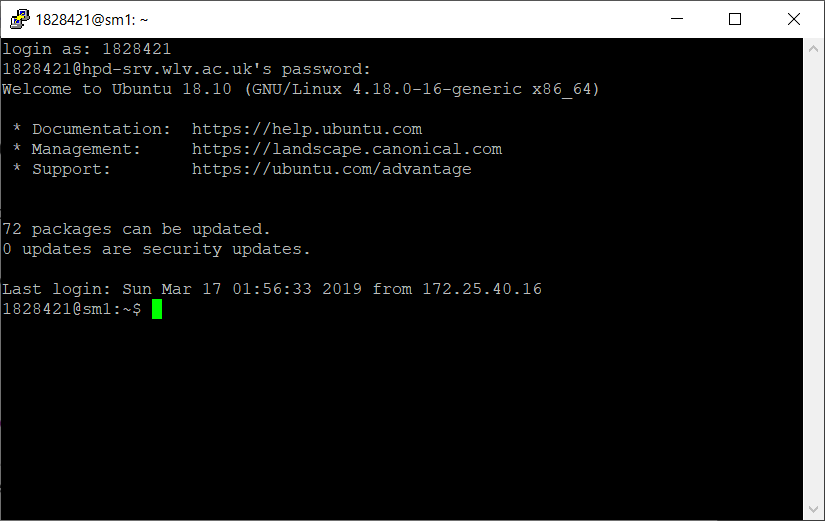
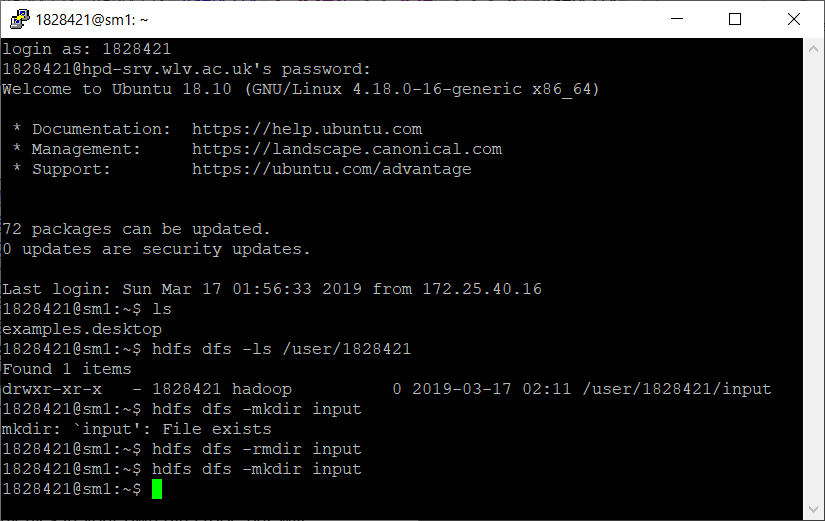
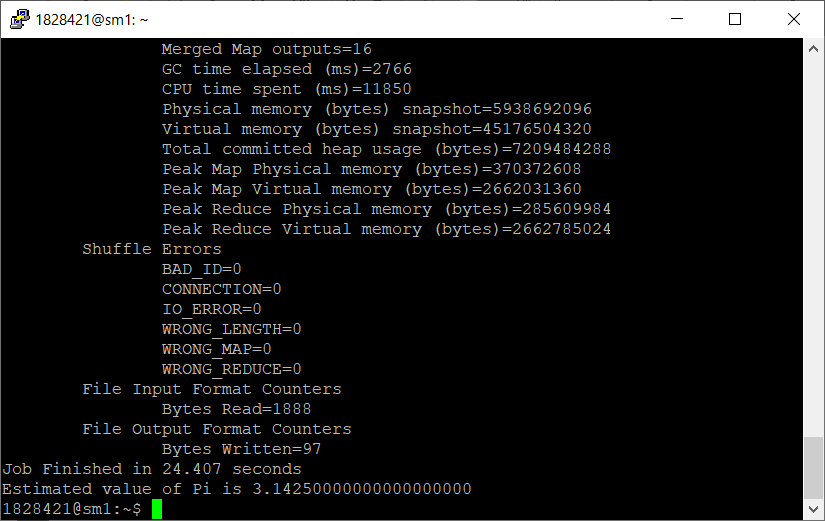
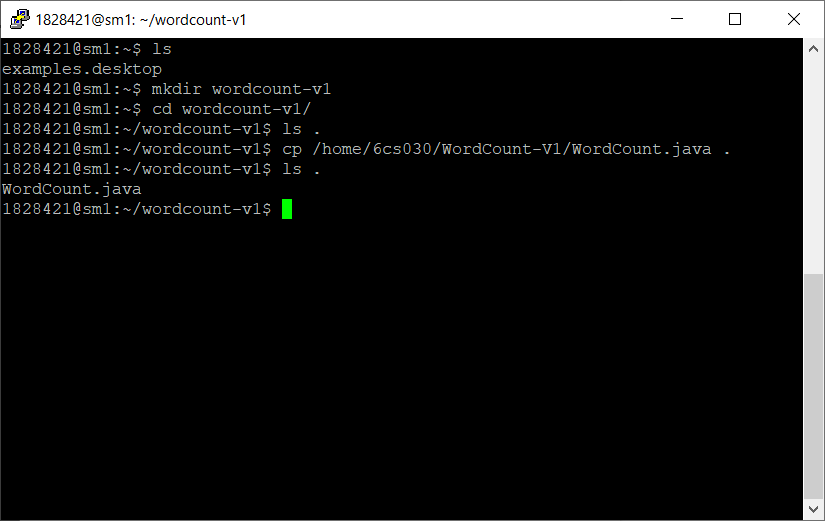
1. Login:
2. Making input directory:



