COMP1204 - Data Management
UNIX Coursework Report

Christos Mousoulides

Student ID: 31225551
Email: cm10g19@soton.ac.uk

March 09, 2021

0.1 Introduction

This report aims to analyse the creation of the countreviews.sh file and give a step-by-step guide on its evolution to fulfill each task of the coursework. In total there are 4 major versions of this bash script that will be discussed below.

0.2 First Implementation

The first version of the countreviews.sh file was created in order to satisfy the requirement of the first task of the coursework which is to create a UNIX script that counts the number of reviews located in a .dat file, that is taken as input from the command line.

```
1 #!/bin/bash
2
3 folderpath="$1"
4
5 cat "$folderpath" | grep '<Author>' | cut -d'>' -f2 | wc -l
```

Firstly, line 3 of the code contains the "folderpath" variable that stores the file taken as input from the command line. Then on line 5 the cat command is used to display the contents of that file and it is piped to the grep command. The grep command leaves only the lines that contain the tag '<Author>', as there's only one author for each review. Then the output of that command is passed onto the cut command which removes everything after the '>' symbol. Finally, the wc command is used to count each line that remains in the file and gives out the correct amount of reviews for the file given.

0.3 Second Implementation

The second version of the countreviews.sh file has a few things added onto the original file that make it satisfy the requirements of the second task of the coursework. This task counts the number of reviews in each file given the folder from the command line.

```
1 #!/bin/bash
2
3 folderpath="$1"
4
5 for i in "$folderpath"/*
6 do
7 cat $i | grep '<Author>' | cut -d'>' -f2 | wc -l
8 done
```

The major change with this implementation is that the code of the 1st version is wrapped around a for loop. More specifically, in line 5 the loop iterates through everything in the folder path that is given from the command line. Then, on

line 7 the cat command now opens the contents of the file i which is inside the folder given. Finally, the output of this bash script is the number of reviews that each hotel data file contains.

0.4 Third Implementation

The third version of countreviews.sh changes the file in order to complete the requirements of the final task which are to display each hotel data file name and review count separated by one line and then sort them in descending order.

```
1  #!/bin/bash
2
3  folderpath="$1"
4
5  for i in "$folderpath"/*
6  do
7  nOReviews=$(grep -c '<Author>' $i | cut -d'>' -f2)
8  hotelID=$i
9
10 output="${hotelID} ${nOReviews}"
11 echo $output;
12 done | cut -d'/' -f2 | sed "s/.dat//" | sort -k2rn
```

This implementation changed the file by storing the 7th line of the previous implementation in to the "nOReviews" variable and also removing unnecessary commands like cat and wc, by using grep's -c command which counts the instances of author by itself. Then, on line 8 the "hotelID" variable stores the path of each hotel file. In line 10 the "output" variable specifies how the output should look like with the hotelID and n0Reviews being separated by one line. Additionally, in line 12 the cut command receives the output of the for loop and separates the folder path, so that only the file name is displayed. Then the sed command substitutes the .dat extension of the file with nothing, effectively leaving only the file name present. Finally, the sort command sorts the 2nd column of the output and displays the numbers in descending order (k2rn). Overall, this is a failed implementation of the final task, because it does not handle both the absolute and relative paths of the folder path. Also, the use of a for loop and many variables slows down the speed of execution significantly.

0.5 Final Implementation

The fourth and final version of the countreviews.sh file entirely alters its contents and successfully meets the requirements of the final task of the coursework.

```
1  #!/bin/bash
2
3  folderpath="$1"
```

```
4 cd $folderpath
5
6 grep -c '<Author>' hotel* | sed -s 's/.dat:/ /' | sort -k2rn
```

Firstly, the folder path that is stored in line 3 is used in line 4 with the cd command so that the directory is changed to where the folder path was specified in the command line. Then, the biggest changed that occured in this file was the removal of the for loop as it was not necessary to complete this task, because the grep command can iterate through all the files in the folder by itself without needing a specification of each file to loop through. Thus, in line 6 the grep command counts the number of instances of '<Author>', (-c), that occured in files starting with hotel in the folder, (hotel*). The files are output in such a way that the filename is displayed with its extension followed by a colon and the number of reviews per file. Then, the sed command substitutes the ".dat:" portion of the text with a space and the -s command is used so that each file is treated separately. Finally, the sort command is used to sort the 2nd column of the text and displays the review numbers in descending order (-k2rn).