**Staffordshire University**

**Department of Computing**

**COMP40004: Web Development & Operating Systems**

**Week 4: Bash Scripts**

For each of the below tasks, create and run individual bash scripts.

After each task, use the Linux files app to copy the script to your shared folder. Then on your host computer (e.g. Windows/macOS), copy the script out of the shared folder to your OneDrive.

I've filled in the first box for you, but check that it works, and go through the above steps.

|  |  |
| --- | --- |
| **Task** | **Script Code, then commands to run** |
| **1. A bash script called task1.sh which;**   1. **Uses read to ask the user to enter their name.**      1. **Outputs "Hello *name*" where name is the name the user typed in.** | #!/bin/bash  echo "What's your name?" read name  echo "Hello ${name}"    chmod u+x task1.sh  ./task1.sh |
| **2. A bash script called task2.sh which;**   * 1. **Uses read to ask the user to enter their nearest city (e.g. Stoke).**      * 1. **Uses read to ask the user to enter the county that city is in (e.g.**   **Staffordshire).**     * 1. **Outputs "*ytic*" where yitc is the city the user entered, reversed.**   **(e.g. Stoke would be ekotS)**     * 1. **Appends "*city* is in *county*" to a file called location.log where *city* is the city the user entered and *county* is the county the user entered. E.g. "Stoke is in Staffordshire".**      * 1. **Run the script three times entering different cities and counties each time, and use cat to check they are all visible in location.log** | #!/bin/bash  Echo “Please enter the nearest city?”  Read city  Echo “Enter the county?”  Read county  Echo $city|rev  Echo “${city} is in ${county}” >> location.log |

|  |  |
| --- | --- |
| **3. A bash script called task3.sh which someone could use to record favourite colour survey results which;**   1. **Takes the name of a colour as a parameter. Do *not*****use read in this script.**      1. **When run, confirms with echo "Response *colour* recorded" where colour is the colour given as a parameter (e.g. "Orange" if**   **called as "./task3.sh Orange")**     1. **Appends to a file called colours.log the following text: "*colour* on *date* at *time*" where *colour* is the colour given as a parameter, date is the current date in DD/MM/YYYY format, and time is the current time in HH:MM:SS format. E.g. "Orange**   **on 31/09/2020 at 15:44:01"**     1. **Run the script three times supplying different colour parameters each time, and use cat to check they are all visible in colours.log** | #!/bin/bash  Echo “Response $1 recorded”  Datevalue=$(date +%D)  Timevalue=$(date +%T)  Echo “$1 on ${datevalue} at ${timevalue}” >> colours.log |
| **4. A bash script called task4.sh which;**   1. **Takes two parameters - a domain to download, then a file extension (like "css" or "jpg" or "png"). Do *not*****use read in this script.**      1. **Uses wget to download the URL given as the first parameter.**      1. **Uses appropriate wget arguments and a pipe to pass the output of wget to grep. Use grep to find a full stop followed by the file extension supplied in the second argument. So if "jpg" is supplied, look for ".jpg"**      1. **Test your script by searching for "png" on "https://twitter.com", then by searching for "js" on**   **"https://www.staffs.ac.uk"** | #!/bin/bash  Wget -q -o - $1 | grep “.$2”  Cat index.html |
| **5. A bash script called task5.sh which helps you to write other bash scripts by writing the shell selection header, sets permissions, and opens nano for you by;**   1. **Your choice of either - takes a parameter or asks the user using read - for a filename**      1. **Add ".sh" to the end of the filename supplied (so "example"**   **becomes "example.sh")**     1. **Outputs (appends) "#!/bin/bash" into the file given by the filename above**      1. **Changes permissions on the file to make it executable**      1. **Opens nano to edit the file**      1. **Test the above script by using it to create a bash script called "t5test ". Check that it creates "t5test.sh", fills the shell header, sets the permissions, and opens nano.** |  |