1. Pi program:

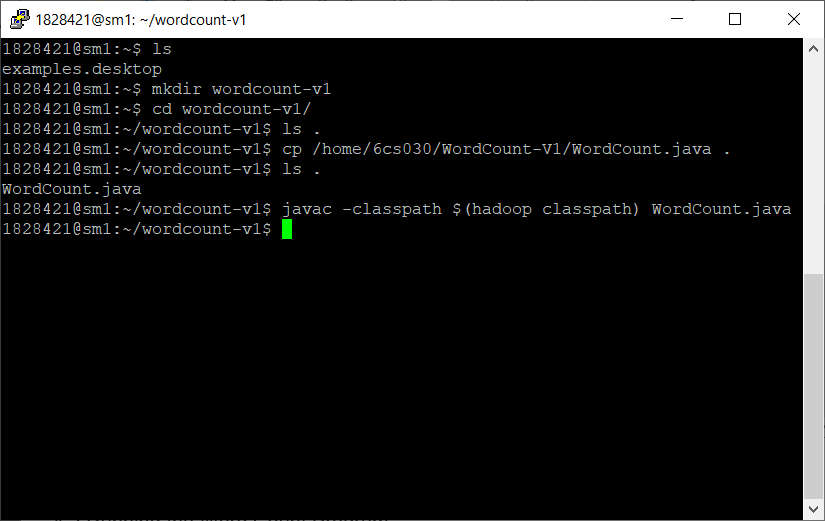


1. Word Count.java
   1. Making dir wordcount-v1 and copying file to this created dir.

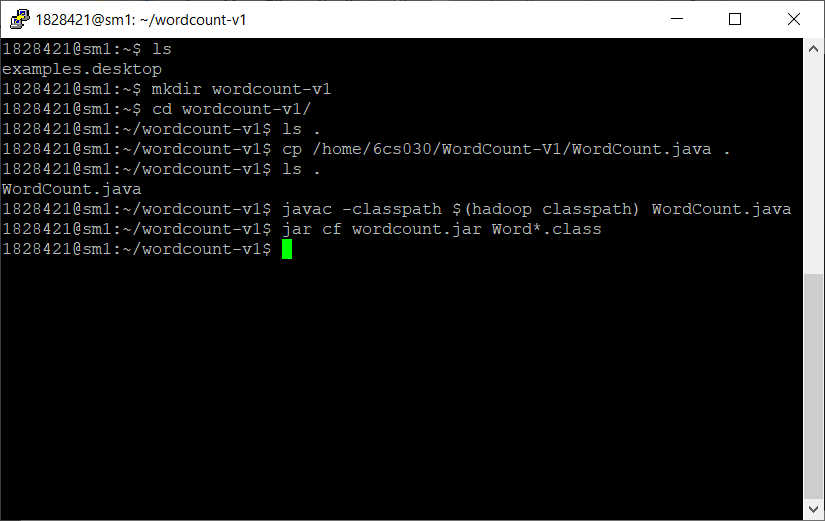


* 1. Running the Word Count Program

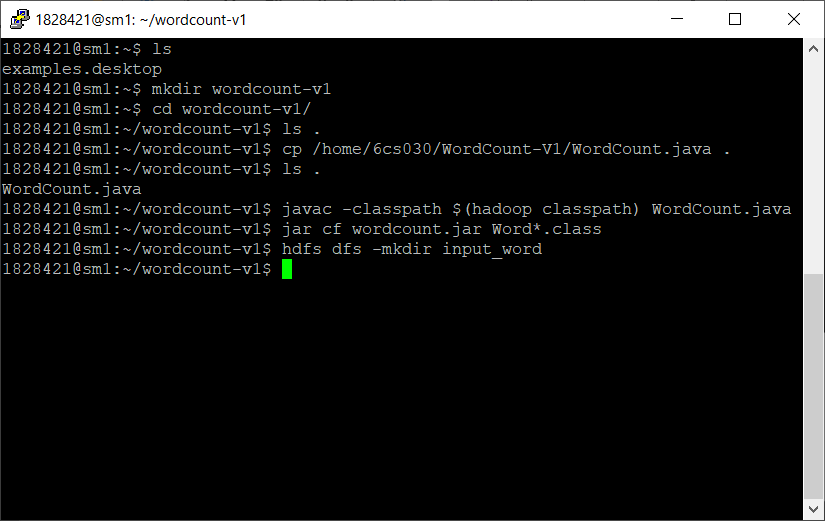
1. Compiling the file



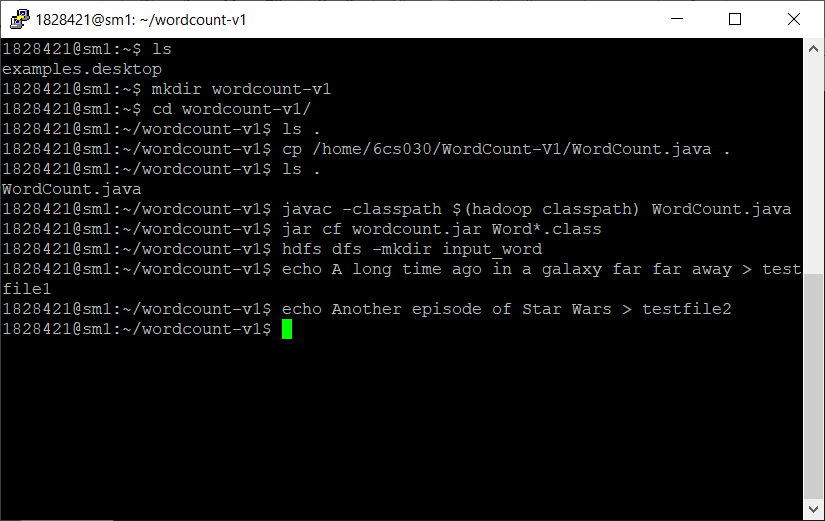
1. Producing the Jar file



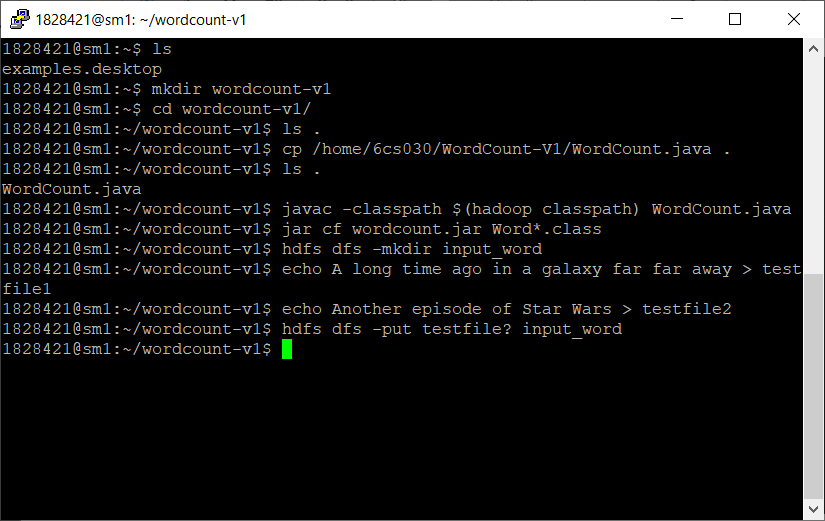
1. Creating input directory on the hdfs:



