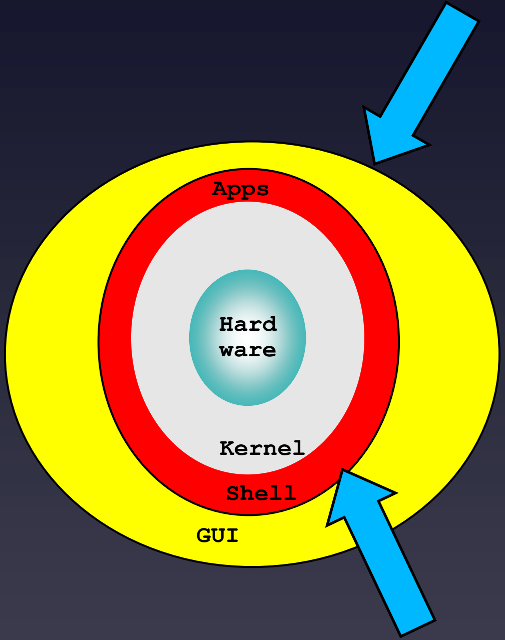
31268 Web Systems

Week 1: Operating Systems

Part 4: CLI Scripting

Week 1



Operating Systems
(User Interfaces)

Command Line Interfaces - scripting

Computer Interfaces

- **Recall:**
  - **The Command Line Interface (CLI).**

    Interact through the **keyboard** and a monitor which only prints **text**.

Batch Files and Scripting Languages

- You can automate CLI’s via a **Batch file**

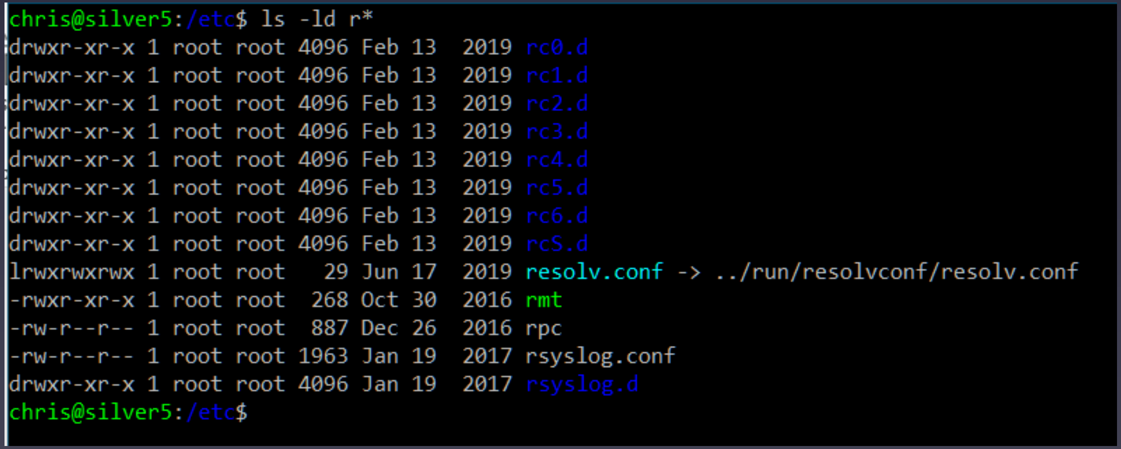
Batch Files and Scripting Languages

- You can automate CLI’s via a **Batch file**

- You can put a sequence of commands into an executable file

  → CLI treat’s the file as a command.

*ls –dl r**



```
chris@silver5:/etc$ ls -ld r*
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc0.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc1.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc2.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc3.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc4.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc5.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc6.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rcS.d
lrwxrwxrwx 1 root root   29 Jun 17  2019 resolv.conf -> ../run/resolvconf/resolv.conf
-rwxr-xr-x 1 root root  268 Oct 30  2016 rmt
-rw-r--r-- 1 root root  887 Dec 26  2016 rpc
-rw-r--r-- 1 root root 1963 Jan 19  2017 rsyslog.conf
drwxr-xr-x 1 root root 4096 Jan 19  2017 rsyslog.d
chris@silver5:/etc$
```

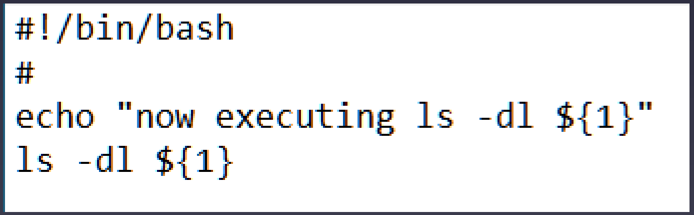
Batch Files and Scripting Languages

- You can automate CLI’s via a **Batch file**

- You can put a sequence of commands into an executable file

  → CLI treat’s the file as a command.

