## 5:W35: Managing Files on Steroids with Shell

1. *Identify the names and format of the 3 biggest files. Can you come up with a command to generate a numerically ordered list of 3 biggest files? (hint: consider using****wc****to gauge image size)*

*Pipeline used to generate answer:*

wc -c \* | sort -rn | head -n 4 | tail -n 3 > biggest\_3\_files.txt

*Step-by-step explanation of the pipeline:*

1. First, we find the byte size of all of the files in the current repository
2. Then we sort numerically and in reverse order, so the biggest files are listed in a descending order, from top to bottom
3. Then we take the top 4 rows (the top row is not at file name, but an indication of the total amount of bytes)
4. To then get the actual 3 biggest files, we take the bottom three files
5. Finally, the output is saved to the text file “biggest\_3\_files.txt”

*Answer:*

*Text

Description automatically generated with low confidence*

1. *Some of the image files are empty, a sign of corruption. Can you****find****the empty photo files (0 kb size) , count them, and generate a list of their filenames to make their later replacement easier?*

This answer to this query is two-fold – the first pipeline counts the files that are of size 0 kB and the second pipeline generates a list of their filenames:

*Pipeline 1 – counting:*

wc -c \*.JPG | sort -n | cut -wf 2 | uniq -c | head -n 1

*Step-by-step explanation of pipeline 1:*

1. First, we find the byte size of all the .JPG files in the current repository
2. Then we sort the output numerically, from smallest to largest files, top to bottom
3. Consequently, using whitespace as the delimiter, we take the second field of the output, which corresponds to the bytesize of the .JPG files
4. Afterwards, we use the uniq command to count how many of the bytesize fields are equal to 0 (the command also gives us how many instances there are of other bytesizes).
5. We then take the uppermost row, which is the count of files with bytesize equal to 0.

*Pipeline 2 – list generation:*

wc -c \*.JPG | sort -n | head -n 73 | cut -wf 3 > files\_size\_0.txt

*Step-by-step explanation of pipeline 2:*

1. First, we find the byte size of all the .JPG files in the current repository, same as in pipeline 1
2. We then sort the output numerically
3. We then take the 73 (the count we got from pipeline 1) uppermost files
4. Afterwards, we take out only the field with the filenames, since we don’t want the bytesize information
5. Finally, we save the list of .JPG files with size 0 to a .txt file called ‘files\_size\_0.txt’