Higher Diploma in Computer Science

Computer Systems & Networks

SPRINGBOARD
www.springboardcourses.ie

Ireland's European Structural and Investment Funds Programmes 2014-2020
Co-funded by the Irish Government and the European Union

OIDEACHAS AGUS EOLAÍOCHTA | EDUCATION AND SCIENCE

HEA | HIGHER EDUCATION AUTHORITY
AN tÚDARÁS um ARD-OIDEACHAS

European Union
Investing in your future
European Social Fund

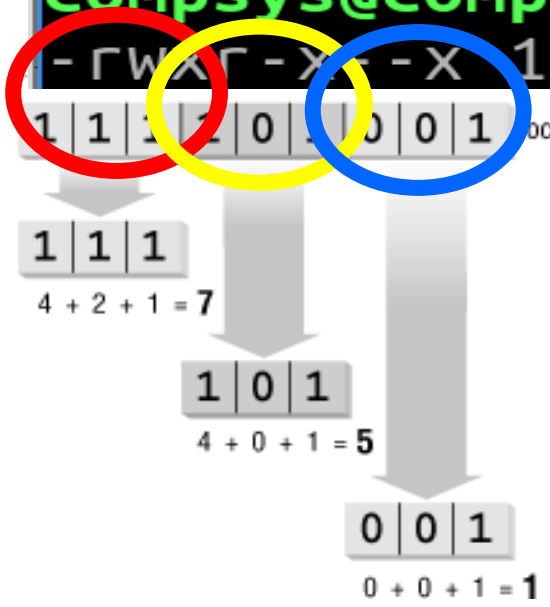
SETU
Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

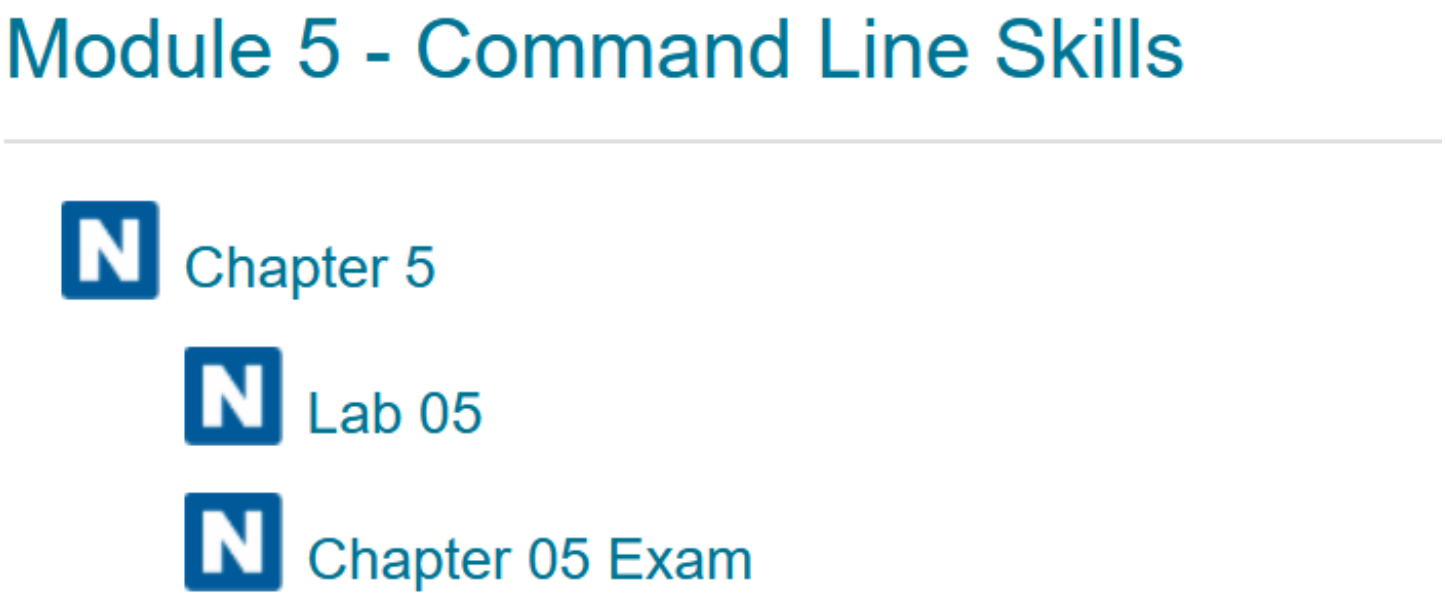
Class 1: Octal Recap

Octal Representation

0	000	-	-	-	No permissions
1	001	-	-	x	Only Execute
2	010	-	w	-	Only Write
3	011	-	w	x	Write and Execute
4	100	r	-	-	Only Read
5	101	r	-	x	Read and Execute
6	110	r	w	-	Read and Write
7	111	r	w	x	Read, Write and Execute

```
compsys@compsys-virtualbox:~$ chmod 751 stats
compsys@compsys-virtualbox:~$ ls -l st*
-rwxr-x--x 1 compsys compsys 248 Aug 29 13:54 stats
```





- Chapter 5 = class notes
  - Lab 05 = exercises to practice
  - Chapter 05 Exam = completed a part of the 5% CA

Command Line Skills

Introduction

- This class will cover the basics of the command line such as:
  - The command line interface
  - The shell
  - Commands
  - Variables
  - Command Types
  - Quoting
  - Control Statements

Command Line Interface

Command Line Interface

- The Linux community promotes the CLI due to its power, speed and ability to accomplish a vast array of tasks with a **single command line** instruction.