list.sh



```
#!/bin/bash
#
echo "now executing ls -dl ${1}"
ls -dl ${1}
```

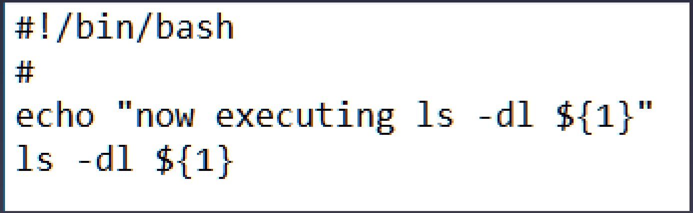
Batch Files and Scripting Languages

- You can automate CLI's via a **Batch file**

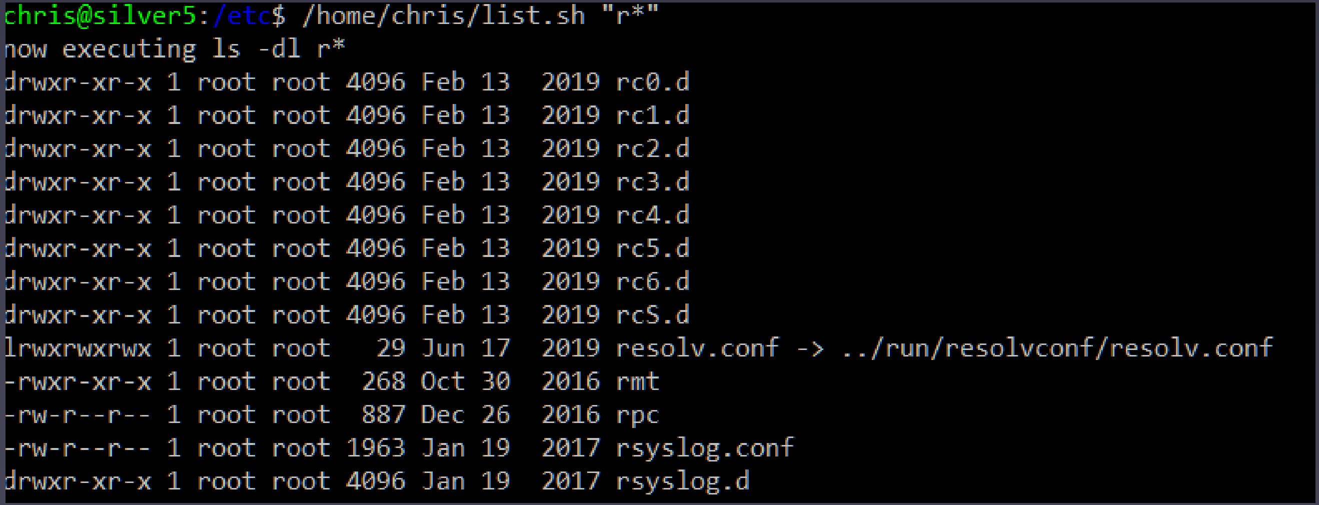
- You can put a sequence of commands into an executable file

  → CLI treat's the file as a command.

list.sh



```
#!/bin/bash
#
echo "now executing ls -dl ${1}"
ls -dl ${1}
```



```
chris@silver5:/etc$ /home/chris/list.sh "r*"
now executing ls -dl r*
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc0.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc1.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc2.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc3.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc4.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc5.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc6.d
drwxr-xr-x 1 root root 4096 Feb 13  2019 rcS.d
lrwxrwxrwx 1 root root   29 Jun 17  2019 resolv.conf -> ../run/resolvconf/resolv.conf
-rwxr-xr-x 1 root root  268 Oct 30  2016 rmt
-rw-r--r-- 1 root root  887 Dec 26  2016 rpc
-rw-r--r-- 1 root root 1963 Jan 19  2017 rsyslog.conf
drwxr-xr-x 1 root root 4096 Jan 19  2017 rsyslog.d
```

