SECTION 8

DEBUGGING

SECTION CHEAT SHEET

DEBUGGING USING SHELLCHECK:

Shellcheck is a great tool that can be used either via its web interface at www.shellcheck.net or by running the shellcheck command in your terminal

Install shellcheck on Ubuntu:

```
sudo apt install shellcheck
```

Command to run shellcheck:

```
shellcheck <script>
```

Shellcheck is able to:

- Check if your script contains syntax errors
- Identify potential issues and make suggestions to improve your script

Shellcheck is not able to:

Detect issues with the environment that the script will be running in

COMMON ERRORS AND HOW TO FIX THEM

Error	What it means	How to fix it
No Such File or Directory	The file or directory that you are trying to access does not exist.	1) Check that you didn't mistype the file path 2) Check that Word splitting has not interfered with how a file path is being interpreted (did you wrap everything in double quotes?) 3) Check that the files/folders you want to work with actually exist.
File Exists	You are trying to create a file/directory that already exists	Modify the script's logic to check whether the file you want to create already exists before you run the command to create it. e.g: [-e <my file/directory>] || create it
Permission Denied	You do not have the required permissions to do what you are trying to do	Put the word sudo before the script name when you run it
Operation Not Permitted	You are trying to do something on the system that regular users are not allowed to do	Put the word sudo before the script name when you run it
Command not found	You are trying to run a program that the shell could not find on its path	1) Check for typo in the command name 2) Ensure that the program you are trying to run is on your system's PATH 3) Ensure that the program you are trying to run has execution permissions 4) Install any required packages

HOW TO FIND HELP:

From a documentation perspective, there are two types of commands:

1

INTERNAL COMMANDS.

These are commands that are built into the bash shell. You can see a full list of internal commands by running the help command.

2

EXTERNAL COMMANDS

These are commands external to the bash shell. You can find help on these using the man and info commands.

How to Check if a command is Internal or External:

```
type -a <command>
```

- If the type command says that the command is a “built-in”, then it is an internal command.
- However, if the type command gives you a path to the executable (e.g. “ls is /bin/ls”), then it is an external command.

THE HELP COMMAND

Use the help command to get information on internal commands.

Structure

- Synopsis
- Description
- Options and details of what each option does

Options

-d	Prints only the description of the command
-s	Prints only the usage information of the command

THE MAN COMMAND

The man command provides access to the systems "manual" and is a great first point of call for documentation on external commands.

Man Page Structure

Due to the creative discretion of the developers that wrote them, the structure of man pages can vary from one page to another.

However, they do tend to follow a similar format:

Name Section	The name of the command and a very short description of what it does.
Synopsis Section	How to type out the command. See here for a detailed explanation of how to read the synopsis within the man pages.
Description Section	Detailed information about how the command works and what it can do
General Info	Exit statuses, author contact details, and how to report bugs

Hint: For more information on man pages, try the command "man man"

Searching the man pages

man -k "<keyword/phrase>" allows you to search the "name section" of all man pages for a description in the man pages for commands that contain the keyword provided.

Here are the top 5 lines I get from running man -k ls

```
_llseek (2)          - reposition read/write file offset
add-shell (8)        - add shells to the list of valid login shells
afs_syscall (2)      - unimplemented system calls
ansi_send (3tcl)     - Output of ANSI control sequences to terminals
assert (3)           - abort the program if assertion is false
```

As you can see, the results aren't very relevant.

Trick: However, we can use a bit of regular expression magic to make the matches more accurate.

For example:

Here are the top 5 results when I run man -k "\bls\b". Wrapping "ls" within "\b" here helps ensure that the matching ls must be its own word before being included in the results.

```
dircolors (1)          - color setup for ls
git-ls-files (1)       - Show information about files in the index and the working tree
git-ls-remote (1)      - List references in a remote repository
git-ls-tree (1)        - List the contents of a tree object
git-mktree (1)         - Build a tree-object from ls-tree formatted text
```

As we can see, these are much more relevant results.

MORE IN DEPTH SEARCH

man -K “<keyword/phrase>” performs a more extensive search, as it will search for the keyword/phrase within all sections of all man pages, not just the name section at the top.

Do note however that this is a slower process, so man -K gives you an alternative way of interaction with man pages.

Viewing a Page

In the above lists, you will notice that each result is accompanied by a number in parentheses.

For example:

add-shell (**8**) - add shells to the list of valid login shells
afs_syscall (**2**) - unimplemented system calls
dircolors (**1**) - color setup for ls

These numbers refer to the section of the manual that these commands are found in.

Therefore, to see the add-shell command, which is in section 8, the best way to open its man page is as follows:

```
man 8 add-shell
```

This approach even works for complicated results like this one:

```
ansi_send (3tcl)      - Output of ANSI control sequences to terminals
```

You could open this result like so:

```
man 3tcl ansi_send
```

Note: To exit a man page you have to press q

MANUAL SECTIONS

Below is a list of the manual's sections, which you can use to get an idea of what type of command you are looking at:

1. Executable programs or shell commands
2. System calls (functions provided by the kernel)
3. Library calls (functions within program libraries)
4. Special files (usually found in /dev)
5. File formats and conventions eg /etc/passwd
6. Games
7. Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
8. System administration commands (usually only for root)
9. Kernel routines [Non standard]

THE INFO COMMAND

The info command also provides information on external commands, and it typically provides more detailed information than the man command does.

Info pages also contain hyperlink-style references, which makes it easy to hop from one info page to another.

Note: Not every command has an entry for the info command, so the man command is often the most sure point of call.

Check out what the info command can do by running "info" in your terminal.

Hint: Press] to see the next page. Press [to see the previous page. Press "enter" on a link to follow it.