- The CLI provides
  - more precise control,
  - greater speed and
  - the ability to automate tasks more easily through scripting.


The Shell

The Shell

- Once a user has entered a command the terminal then accepts what the user has typed and passes to a *shell*.

```
#!/bin/bash
# This is a description of my script...

date
who | wc -l
pwd
```

- The shell is the CLI that translates commands entered by a user into actions to be performed by the OS.

The Shell

- The Bash shell also has many popular features, such as:
  - Command line history
  - Inline editing
  - Scripting
    - The ability to place commands in a file and then interpret (effectively use Bash to execute the contents of) the file, resulting in all of the commands being executed.
  - Aliases
    - The ability to create short nicknames for longer commands.
  - Variables
    - Used to store information for the Bash shell and for the user.

The Shell

- Typically the prompt contains information about the user and the system. Below is a common prompt structure:

```
sysadmin@localhost:~$
```

```
compsys@compsys-virtualbox:~$
```

- A typical prompt shown contains the following information:
  - Username (`sysadmin`)
  - System name (`localhost`)
  - Current Directory (~)

The ~ symbol is used as shorthand for the user's home directory.

Commands

Commands

- A command is a software program that when executed on the CLI, performs an action on the computer.
- To execute a command, the first step is to type the name of the command.
- If you type `ls` and hit **Enter**. The result should resemble the example below:

```
compsys@compsys-virtualbox:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

Commands

- Some commands require additional input to run correctly.
- This additional input comes in two forms: *options* and *arguments*.
  - Options are used to modify the core behavior of a command.
  - Arguments are used to provide additional information (such as a filename or a username).
- The typical format for a command is as follows:

```
command [options] [arguments]
```

Arguments

Commands

```
command [options] [arguments]
```

- An argument can be used to specify something for the command to act upon.
- If the `ls` command is given the name of a directory as an argument, it lists the contents of that directory:

```
compsys@compsys-virtualbox:~$ ls /etc/ppp
ip-down.d  ip-up.d
```

- Some commands (such as `ls`) accept **multiple** arguments:

```
compsys@compsys-virtualbox:~$ ls /etc/ppp /etc/ssh
```

Options

Options

```
command [options] [arguments]
```

- Options can be used with commands to expand or modify the way a command behaves.
- For example,

```
compsys@compsys-virtualbox:~$ ls -l
total 40
drwxr-xr-x 2 compsys compsys 4096 Sept 29  20:21 Desktop
drwxr-xr-x 1 compsys compsys 4096 Sept 29  20:25 Documents
```

```
compsys@compsys-virtualbox:~$ ls -lr
```


Variables

Variables

- A variable is a feature that allows the user or the shell to store data.
- Variables are given names and stored temporarily in memory.
- There are two types of variables used in the Bash shell,
  - *local* and
  - *environment*.

Local Variables

- Local or *shell*, variables **exist only in the current shell.**
  - When the user closes a terminal window or shell, all of the variables are lost.
  - To set the value of a variable, use the following assignment expression.

```
variable=value
```

- The following example **creates a local variable** named `variable1` and assigns it a value of `Something`:

```
compsys@compsys-virtualbox:~$ variable1='HDip are ready'
```

- **To display the value of the variable**, use a dollar sign `$` character followed by the variable name as an argument to the `echo` command:

```
compsys@compsys-virtualbox:~$ echo $variable1
HDip are ready
```

Environment Variables

- *Environment variables*, also called *global variables*, are available system-wide.
- Examples include the `PATH`, `HOME`, and `HISTSIZE` variables.

