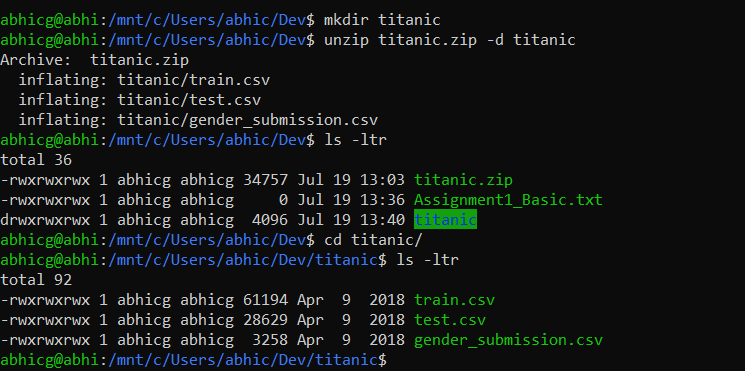
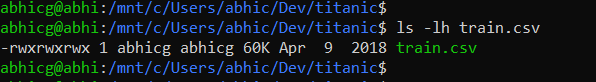
Basic

Use External Tools and Gnu Core Tools to enhance your shell skills. Compose them using pipes and filters.

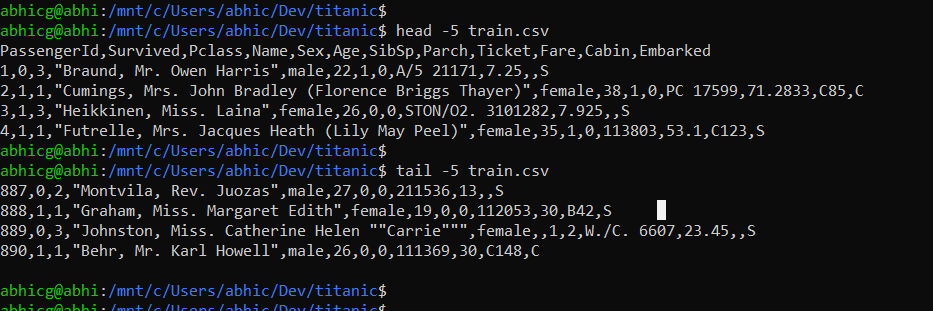
1. Unzip (using the terminal) our ​titanic.zip​ file to ​titanic​,



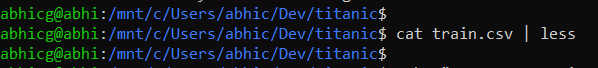
2. Provide the shape/dimensions of the file ​train.csv​?



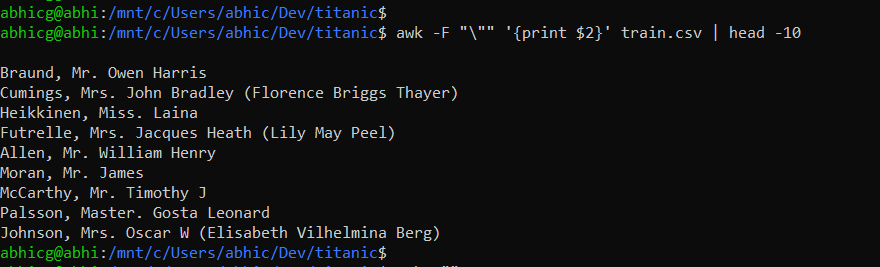
3. List the first 5 rows of the file. Now list the last 5.



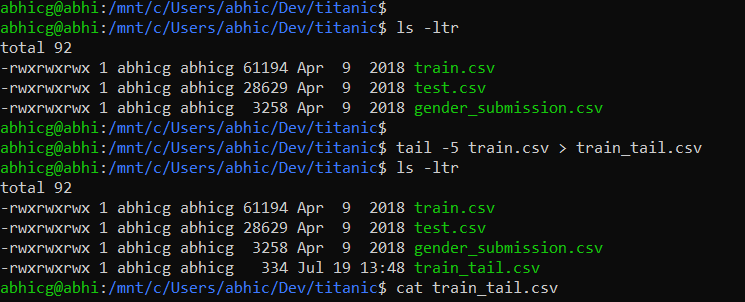
4. Print this file in your screen using ​cat​ now use the ​less​ command.



