

Trevor Mileur

PSET 2

#Problem 1

The alias command recognizes two tokens. Alias_name="command". The setup in problem 1, sets the alias name to "hello", such that when the name is called the command "echo hello world" is executed. This alias is temporary inside the shell and will cease to exist once exiting the terminal. We could make this command more permanent by adding it to the configuration using the .bashrc.

Alternatively, we could generate a "hello_world.txt" file and echo it's contents via the command line using the "cat" command.

```
cat hello_world.txt
```

#Problem 2

```
Alias zz=zz
```

```
zz
```

Generates a "command not found error". The problem is that zz is not a defined command or function. When you the alias_name, zz the shell expands to the command zz, which bash readily recognizes is not executable.

#Problem 3

```
echo "argument number one is $1"  
echo "argument number two is $2"  
echo "rest of the arguments ${@:3}"  
echo "all arguments $@"
```

```
chmod u+x ./experiment.sh  
./experiment.sh a b c d e
```

This bit of script takes in several arguments, in our case a, b, c, d, e, and then passes them to the shell script we saved to our local drive. They are then saved as local variables as an array [a, b, c, d, e]. When we execute the script the variables are called using the "\$" sign.

In the first argument \$1 corresponds to "a" stored in our array.

In the third argument \${@:3} corresponds to position 3 in the array and every position afterwards, so [c, d, e].

Finally the `$@` command alone corresponds to all arguments and will output the arguments passed to the shell script.

#Problem 4

I hope this answer is sufficient. In my own words, `#!/bin/bash` modifies the current shell such that it will begin to operate under the conditions of a bash shell.

Purpose of `#!/bin/bash`

1. **Specifies the Interpreter:** The shebang line tells the operating system which interpreter to use to run the script. In this case, `/bin/bash` specifies the Bash shell as the interpreter.
2. **Script Portability:** By explicitly defining the interpreter, the script becomes more portable and consistent across different environments. This ensures that the script will run with the correct shell, even if the user's default shell is different (e.g., `zsh`, `sh`, `dash`).

#Problem 5

```
bash ./experiment.sh a 'b c' d
```

The bash shell recognizes specific tokens within the arguments we pass as variables. In this case it recognizes what is between the two `'` symbols as being one unique token.

```
[TRVR HW2]$ bash ./experiment.sh a 'echo hello world' d
argument number one is a
argument number two is echo hello world
rest of the arguments d
all arguments a echo hello world d
[TRVR HW2]$ █
```

As shown here, the `'echo hello world'` is recognized as literal characters and is not further expanded.

Trevor Mileur

PSET 2

#Problem 6

RUN is used to configure the container before it's been generated, whereas CMD is utilized to execute a command inside the container when it's built. A key difference being the timing in which RUN vs CMD is utilized.

As my old friend, GPT would say, "Understanding these differences helps ensure that Dockerfiles are structured correctly for both building reliable images and defining expected runtime behavior."

As I would say, "RUN helps structure the container properly, whereas CMD defines the expected behavior. "

#Problem 7

1. apt
2. pip
3. install.packages

All of these are package managers of various kinds.

The first "apt" is a package manager for linux distributions. This command would allow the installation of packages across the operating system. Since I'm running a MBP-M3, I will use brew install <package_name> instead of "apt" to achieve the same outcome. To utilize brew, I added the brew to my /.bash_profile using nano. Thanks for helping!

Pip is a package manager specifically for python. This command will allow one to install, remove, or update packages in the python environment.

Install.packages is unique to R. This command will allow me to install packages in the RStudio environment, which will likely be helpful in the next parts of the course.