```
compsys@compsys-virtualbox:~$ echo $HISTSIZE
1000
```

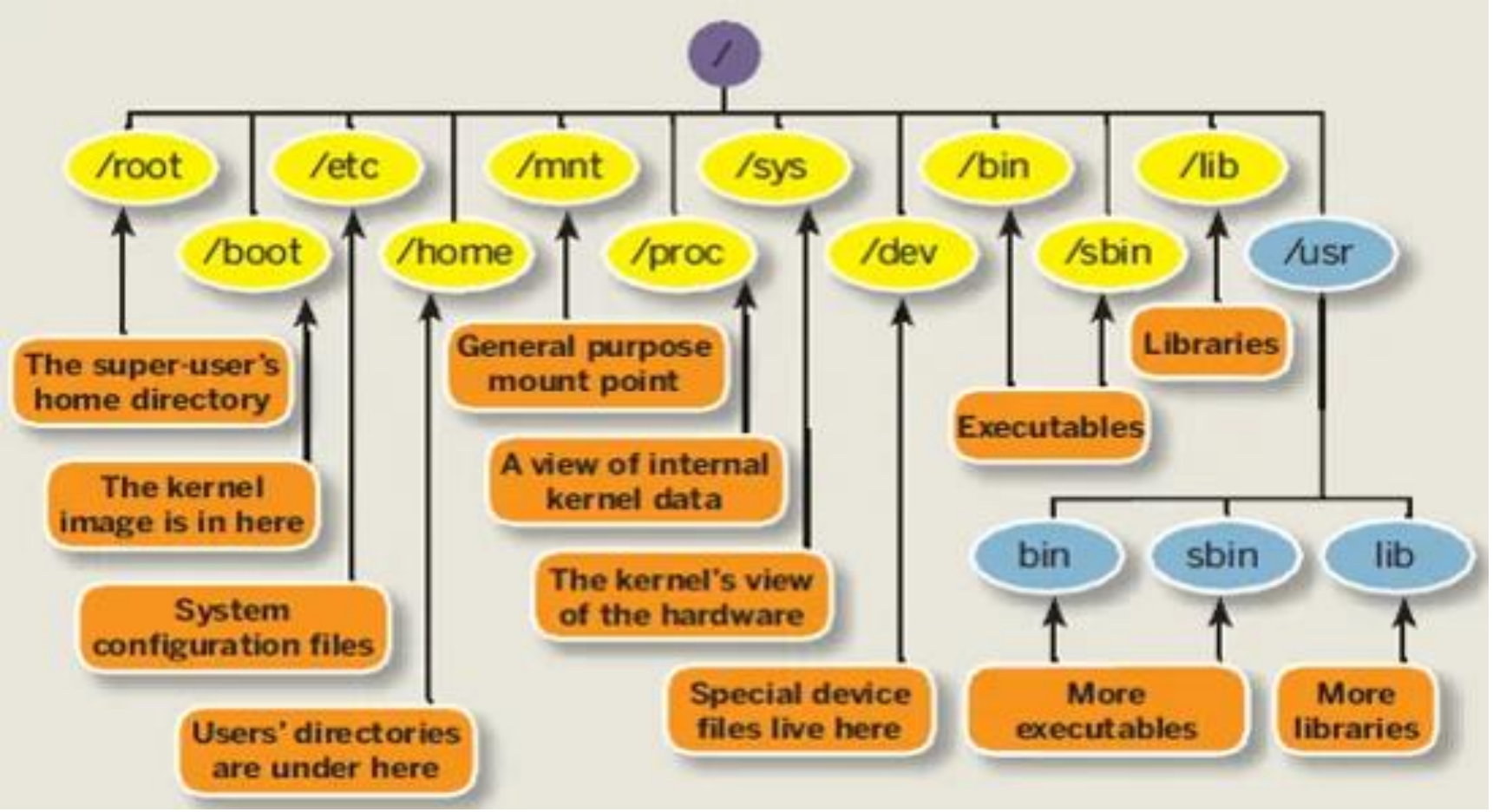
- The `env` command outputs a list of the environment variables.

Path Variable

- A very important Bash shell variables to understand is the `PATH` variable.
  - The `PATH` variable lists all the places that the system can look for programs to execute.
  - EXER: Display the path of the current shell:

```
compsys@compsys-virtualbox:~$ echo $PATH
/home/compsys/.config/nvm/versions/node/v14.3.0/bin:/usr/local/sbin:/usr/local/bin:/us
r/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
```

Filesystem Hierarchy Structure (FHS)



Command Types

Command Types

- The `type` command can be used to determine information about command type.

```
type command
```

- There are several different sources of commands within the shell of your CLI:
  - Internal commands
  - External commands
  - Aliases
  - Functions

Internal (built-in) Commands

- Also called *built-in* commands, these commands are built into the shell itself.
  - `cd` (change directory) command as it is part of the Bash shell.

- The `type` command identifies the `cd` command as an internal command:

```
compsys@compsys-virtualbox:~$ type echo
echo is a shell builtin
compsys@compsys-virtualbox:~$ type cat
cat is /usr/bin/cat
```

External Commands

- *External commands* are stored in files that are searched by the shell.

- The `which` command searches for the **location** of a command by searching the `PATH` variable.

```
compsys@compsys-virtualbox:~$ which ls
/usr/bin/ls
```

External Commands

- External commands can be executed by typing the complete path to the command.

```
compsys@compsys-virtualbox:~$ /bin/ls

Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

- For external commands, the `type` command displays the location of the command:

```
compsys@compsys-virtualbox:~$ type mkdir
mkdir is /usr/bin/mkdir
```

- To display all locations that contain the command name, use the `-a` option to the `type` command:

```
compsys@compsys-virtualbox:~$ type -a echo
```

Aliases

- An *alias* can be used to map longer commands to shorter key sequences.

- To determine what aliases are set on the current shell use the `alias` command:

```
compsys@compsys-virtualbox:~$ alias
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error
)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
```

- The `type` command can identify aliases to other commands:

```
compsys@compsys-virtualbox:~$ type ll
ll is aliased to `ls -alF'
```

Functions

- Functions can also be built using existing commands to:
  - Create new commands
  - Override commands built-in to the shell or commands stored in files