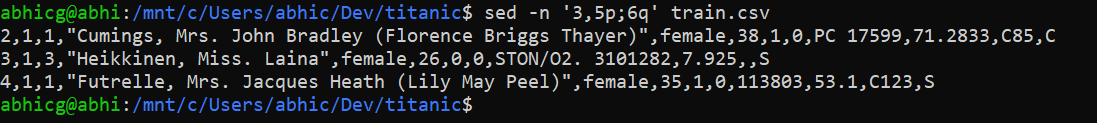
5. Can you print only the names of all people in the file?



6. Print this file last 5 lines save the output to train\_tail.csv



7. Print only the lines 3 to 5 of the file?

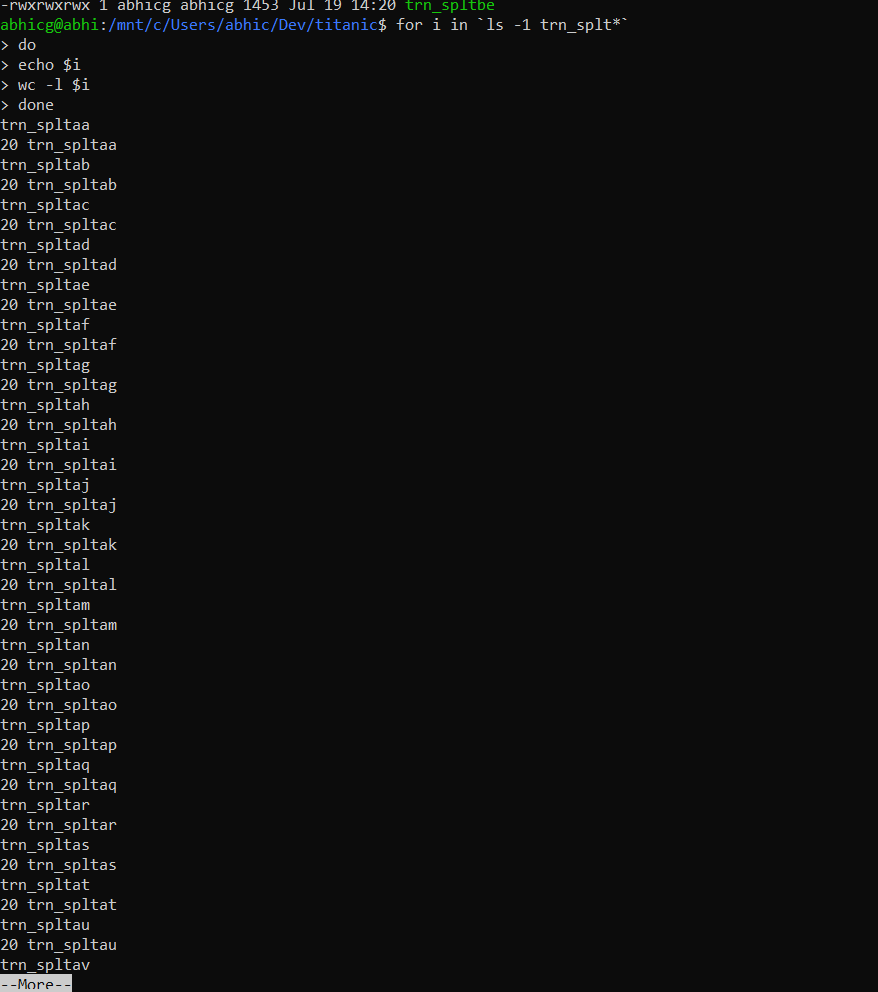


8. Can you explain the command ​du -a . | sort -n -r | head -n 20​ and why would you use it?

**du command is used to get the disk usage summary for all files in the current directory and sort in reverse numeric order to get the top 20 biggest files summary.**