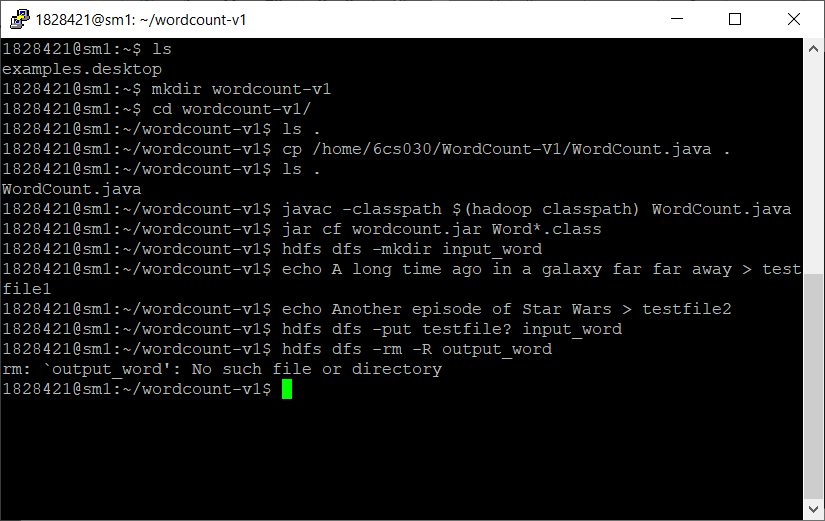
1. Create the input files:



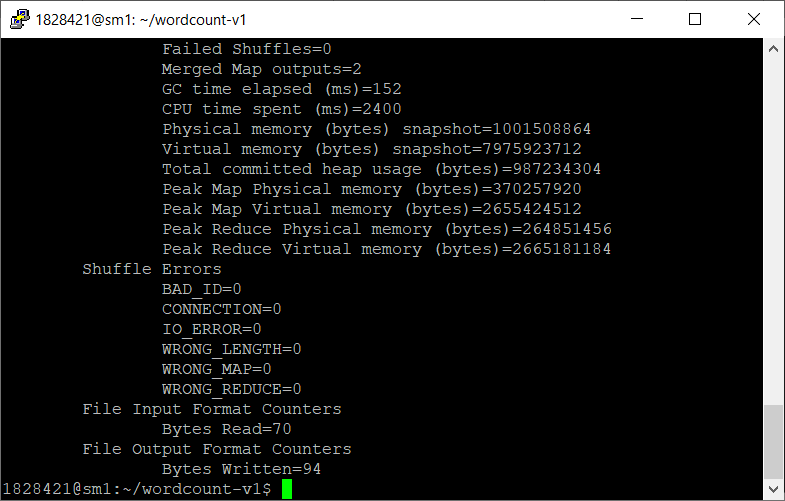
1. Save the files to input dir



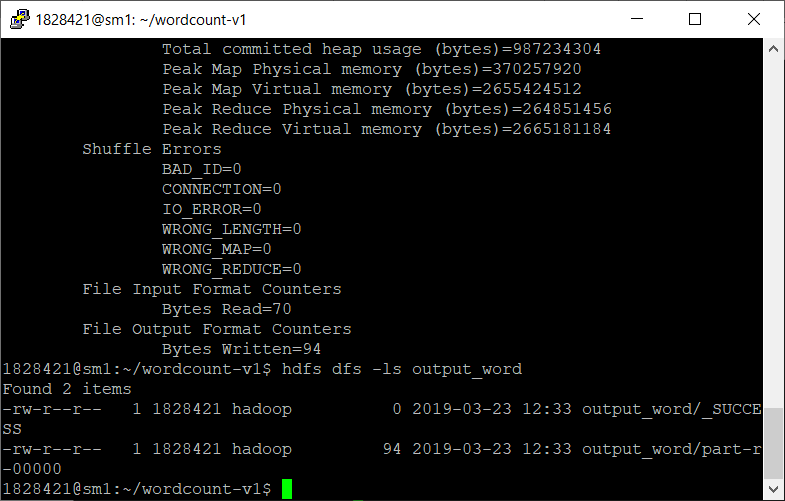
1. Removing file if it exists:



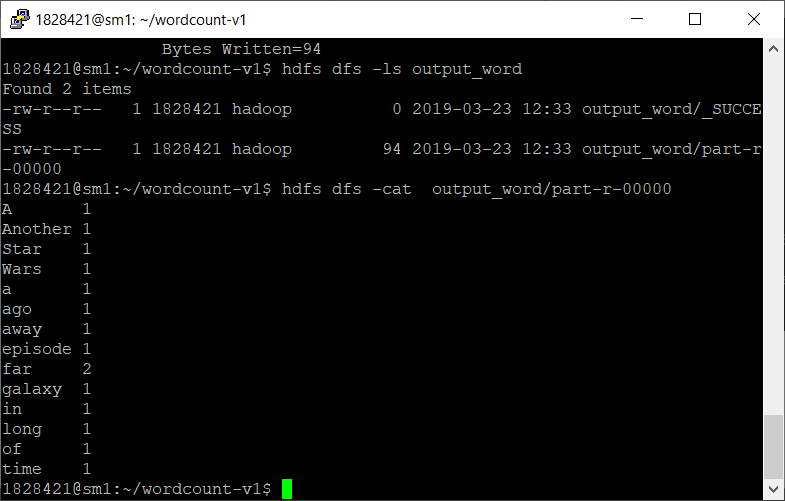
1. Running map reducing program



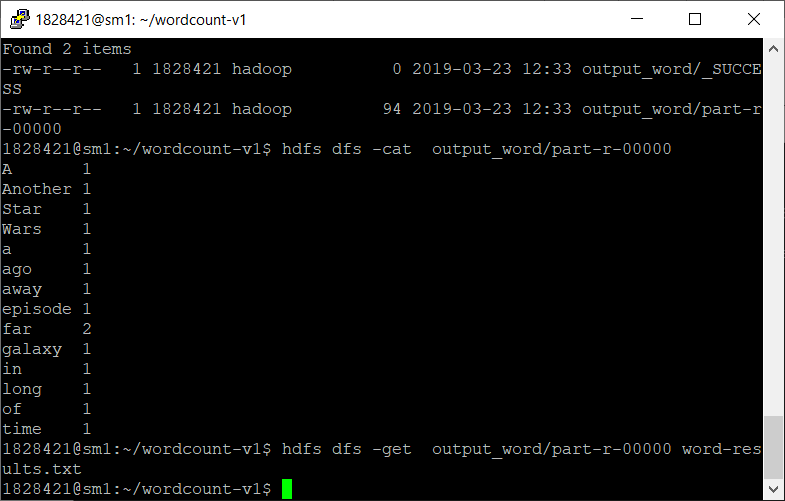
1. Check what files are in the output dir:



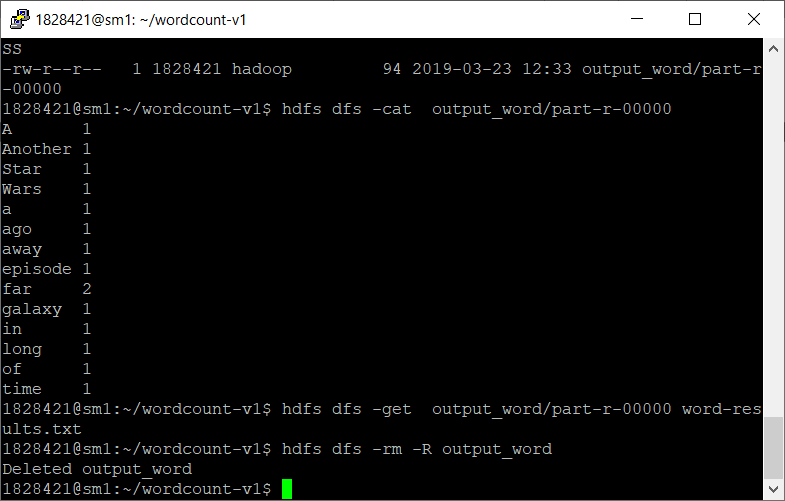
1. What is inside input file:

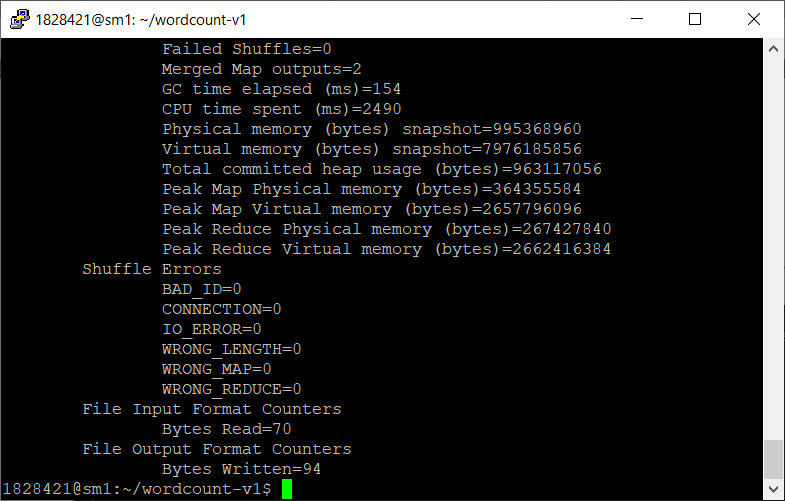


1. Retrieving file outside hdfs:

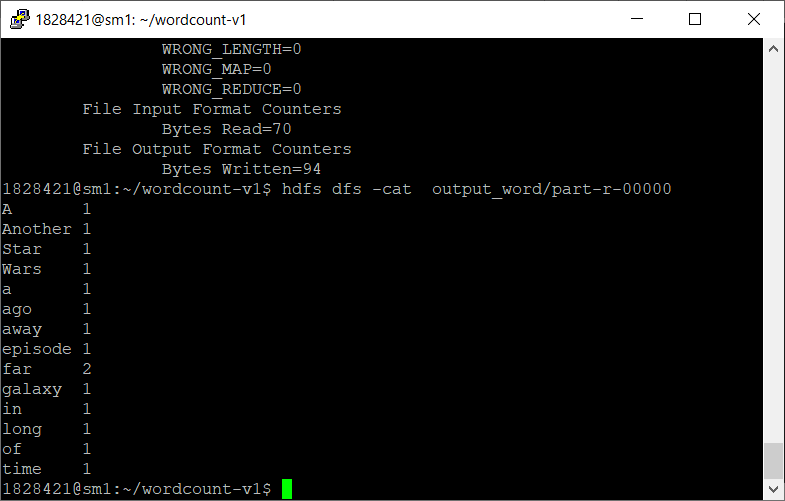


* 1. Rerunning the word count:

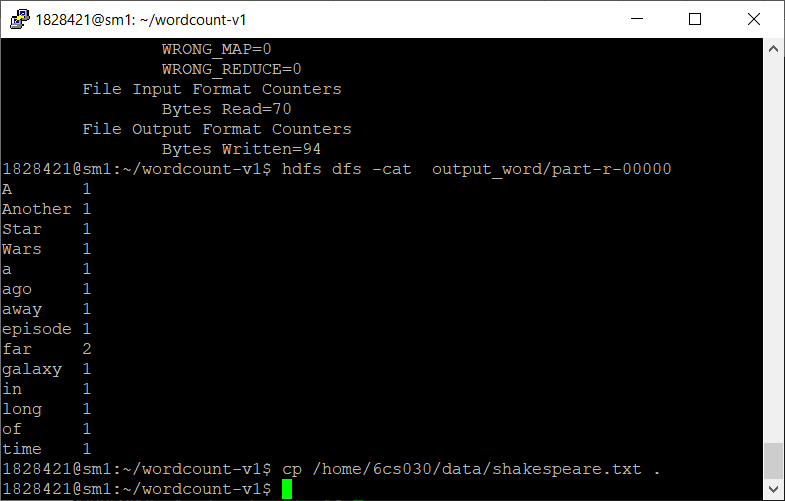




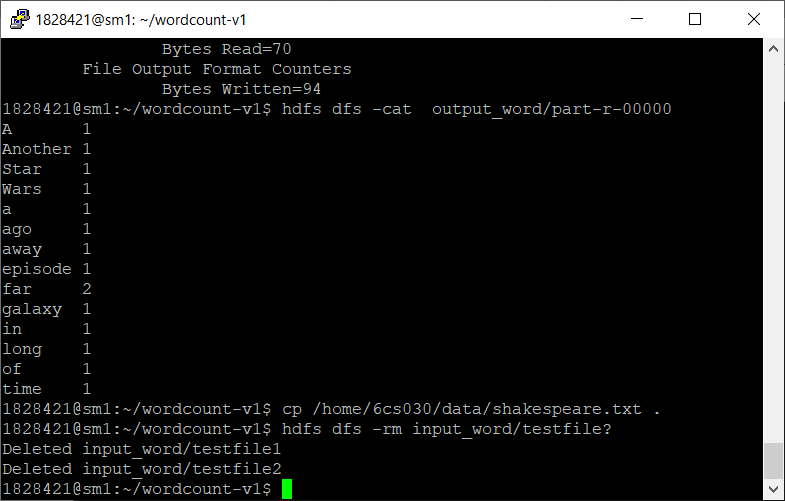
View output file:



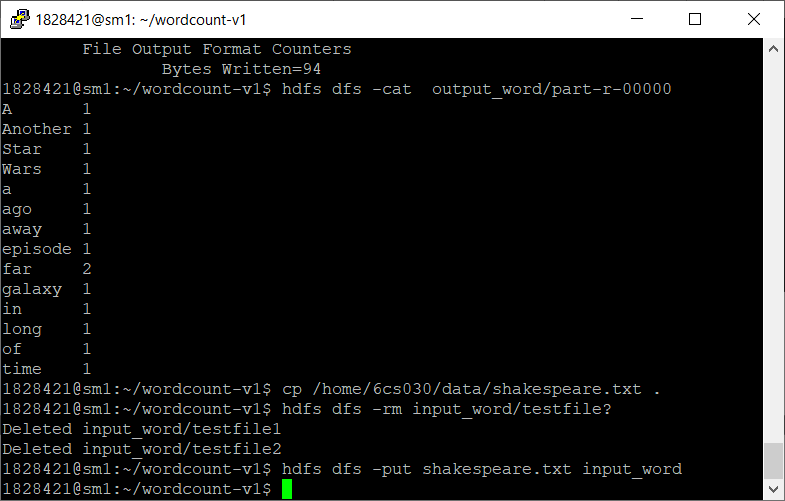
* 1. Using larger dataset



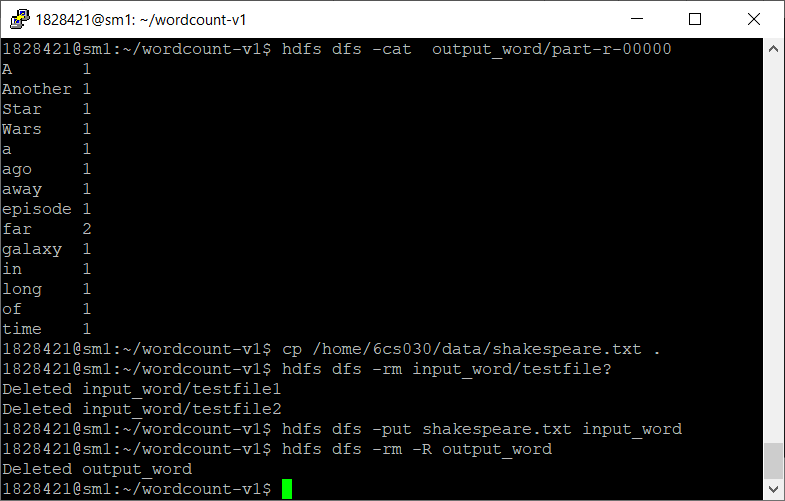
1. Removing the previous test files:



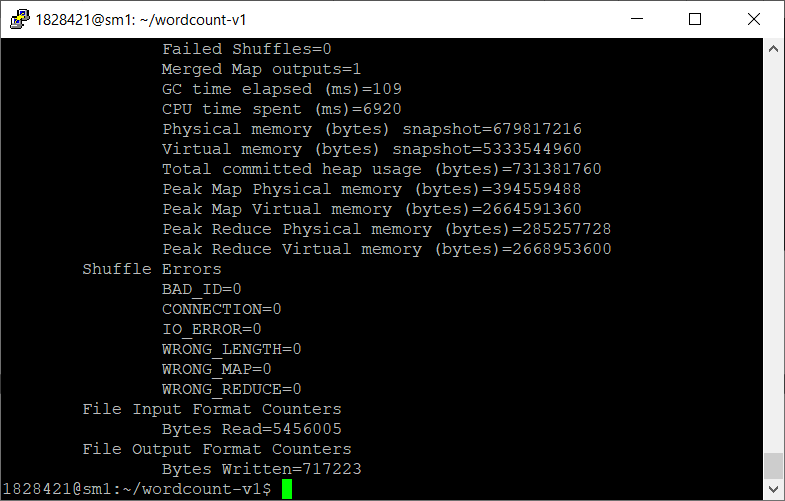
1. Saving new file to the input directory



