

Operating Systems CS/CE 232L/324L

Lab 02

Ali Muhammad Asad
aa07190

Exercises:

Prompt: Create a text file with at least three columns and ten lines. Name it as your student ID, and have dummy data in the file, however, the third column should be integer values.

Command: `touch aa07190.txt`

The above command creates a txt file, and then the file can be opened in a text editor and updated accordingly.

Command: `nano aa07190.txt`

This command creates and opens the txt file in the terminal where changes can be made

Dummy data:

“

Column1	Column2	Column3
value1	valueA	1
value2	valueB	2
value3	valueC	3
value4	valueD	4
value5	valueE	5
value6	valueF	6
value7	valueG	7
value8	valueH	8
value9	valueI	9
value0	valueJ	10

”

Tasks:

1) Change the permissions of your newly created dummy data file to enable read and write access for **others** set of people.

Ans: `chmod o+rw aa07190.txt`

The above command changes the permissions of the file to allow read and write access to others. ‘chmod’ is the command to change file permissions, ‘o’ stands for others, and ‘+rw’ stands for read and write access where ‘r’ is read and ‘w’ is write.

2) Show the first three lines of your dummy data file.

Ans: `head -n 3 aa07190.txt`

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ head -n 3 aa07190.txt
Column1 Column2 Column3
value1  valueA  1
value2  valueB  2
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ |
```

The ‘head’ command displays the head of the file, and the rest displays the first 3 lines of the file.

3) Show the last four lines of your dummy data file.

Ans: `tail -n 4 aa07190.txt`

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ tail -n 4 aa07190.txt
value7  valueG  7
value8  valueH  8
value9  valueI  9
value0  valueJ  10
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ |
```

The ‘tail’ command displays the tail of the file, and the rest displays the last 4 lines of the file.

4) Show the sixth line of your dummy data file.

Ans: `sed -n '6p' aa07190.txt`

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ sed -n '6p' aa07190.txt
value5  valueE  5
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ |
```

5) Display the third column of your dummy data file.

Ans: `cut -f 3 aa07190.txt`

Alternatively we can also use:

`awk '{print $3}' aa07190.txt`

Both commands print or extract the third field / column from the .txt file. The display for both the commands is as follows:

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ cut -f 3 aa07190.txt
Column3
1
2
3
4
5
6
7
8
9
10
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ awk '{print $3}' aa07190.txt
Column3
1
2
3
4
5
6
7
8
9
10
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ |
```

6) Display the first column of your dummy data file in sorted order.

Ans: `cut -f 1 aa07190.txt | sort`

The above command first extracts the first column from the .txt file and then sorts it.

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ cut -f 1 aa07190.txt | sort
Column1
value0
value1
value2
value3
value4
value5
value6
value7
value8
value9
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ |
```

7) Display the maximum value in the third column of your data file.

Ans: `cut -f 3 aa07190.txt | sort -n | tail -n 1`

Alternatively:

`cut -f 3 aa07190.txt | sort -n -r | head -n 1`

Both the commands give the maximum value in the third column. The first command sorts in ascending order, and then 'tail' is used to extract the first element from the tail. The second command sorts in descending order and the first element is extracted from the head.

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ cut -f 3 aa07190.txt | sort -n | tail -n 1
10
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ cut -f 3 aa07190.txt | sort -n -r | head -n 1
10
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ |
```

8) Obtain the number of words in your dummy data file, and store them in another file (name it count.txt).

Do this in a single command.

Ans: `wc -w < aa07190.txt > count.txt`

The 'wc -w' counts the number of words in the dummy data file. '<' redirects the contents of the dummy data file to 'wc' and '>' redirects the output of 'wc' which is the word count to the count.txt file.

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ wc -w <aa07190.txt> count.txt
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ wc -w aa07190.txt
33 aa07190.txt
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ cat count.txt
33
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ |
```

After running the command, we can see that there were 33 words in our dummy data file, and the count.txt file has been created containing the word count.

9) In count.txt append the number of characters in your dummy data file. Do this in a single command.

Ans: `wc -c < aa07190.txt >> count.txt`

The 'wc -c' counts the number of characters in the file, then similarly the < redirects the contents of the dummy data file to 'wc' and >> redirects and appends the output of 'wc' which is the character count to the count.txt file.

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ wc -c aa07190.txt
185 aa07190.txt
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ wc -c < aa07190.txt >> count.txt
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ cat count.txt
33
185
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ |
```

10) Display the second column of your dummy data file in reverse order (not in alphabetical terms) and store it in another file (name it **reverse.txt**). Use piping to do this in one command.
Ans: Since we know 'tac' is the 'cat' command in reverse, we can simply use 'cut' to extract the second column and use 'tac' to reverse the order.

We can use the following command:

```
cut -f 2 aa07190.txt | tac > reverse.txt
```

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ cut -f 2 aa07190.txt
Column2
valueA
valueB
valueC
valueD
valueE
valueF
valueG
valueH
valueI
valueJ
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ cut -f 2 aa07190.txt | tac > reverse.txt
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ cat reverse.txt
valueJ
valueI
valueH
valueG
valueF
valueE
valueD
valueC
valueB
valueA
Column2
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/Labs$ |
```

The contents of the **reverse.txt** file can be seen to have been reversed as compared to the second column of our dummy data file.

11) Display the first letter of each text value in the first column of your dummy file. Use piping to do this in one command.

Ans: `cut -c 1 aa07190.txt | cut -c 1`

The first `cut -c 1 aa07190.txt` extracts the first character of each line in the first column, and the second `cut -c 1` extracts the first character of each line in the output of the first `cut` command. Therefore, we get the first letter of each text value in the first column of our dummy data file as shown below:

```
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/CS232-Operating-Systems-Fall23/Labs/Lab2$ cut -c 1 aa07190.txt | cut -c 1
C
v
v
v
v
v
v
v
v
v
v
alimuhammad@alimuhammad-Inspiron-7559:~/Desktop/Habib/Sem5/OS/CS232-Operating-Systems-Fall23/Labs/Lab2$ |
```