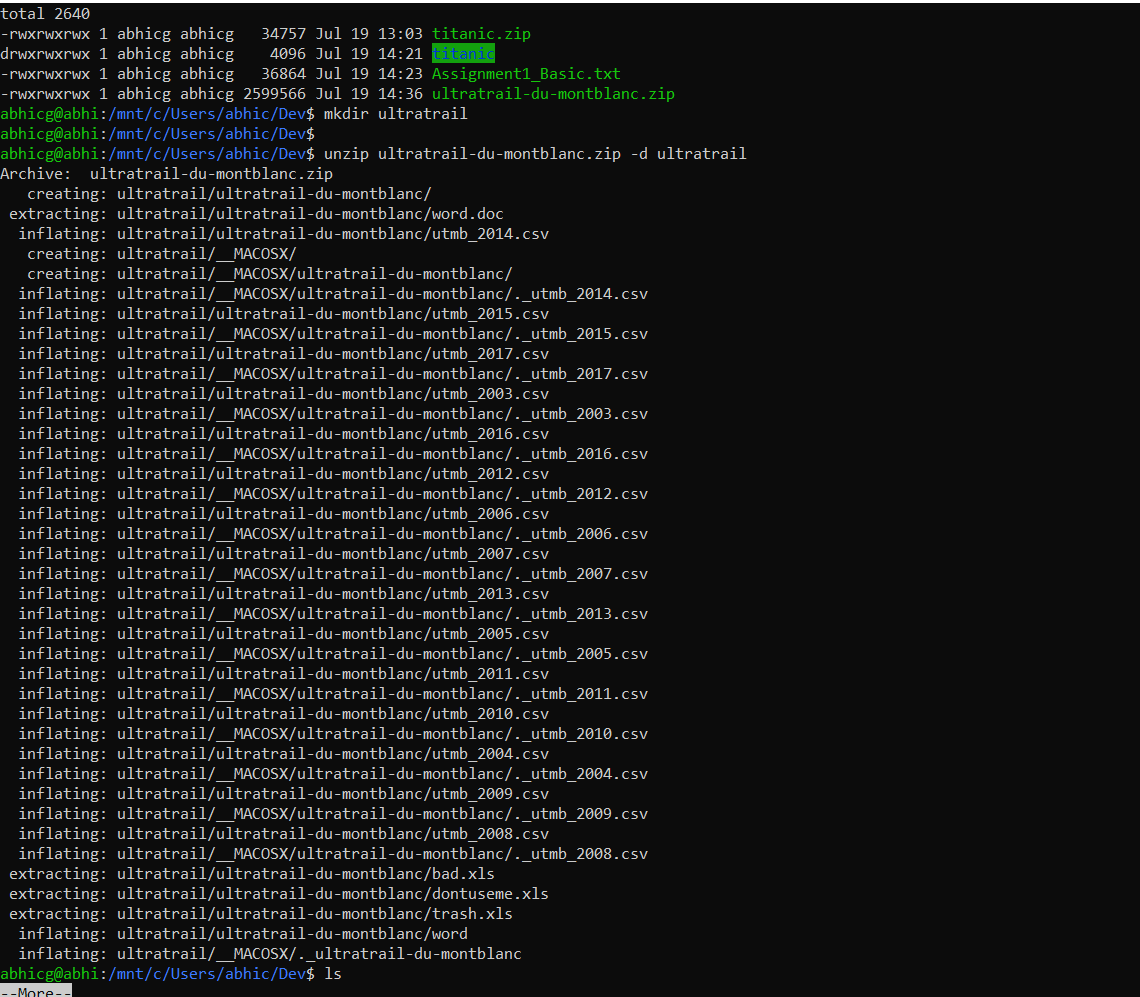
9. Split the train.csv file in multiple files with 20 lines each.



