

Scripting Languages: Workshop 7

Pre-requisites:

- If you have not already done so, log into your Linux instance, start VS Code and navigate your way into the **ws7** folder.
- To complete these tasks you will need to place the following three (3) files into your current working directory (from *Canvas* -> *Week 7* -> *Workshop Tasks* -> *ws_files.zip*):
 - **inputfile.html**
 - **cleaner.sh**
 - **server_logs.csv**
 - **formatter.sh**
 - **employee_records.csv**

Write the Code

Task 1

1. Write a script named **txtproc.sh** that automates the process of updating a html file to remove legacy tags and make other changes to meet specific requirements
2. To develop and test your script, use the **inputfile.html** file
3. Ensure your script uses **sed** to achieve the following outcomes:
 - a. Converts instances of **** to **** and **** to ****
 - b. Converts instances of **<i>** to **** and **</i>** to ****
 - c. Corrects the misspelling **kernal** to **kernel** in all cases
 - d. Converts all instances of **http** to **https**
 - e. Pushes each new paragraph **<p>** to a new line
 - f. Converts each instance of **<p>** to **<p style="standardtext">**
 - g. Write the processed text to a file named **inputfile_u.txt**
4. Once the script has executed, run the command **cat inputfile_u.txt**, and you should get the following output to terminal:

```

$ ./txtproc.sh
$ cat inputfile_u.html
<h2>The creation of Linux</h2>
<p style="standardtext">In 1991, while studying computer science at the <a href="https://www.helsinki.fi/en">
University of Helsinki</a>, <strong>Linus Torvalds</strong> began a project that later became the Linux kerne
l. He wrote the program specifically for the hardware he was using and independent of an operating system bec
ause he wanted to use the functions of his new PC with an 80386 processor. Development was done on MINIX usin
g the GNU C Compiler. The GNU C Compiler is still the main choice for compiling Linux today, but can be built
with other compilers, such as the Intel C Compiler.</p>
<p style="standardtext">As Torvalds wrote in his book <em>Just for Fun</em>,he eventually ended up writing an
operating system kernel. On 25 August 1991, he (at age 21) announced this system in a Usenet posting to the
newsgroup <em>comp.os.minix</em></p>
<p style="standardtext">According to Torvalds, Linux began to gain importance in 1992 after the Windows Syste
m was ported to linux by Orest Zborowski, which allowed Linux to support a GUI for the first time.</p>
<p style="standardtext">Linus Torvalds had wanted to call his invention Freax, a portmanteau of "free", "frea
k", and "x" (as an allusion to Unix). During the start of his work on the system, he stored the files under t
he name "Freax" for about half of a year. Torvalds had already considered the name "Linux", but initially dis
carded it as too egotistical.</p>

```

5. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the **txtproc.sh** script again or ask your tutor for assistance

Task 2

1. Write a script named **getjpgs.sh** that creates a list of .jpeg images present in a nominated web page (URL)
2. If no .jpeg files are found at the URL provided, the user is to be advised accordingly
3. When .jpeg files are found at the URL provided, display them to the terminal, accompanied by an accurate count of how many were found
4. If you encounter an error, read the error message printed to the terminal carefully and attempt to resolve the issue and run the **getjpgs.sh** script again or ask your tutor for assistance

Comment the Code

Task 3

1. Download the files *cleaner.sh* and *server_logs.csv* to your Linux development environment into the *ws7* directory you created in Week 1. The *server_logs.csv* file is the data source the *cleaner.sh* script will act upon.
2. Using only the lecture notes (Modules 1-7 inclusive) and what you have learned so far, fully comment the *cleaner.sh* script to explain:
 - a. The purpose of the script
 - b. Its inputs
 - c. Its main processing logic
 - d. Its outputs

Do **not** run the script before you comment it. Complete the commenting in full and then run the script to see how much of your commenting was accurate.

Do **not** ask any AI tool to comment the script for you, otherwise you will learn nothing!!!

Fix/Debug the Code

Task 4

1. SCENARIO: As the senior programmer, you are quite happy with the *cleaner.sh* script, however, you did point out that the dates in the first column of the output file are in U.S. format rather than

Australian format as required. The junior programmer has gone away and tried to fix the issue, but has not been successful.

2. Guided the junior programmer in what they need to do and modify the script as required so that it produces the dates in the required Australian format; call this file *cleaner_corrected.sh*

Use only the lecture notes (Modules 1-7 inclusive) and what you have learned so far to guide you in this process

Use comments to identify/document the issues within the script

*Do **not** ask any AI tool to tell you what the issues are or how to fix them, otherwise you will learn nothing!!!*

Critique the Code

Task 5

1. Download the files *formatter.sh* and *employee_records.csv* to your Linux development environment into the *ws7* directory you created in Week 1. The *employee_records.csv* file is the data source the *formatter.sh* script acts upon.
2. SCENARIO: You asked a junior team member to write a shell script that processes the contents of a .csv file input by the user that a) removes duplicate header rows if they are present, b) anonymises the salary field by replacing the salary amount with the string "Confidential", c) changes the *DateOfJoining* column from US date format to Australian date format, and writes the processed results to a new .csv file with the header preserved. The junior team member has now come to you with the script they've written (*formatter.sh*) and asked if you will approve it for production use. As the senior team member, would you approve this script for use in production? If not, record a short Panopto video explaining to the junior team member why you will not approve their script for production and outline what they need to do to make it acceptable for production use. Then send this video to your lecturer along with your version of the script (call it *formatter_better.sh*) to show the junior team member how you would have coded it as a learning opportunity for them.

*Do **not** ask any AI tool to critique the junior team members script for you or write a more efficient version, otherwise you will learn nothing!!!*

Task 6

1. **Copy** the .sh files you created in today's workshop to the *backups* directory using the same *_bu* name modification you used in last week's workshop
2. Navigate to the *backups* directory and make sure the copy procedure was successful

Conclude:

Close the *RDP connection* to your Azure VM (if you're using one) and then **power off** your VM in Azure.

