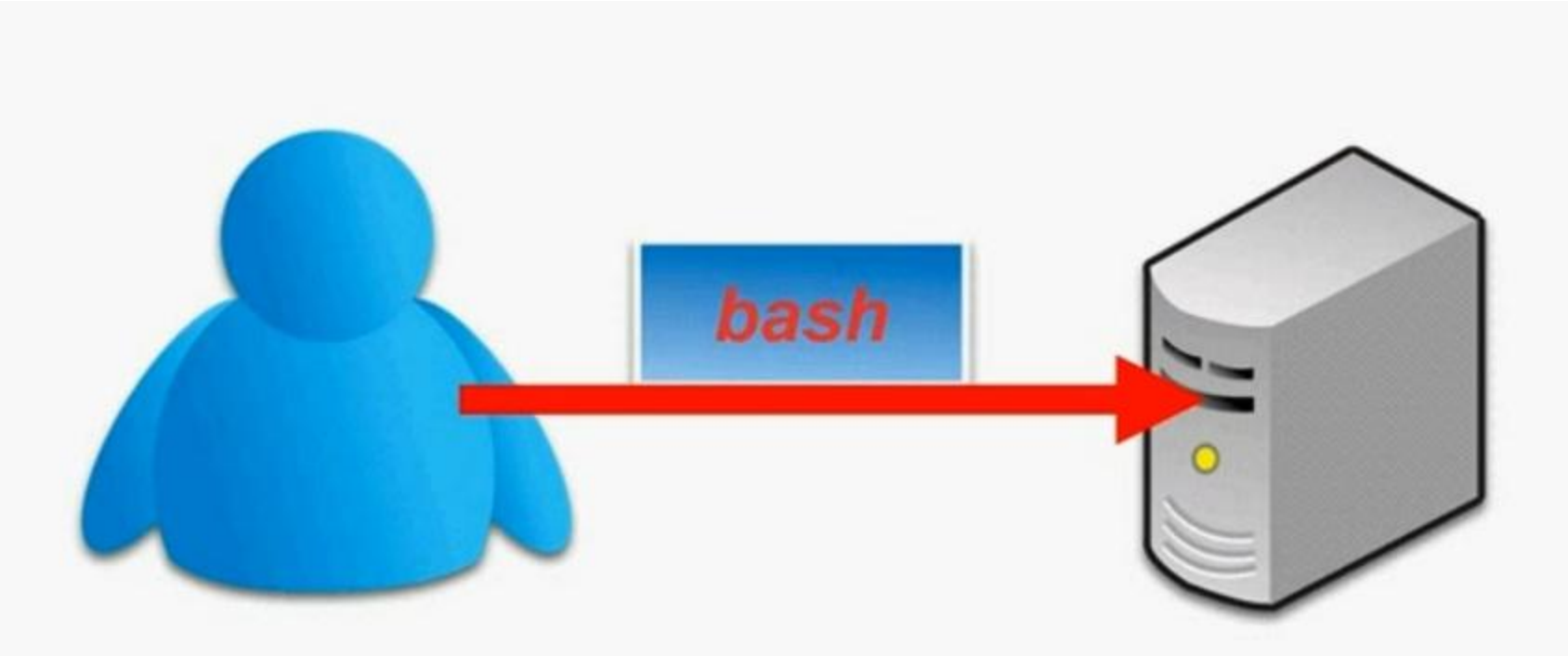
- Use command-line shortcuts
- Use command-line expansion
- Use history and editing tricks
- Use the **gnome-terminal**

Bash Introduction

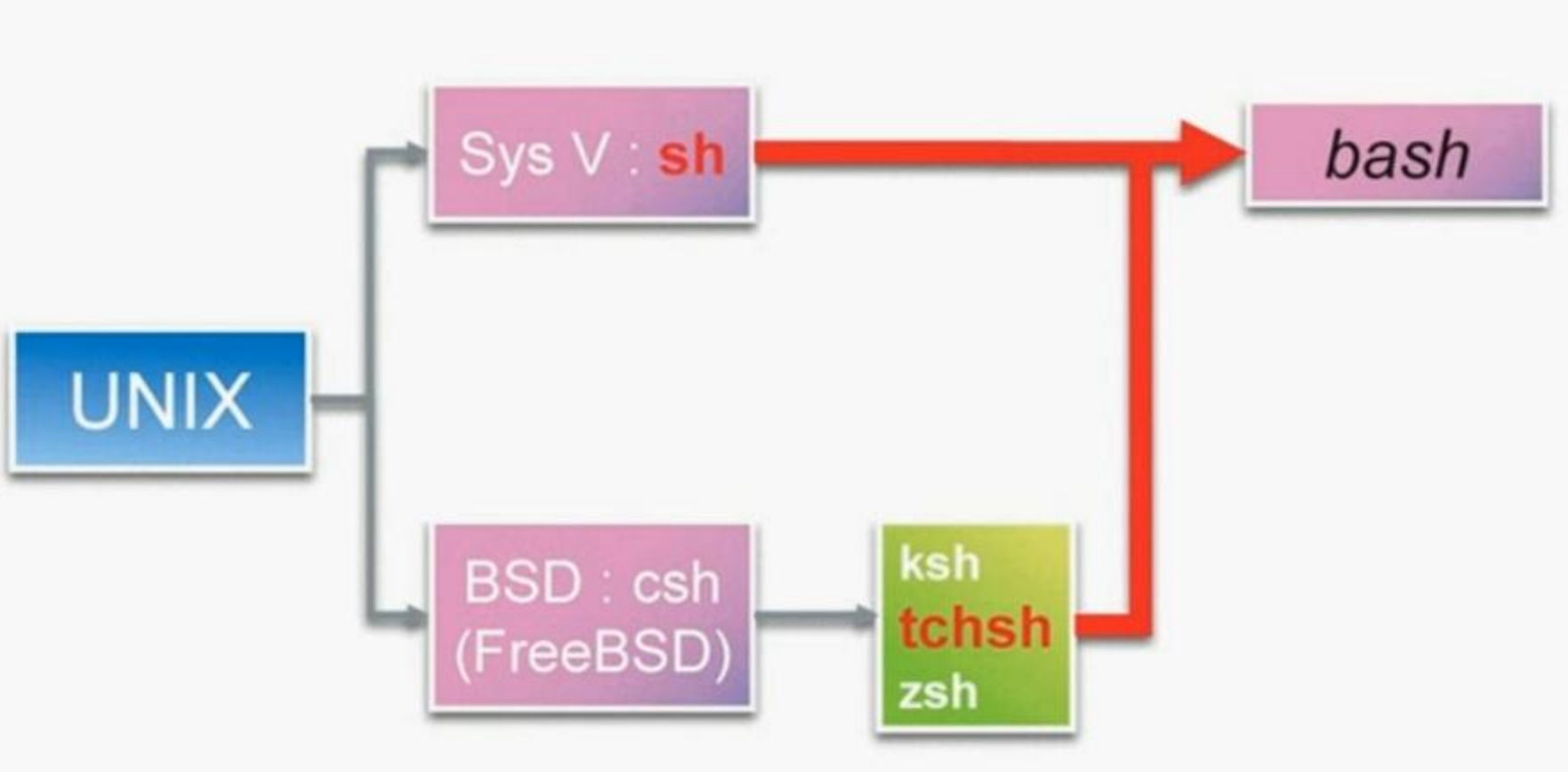
- "bourne Again Shell"
- Successor to sh, the original Unix shell