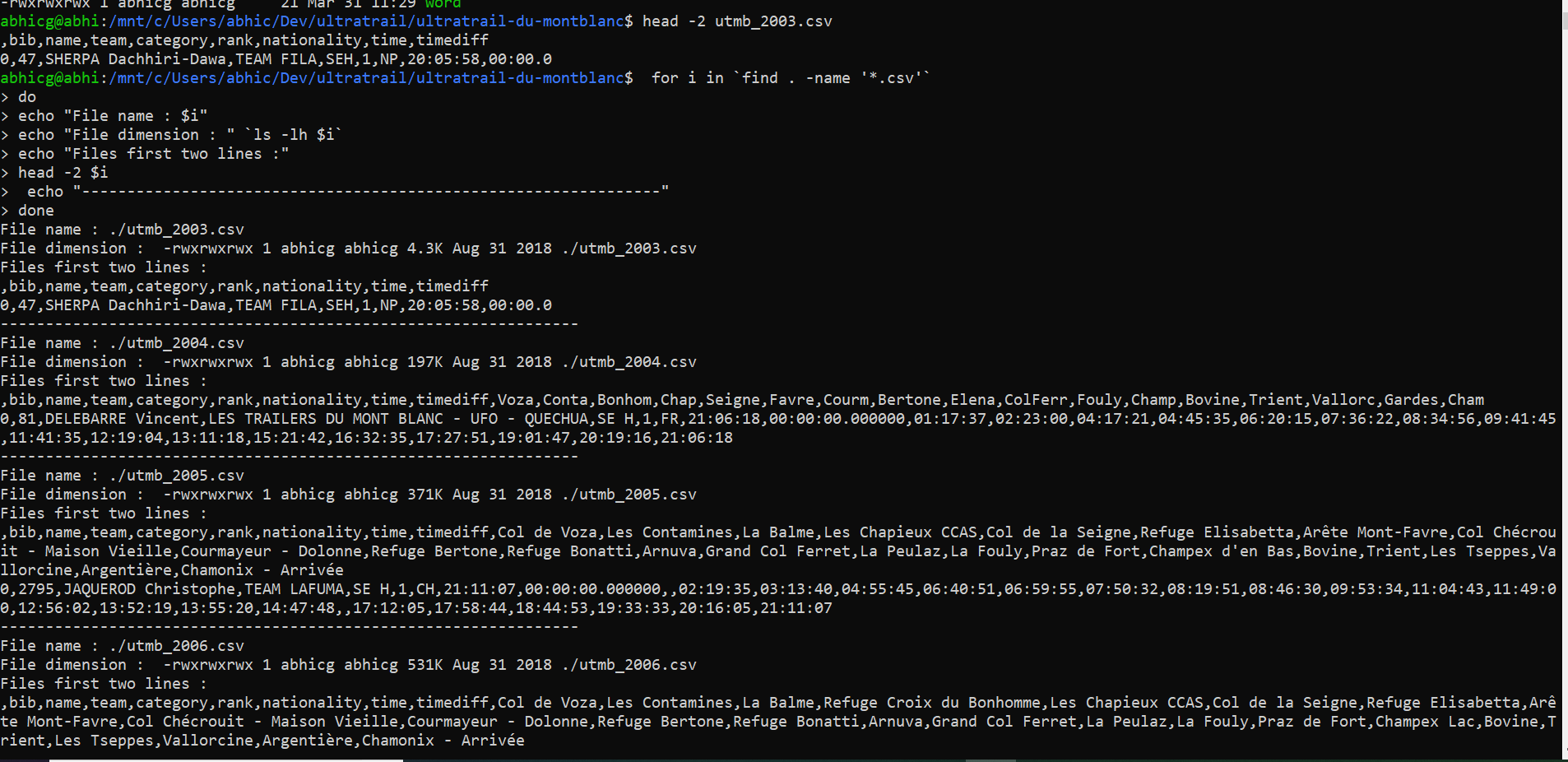


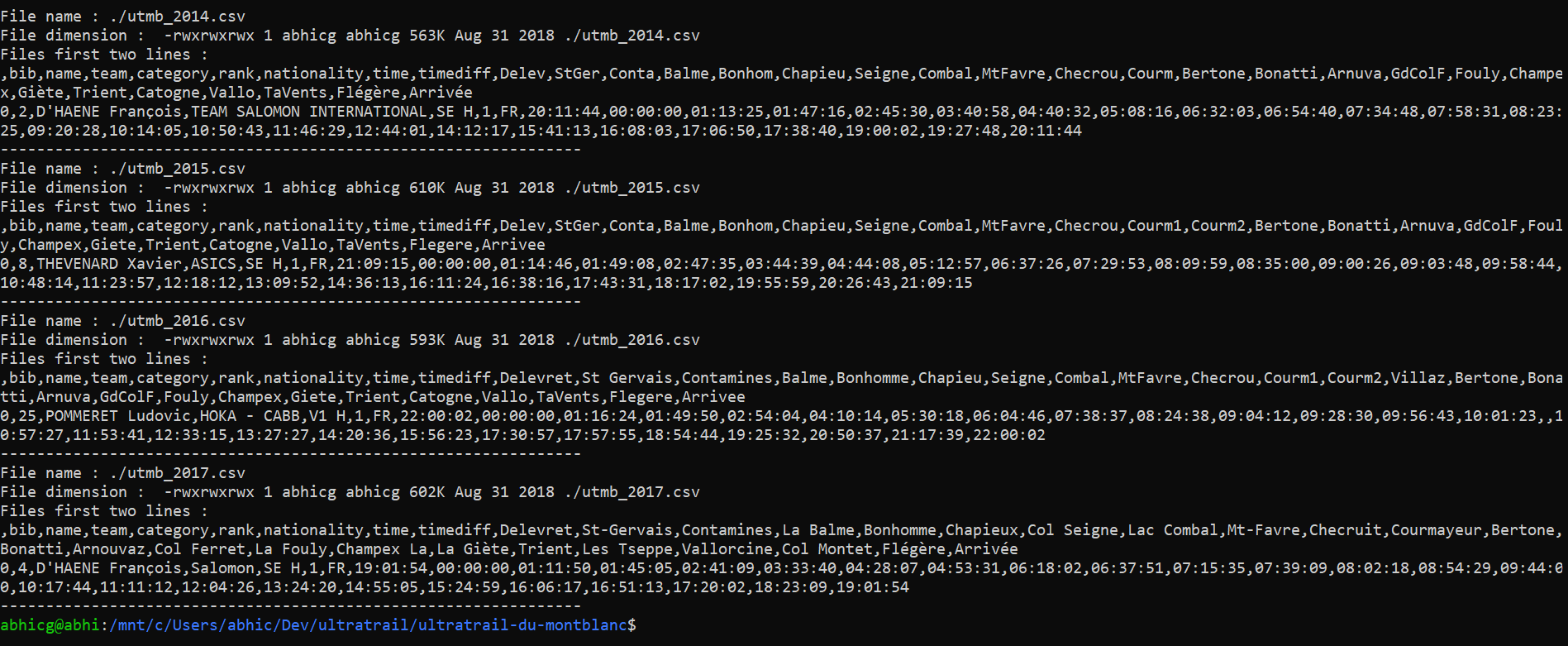
Write loops to iterate over lists

1. Download the ​ultratrail-du-montblanc.zip​ file from Slack and unzip it to /Users/<myusername>/ultratrail

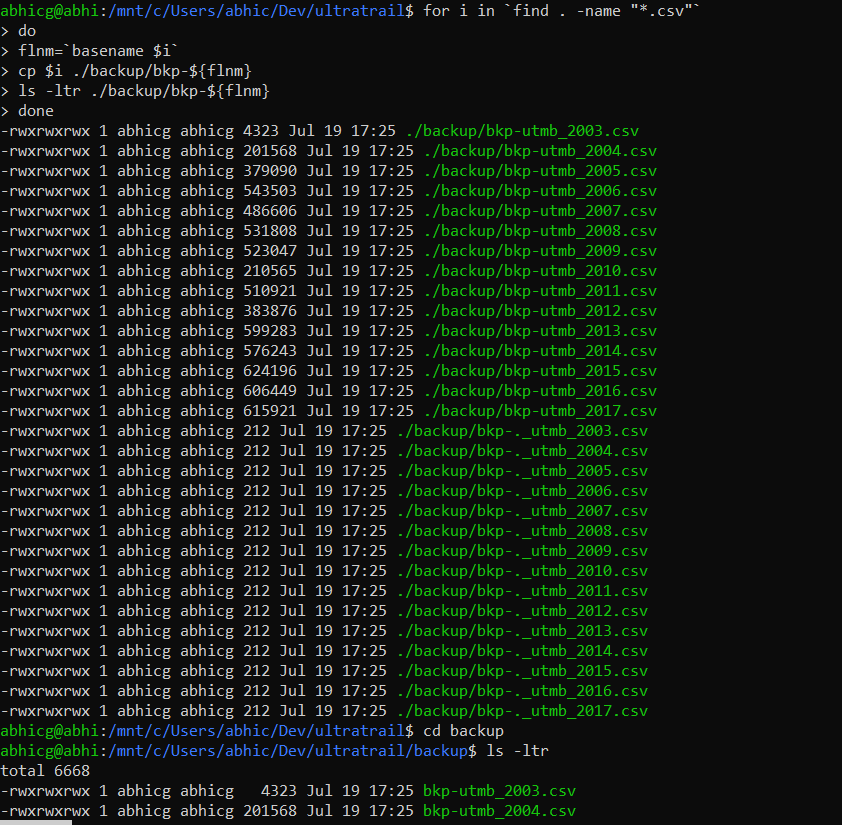


2. Write a loop that prints the name, dimension and first 2 lines for each of the ​.csv files.





3. Write a loop that copies each of the ​.csv​ files with the prefix ​bkp-​ to a folder /Users/<myusername>/ultratrail/backups​.



Reach

Create scripts to automate basic processes

1. Write a script that suggests the data formats: csv, xlsx, pdf, doc and txt. It should allow the user to pick their desired extension then create a file named selected.<extension selected>​. Use the ​read​ command to read the user input!

select\_extn.sh

#!/bin/bash

create\_file()

{

echo "The Selected file extension is : ${1}"

touch selected.${1}

rc=$?

if [ $rc -eq 0 ]; then

