The Shell

Computer Science and Security: The Missing Course #1

What is the shell?

- **Graphical interfaces** are common: GUIs, voice commands, AR/VR.
- But they have limitations—can't interact with non-existent buttons or undefined voice commands.
- **The Shell** is a textual interface that offers deeper computer control.

The Shell Across Platforms

- Available on nearly all platforms.
- Many options available, but all share core functionalities:
  - Run programs
  - Provide input
  - Inspect output

Focus: Bourne Again SHell (bash)

- Widely used shell with common syntax.
- To start, open a *terminal*.
- Terminal comes pre-installed or can be easily added.

Using the shell

The Prompt

- Launch terminal to see the **prompt**.
- Example of a prompt:

```
missing:~$
```

- Indicates machine name (`missing`) and current directory (`~`).
- The `$` suggests you are a non-root user.

Executing Commands

- Type a **command** at the prompt.
- Basic command execution:

```
missing:~$ date
Fri 10 Jan 2020 11:49:31 AM EST
```

- The `date` program outputs the current date and time.

Commands with Arguments

- Execute a program with **arguments**:

```
missing:~$ echo hello
hello
```

- `echo` prints out its arguments.
- Use quotes for arguments with spaces or special characters.

The Shell's Understanding

- The shell is a programming environment with variables, conditionals, loops, and functions.
- It uses `$PATH` to find programs:

```
missing:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

The Shell's Understanding (Cout)

- Use `which` to locate a program:

```
missing:~$ which echo
/bin/echo
```

- Specify full path to bypass `$PATH`:

```
missing:~$ /bin/echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Navigating in the shell

Understanding Paths

- **Linux/macOS**: `/` is the root directory.
- **Windows**: Root directory is per partition, e.g., `C:\`.
- **Absolute** path: starts with `/`.
- **Relative** path: doesn't start with `/`.

Working with Paths

- Use `pwd` to print the current working directory.
- Change directories with `cd`.
- `.` refers to the current directory.
- `..` refers to the parent directory.

```
missing:~$ pwd
/home/missing
missing:~$ cd ..
missing:/$ pwd
/
```

Directory Navigation Example

```
missing:~$ cd /home
missing:/home$ ls
missing
missing:/home$ cd missing
missing:~$ pwd
/home/missing
missing:~$ ../../bin/echo hello
hello
```

- Prompt can show current directory (configurable).
- Programs operate in the current directory by default.

Listing Directory Contents

- `ls` command lists contents of a directory.
- Without arguments, it lists the current directory.

```
missing:~$ ls
missing:/$ ls
bin boot dev etc home ...
```

Understanding `ls` Output

- `ls -l` provides detailed information.
- Permissions: `drwxr-xr-x`
  - `d` indicates a directory.
  - `rwx` indicates read, write, execute permissions.
- Permissions are shown for owner, group, and others.

```
missing:~$ ls -l /home
drwxr-xr-x 1 missing users 4096 Jun 15 2019 missing
```

Other Useful Commands

- `mv` : Rename/move a file.
- `cp` : Copy a file.
- `mkdir` : Make a new directory.
- `man` : Show manual page for a command.

```
missing:~$ man ls
```

Connecting Programs

Input and Output Streams

- Programs read input from the input stream.
- Programs print to the output stream.
- Default input: keyboard.
- Default output: screen.

Redirection and Piping

- Redirect input: `< file`
- Redirect output: `> file`
- Append to file: `>> file`
- `|`: Pipe the output of one program to another.

```
missing:~$ echo hello > hello.txt
missing:~$ cat hello.txt
hello
missing:~$ ls -l / | tail -n1
drwxr-xr-x 1 root root 4096 Jun 20 2019 var
```

Example: Using Pipes

```
missing:~$ curl --head --silent google.com | grep --ignore-case content-length | cut --delimiter=' ' -f2
219
```

- We will explore pipes more in the lecture on data wrangling.

A versatile and powerful tool

The Root User

- The **root** user has almost unlimited access.
- Can create, read, update, and delete any file.
- Typically, you don't log in as root due to risks.

Using `sudo`

- `sudo`: Execute commands as the super user or root.
- Use it when facing permission denied errors.
- Always double-check before using `sudo`.

The `sysfs` File System

- `sysfs` exposes kernel parameters as files.
- Allows reconfiguring the kernel on the fly.
- **Note**: `sysfs` is not available on Windows or macOS.

Changing Screen Brightness

- Brightness controlled through `brightness` file in `/sys/class/backlight`.
- Writing a value to this file adjusts brightness.
- Direct `sudo` command may fail due to shell permissions.

```
$ sudo echo 3 > brightness
An error occurred while redirecting file 'brightness'
open: Permission denied
```

Correct Way to Adjust Brightness

- Use `tee` command to circumvent permission issue.

```
$ echo 3 | sudo tee brightness
3
```

- `tee` opens the file for writing as root, solving the permission problem.

Controlling System LEDs

- Control system LEDs through `/sys`.

```
$ echo 1 | sudo tee /sys/class/leds/input6::scrolllock/brightness
1
```

- Example turns on the scroll lock LED.

Next steps

- You can now navigate, find files, and use basic program functions.
- Next lecture: Automating complex tasks with the shell and command-line programs.