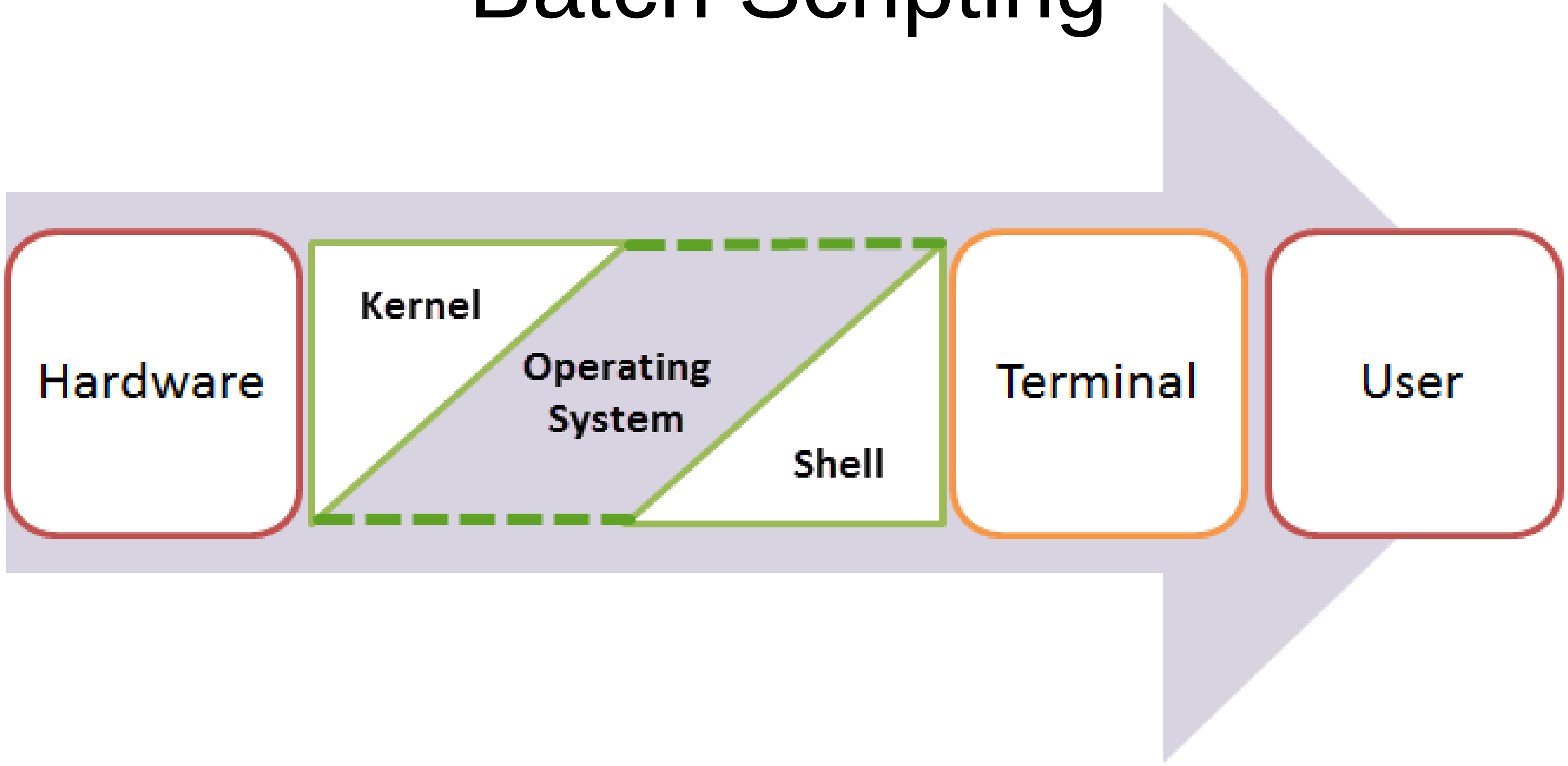
Batch Scripting



Batch Scripts

- Automate command sequences
- Perform tasks repetitively
- Allows input from user (optional)
- Uses control structures
  - if
  - for
  - while
- Advanced features
  - Functions
  - Arrays
- Can use other programming languages
  - Perl
  - Php
  - python

Common uses

- Automating house keeping activities
  - Deleting logs
  - Deleting unwanted files
- Automate copying of files
  - Backups
- Installing programs

Batch Scripting

- Complicated tasks
  - Manually difficult or time consuming
- Repetitive tasks.
  - These scripted tasks include any tasks that must be performed over and over again, such as the deletion of certain file types from specific folders on a regular basis.

Batch Scripting

- Lengthy tasks.
  - Any task that takes too long to perform manually, such as the creation of a few hundred new user accounts.
- Scheduled tasks.
  - Tasks that must be run when users and administrators are not using their computers.

Script files

- Stored in simple text files
- Each line is a valid command
- Recognised by the system
  - Command interpreter (shell or cmd.exe)
  - .bat, .cmd, on windows
  - .sh on linux
- Usually run in command prompt

Batch Files

- Unformatted text file
- Multiple commands to achieve a certain task.
- Commands are executed by command line interpreter.
- .bat or .cmd extension
- No extra s/w required
- Can perform loops and iterations

Batch Files Exercise

- Open Notepad and type the following:

```
@echo off
echo This is my first script
pause
```

- Save the file as script.bat
- Close Notepad
- Double click script.bat to run the batch file
- Now edit the script and remove @echo off
  - What is the output difference?

Batch Files Comments

- REM This is first comment
- :: This is another comment

Batch File Commands

- http://www.robvanderwoude.com/batchcommands.php
- Type *help* at the command prompt
- Not as powerful as PowerShell
-

Batch Files – A Warning

- Very powerful
- Scripts can crash the PC.
- Make sure you know what you are doing!

Unix Script Files

- Have a .sh extension (no batch files in Linux)
- Parsed (read line by line)
- First line is always #!/bin/bash
  - #! is called shebang
  - Comment lines start with hashes (#), but adding the bang (!) and then the shell path after bypasses the comment rule and forces the script to execute

Script Comments

- # indicates a comment
- A word or line beginning with # causes that word and all remaining characters on that line to be ignored.
- Ignored by the parser
- It is nothing but explanatory notes about the script.
- It makes script easier to understand.
- Comments are to help other sys admins to understand your code, logic and it helps them to modify the script you wrote.

Unix Example Script

```
#!/bin/bash
# A Simple Shell Script To Get Linux Network
Information
# A. Person – 14/Sept/2018
echo "Current date : $(date) @ $(hostname)"
echo "Network configuration"
/sbin/ifconfig
```

Enable script to run on unix

- The chmod command (change mode) is a shell command in Linux. It changes properties of files and directories.
- Each shell script must have the execute permission.
- Allowing everyone to execute the script, enter:
- *chmod +x script.sh* (allows everyone to execute)
- *chmod u+x script.sh* (allows only the owner to execute)
- To see permissions use *ls -l script.sh*
- *man chmod*
- Scripts must have both executable and read permission.
- To run the script type: *./script.sh*