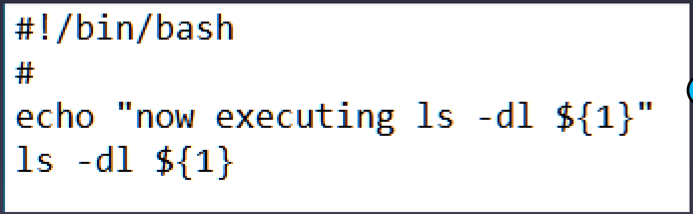
Batch Files and Scripting Languages

- You can automate CLI’s via a **Batch file**

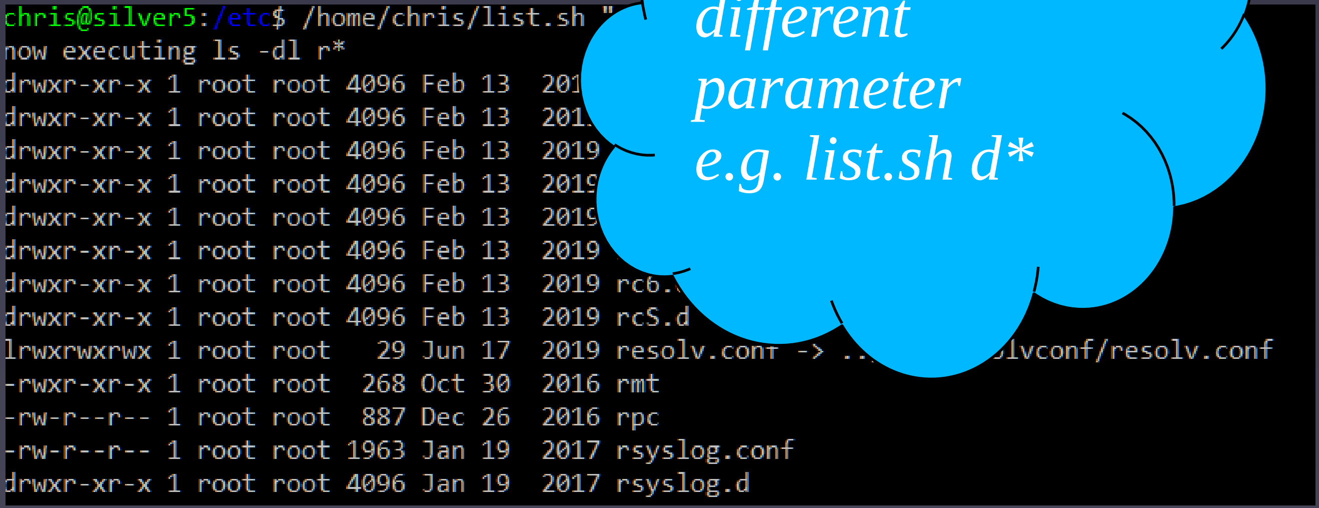
- You can put a sequence of commands into an executable file

  → CLI treat’s the file as a command.

list.sh



```
#!/bin/bash
#
echo "now executing ls -dl ${1}"
ls -dl ${1}
```



```
chris@silver5:/etc$ /home/chris/list.sh "
now executing ls -dl r*
drwxr-xr-x 1 root root 4096 Feb 13  201
drwxr-xr-x 1 root root 4096 Feb 13  201
drwxr-xr-x 1 root root 4096 Feb 13  2019
drwxr-xr-x 1 root root 4096 Feb 13  2019
drwxr-xr-x 1 root root 4096 Feb 13  2019
drwxr-xr-x 1 root root 4096 Feb 13  2019
drwxr-xr-x 1 root root 4096 Feb 13  2019 rc6.
drwxr-xr-x 1 root root 4096 Feb 13  2019 rcS.d
lrwxrwxrwx 1 root root   29 Jun 17  2019 resolv.conf -> ..          lvconf/resolv.conf
-rwxr-xr-x 1 root root  268 Oct 30  2016 rmt
-rw-r--r-- 1 root root  887 Dec 26  2016 rpc
-rw-r--r-- 1 root root 1963 Jan 19  2017 rsyslog.conf
drwxr-xr-x 1 root root 4096 Jan 19  2017 rsyslog.d
```

*Can reuse via different parameter e.g. list.sh d**

Batch Files and Scripting Languages

- You can automate CLI's via a **Batch file**

- You can put a sequence of commands into an executable file

  → CLI treat's the file as a command.

- Most CLI's include **programming** features

  → logic, calculations, variables, user input…

- Some GUI's also have batch facilities.

Batch Files and Scripting Languages

- You can automate CLI’s via a **Batch file**

- You can put a sequence of commands into an executable file

  → CLI treat’s the file as a command.

- Most CLI’s include **programming** features

  → logic, calculations, variables, user input…

- Some GUI’s also have batch facilities.

This type of programming language is called a “**scripting language**”

Example Scripting Languages

- **sh, Bash, K Shell, C shell, Z shell …**
- **Windows DOS/CMD Batch Language**
- **WMI (Windows Scripting Language), VBScript**
- **JCL (Job Control Language)** (used in Mainframes)
- **Applescript**
- **Python**
- **……**

Characteristics

- Variables are usually untyped
  (called “*loosely bound*”)
  → the same variable can be used as a number or a string.

Characteristics

- Variables are usually untyped
  (called “loosely bound”)
  → the same variable can be used as a number or a string.

- Language syntax is often inconsistent.

- Often designed and created by one person to get a particular job done.

Characteristics

- Variables are usually untyped
  (called “loosely bound”)
  → the same variable can be used as a number or a string.

- Language syntax is often inconsistent.

- Often designed and created by one person to get a particular job done.

- Usually run through an interpreter, not a compiler.

Evolution of Scripting Languages

- Scripting Languages tend to gain extra features as they evolve.

- **Perl** is a good example of this.
  - Started as a scripting language
  - now almost a generic programming Language.

- **Windows Shell** replaced by powershell
  - "Real" programming features

- **Bash** (Linux default CLI) also evolved.
  - Now includes arrays, data types etc

Next module

- We will continue with operating systems
  - History of Unix (& Linux)
  - File Systems & storage