Bash Introduction



Bash Heritage



Command Line Shortcuts: File Globbing

- Globbing is wildcard expansion:

. *	matches zero or more characters
. ?	matches any single character
.[a-z]	matches a range of characters
. [^a-z]	matches all except the range

ex: touch test1.txt test2.txt test11.txt test1.mp3 test2.mp3

ls *.txt ; ls test??.txt ; ls test[1-2].txt; ls test[^1].*

Command Line Shortcuts: The Tab Key

- Type Tab to complete command lines:
  - For the command name, it will complete a command name
  - For an argument, it will complete a file name
- Examples:

$ xte<*Tab*>

$ xterm

$ ls myf<*Tab*>

$ ls myfile.txt

Command Line Shortcuts: history

- **Ex: history**
- **Try: !3**
- **Try: ^2^1**

  example: ping 192.168.1.2

  try: ^2^1

Command Line Expansion: Tilde (~)

- **Ex: pwd**
- **cd ~**
- **cd /etc/**
- **cd ~**
- **cd ~user1**

Command Line Expansion: Variable and Curly braces ({})

- echo $HOME
- cd /tmp
- touch {a,b}
- touch a{a,b}
- touch {a,b}.{1,2}

Command Line Expansion : Command and Math

Command Line Expansion : Command and Math

- hostname => localhost.localdomain (?)
- echo 'Hostname: "
- Echo Hostname: `hostname`“ ; echo Hostname:$(hostname)"
- Math: echo a, echo $HOME,

  echo $a . a=3 , echo $a

  echo $[$a + $c] ; echo $[$b + $c];

  echo $[$c / $b]; echo $[$b%$c]

  echo $[$a**$c]

Command Line Expansion: Backslash (\)

- Backslash (\) is the escape character and makes the next character literal

  ex : echo Your cost is $5.00

- Used and last character on line to "continue command on next line“

  ex : ls \

Command line expansion: Quotes

- Single quotes (') inhibit all expansion
- Double quotes (") inhibit all expansion,
  - except:
    - $
    - `
    - \
    - !

History Tricks

- use the up and down arrow keys to scroll throuht previous commans
- Type <CTRL-R> to search for a command in command history.
  - (reverse-i-search)":
- To recall last argument from previous command:
  - **<ESC>.**
  - <**ALT + .**>

Ex: ping 192.168.1.1

telnet **<ESC> .**

Command Editing Tricks

- **Ctrl-a** moves to beginning of line
- **Ctrl-e** moves to end of line
- **Ctrl-u** deletes to beginning of line
- **Ctrl-k** deletes to end of line
- ***Ctrl-arrow*** moves left or right by word

gnome - terminal

- Applications/ System Tools / Terminal
- <Ctr - shift -t> open a new tab
- <Ctrl - pgUp/ PgDn> Next/ Prev tab
- <Alt-"N"> changr to go to "N" tab
- <Ctrl - Shift-c/v> Copy/ Paste
- <Ctrl-Shift-w> Close a tab

Scripting Basics

- Shell scripts are text files that contain a series of commands or statements to be executed.
- Shell scripts are useful for:
  - Automating commonly used commands
  - Performing system administration and troubleshooting
  - Creating simple applications
  - Manipulation of text or files

Creating Shell Scripts

- **Step 1**: Use such as **vi** to create a text file containing commands
  - First line contains the magic shebang sequence: **#!**
    - **#!/bin/bash**
- Comment your scripts!
  - Comments start with a #

Creating Shell Scripts continued

- **Step 2**: Make the script executable:

```
$ chmod u+x myscript.sh
```

- To execute the new script:
  - Place the script file in a directory in the executable path -OR-
  - Specify the absolute or relative path to the script on the command line

Sample Shell Script

```
#!/bin/bash

# This script displays some information about your
   environment

echo "Greetings. The date and time are $(date)"

echo "Your working directory is: $(pwd)"
```

Thank You !