echo "Successfully created the file selected.${1} !!!"

else

echo "Failed to create the file !!!"

fi

}

while true; do

echo "Select one of the file extension options"

echo "1 to create .csv file"

echo "2 to create .xlsx file"

echo "3 to create .pdf file"

echo "4 to create .doc file"

echo "5 to create .txt file"

echo "6 to exit"

read file\_extn

case $file\_extn in

1) create\_file csv

;;

2) create\_file xlsx

;;

3) create\_file pdf

;;

4) create\_file doc

;;

5) create\_file txt

;;

6) exit 1

;;

\*) echo "Invalid input"

;;

esac

done

2. Write a script that keeps only the first N number of lines of all files in '/Users//files\_to\_clean/\*.csv'. N should be an argument passed before starting the script! If other people depend on this being done daily, how can we automate it's daily execution at 8:00AM?

abhicg@abhi:/mnt/c/Users/abhic/Dev/files\_to\_clean$ cat clean\_up\_script.sh

#!/bin/bash

echo "Input the number of lines to preserve in the files under files\_to\_clean directory :"

read line\_num

echo $line\_num

if [ $(echo $line\_num | grep -qE '^[0-9]+$'; echo $?) -ne "0" ]; then

echo "Error : Not a number - $line\_num"

else

echo "Valid number - $line\_num"

fi

echo "The number of files in the current directory : " `find . -name "\*.csv" | wc -l`

for i in `find . -name "\*.csv"`

do

echo "Record count before file modification :" `wc -l $i`

filename=`basename $i`

tail -${line\_num} $i > ./bkp\_${filename}

mv ./bkp\_${filename} ${i}

echo "Record count after file modiciation :" `wc -l ${i}`

done

abhicg@abhi:/mnt/c/Users/abhic/Dev/files\_to\_clean$

Using crontab we can schedule in our case to set for daily at 8:00 am

The syntax is

minute(0-59) hour(0-23) day(1-31) month(1-12) weekday(0-6) command

The entries will be as below

0 8 \* \* \* /mnt/c/Users/abhic/Dev/files\_to\_clean/clean\_up\_script.sh