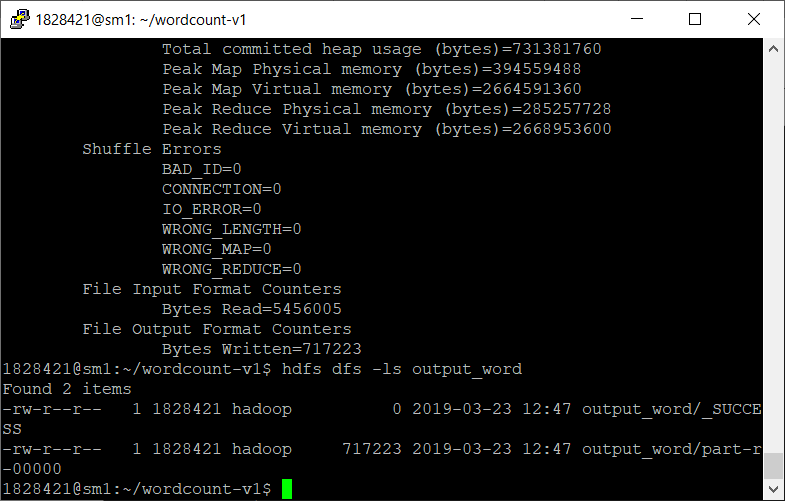
1. Deleting existing output



1. Map reduce program



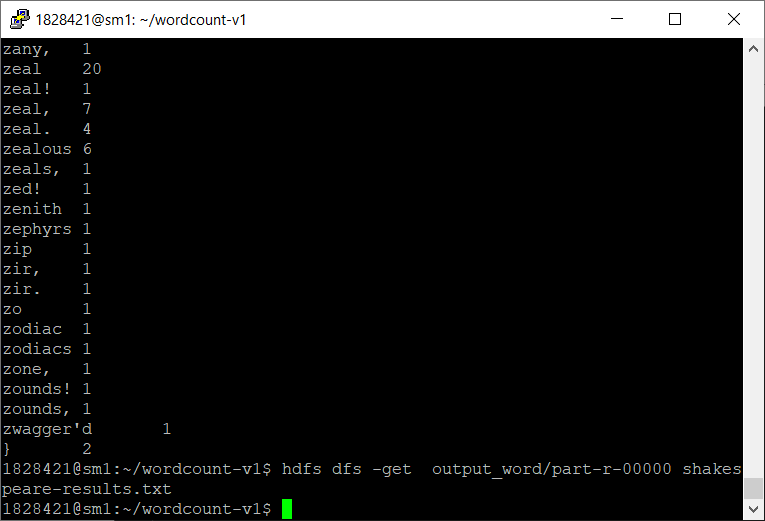
1. Checking output directory containing



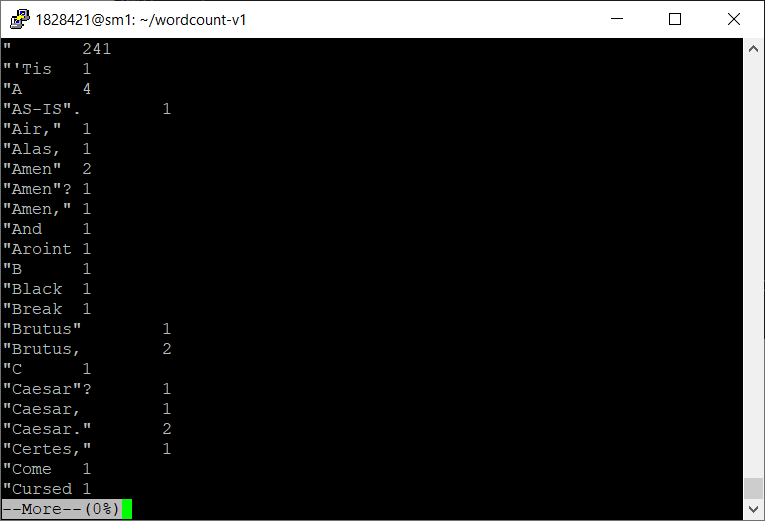
1. What is in output file



1. Coping the file locally

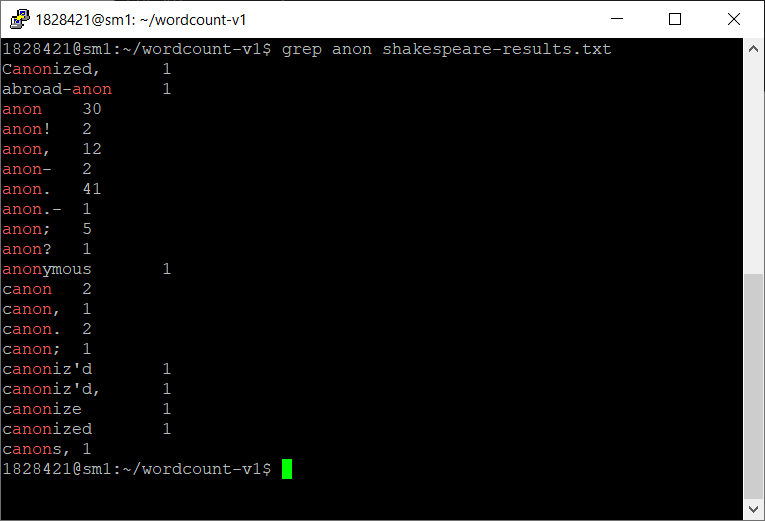


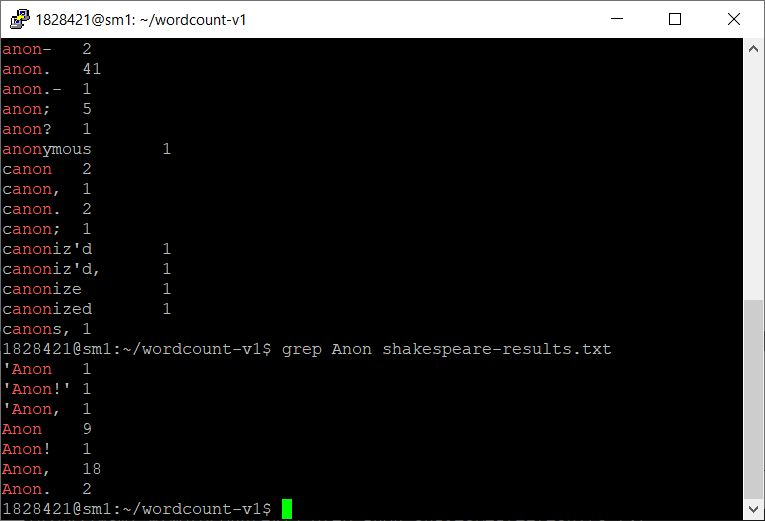
1. To view more in Linux OS:



1. Word Count Version 2

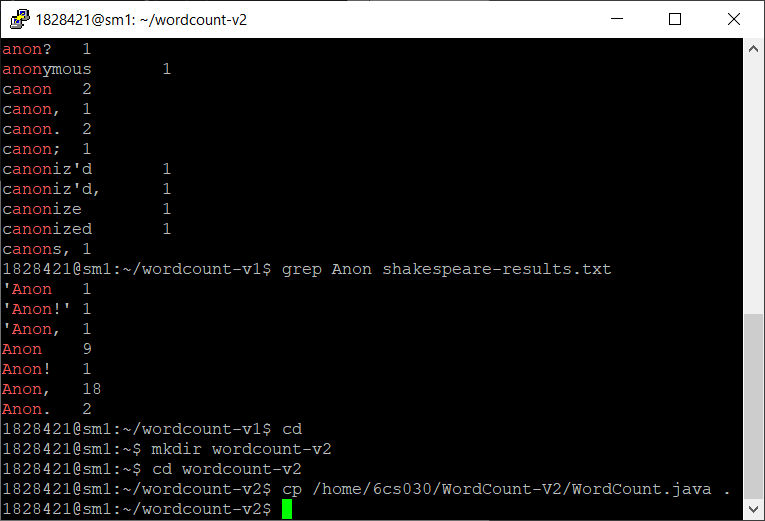
Just to see specific word count like search - grep is used.



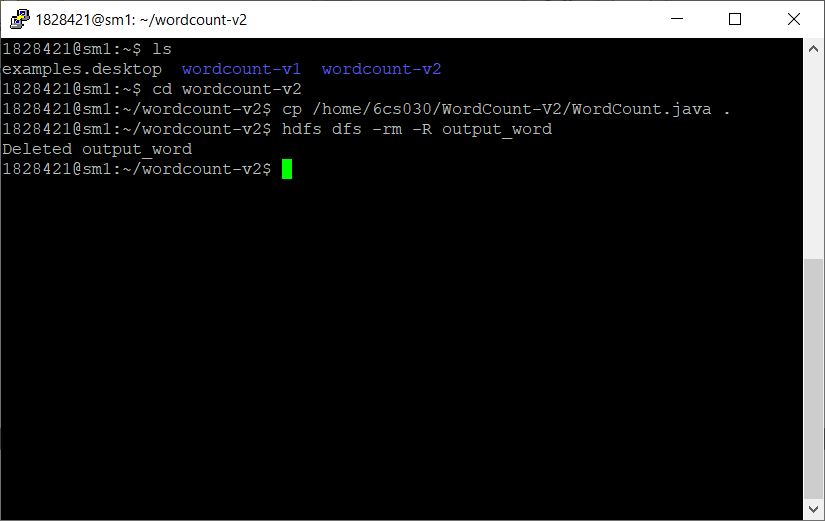


* 1. Making Word Count

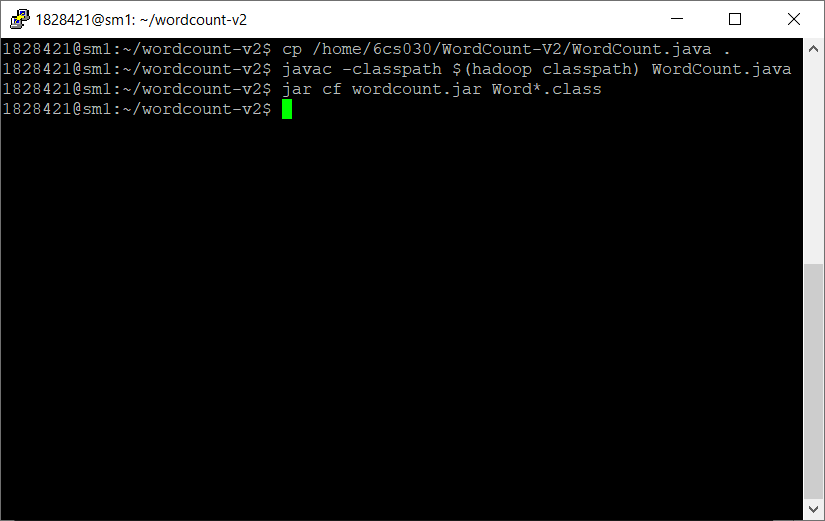
Creating new dir for word count 2 and coping file in it.

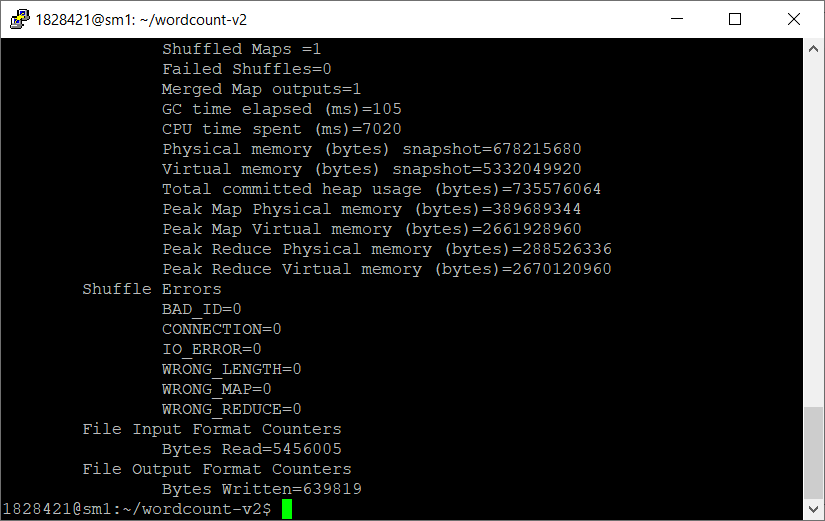


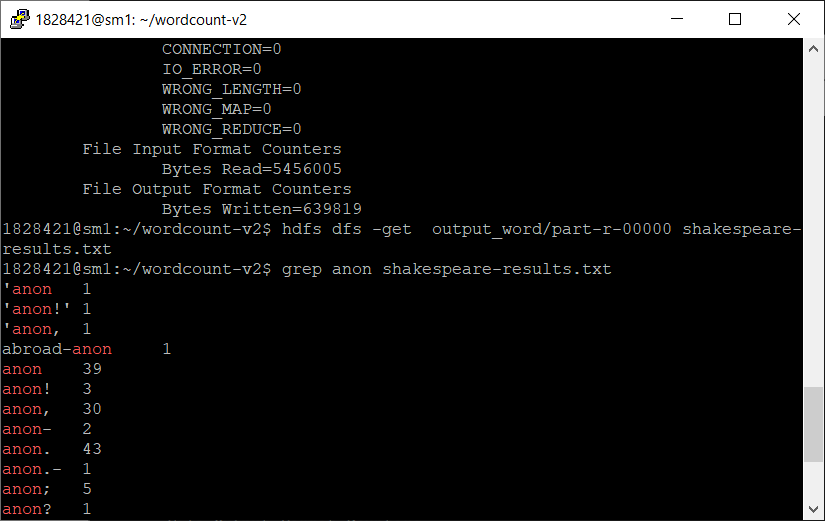
Copying wordcount file and removing output word:

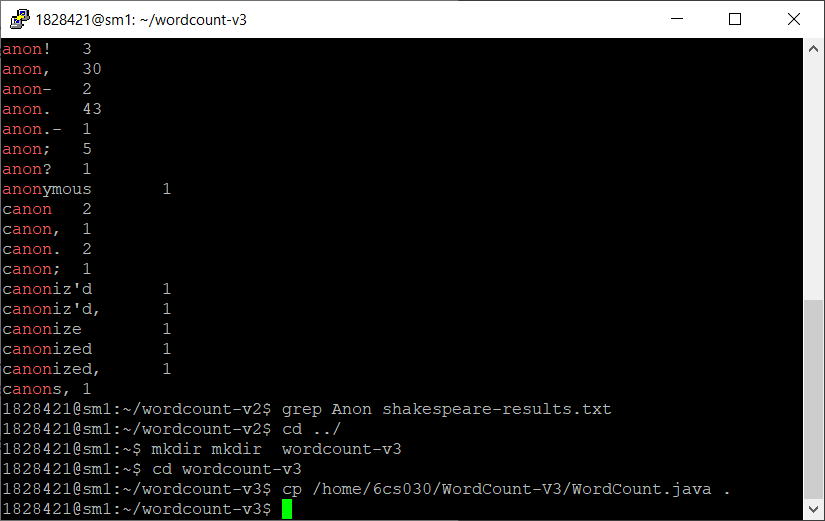


Making jar file

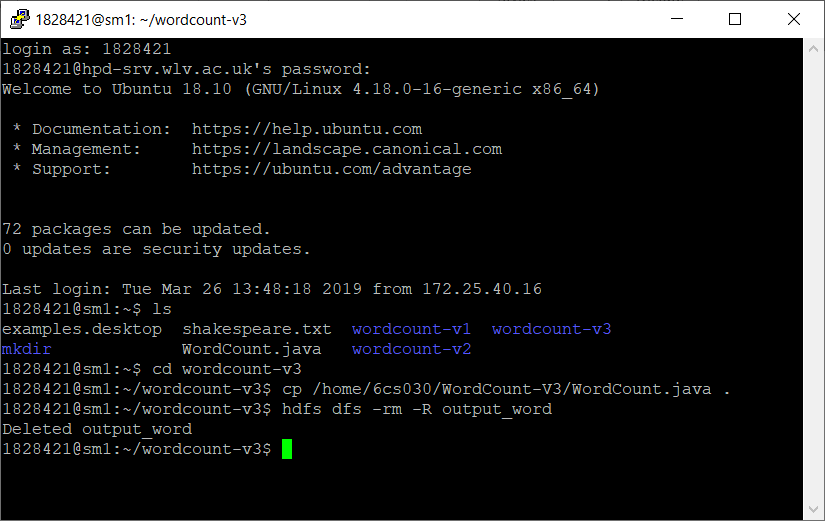


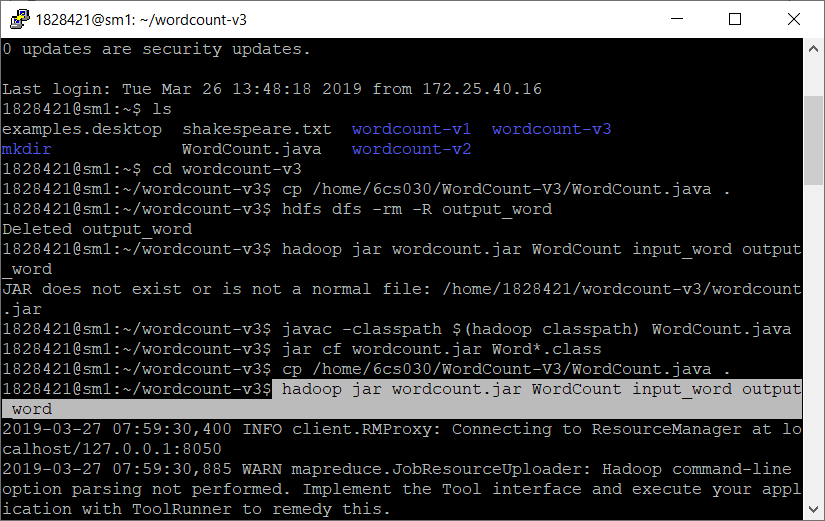


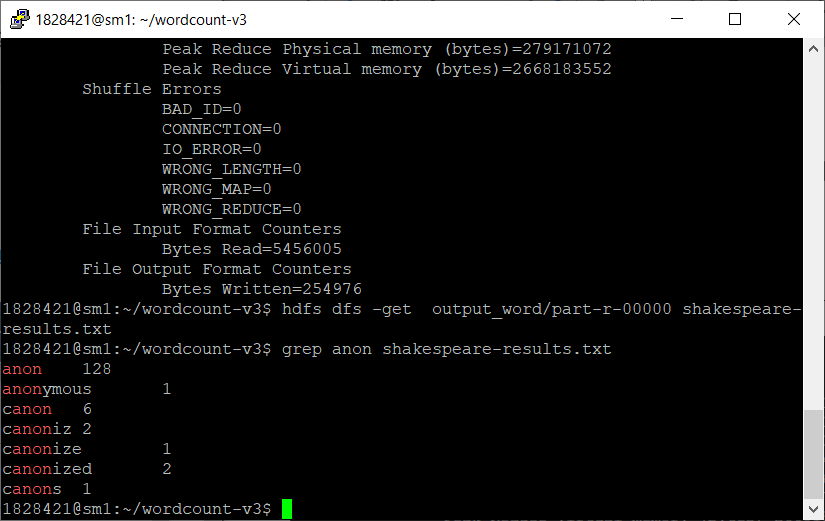




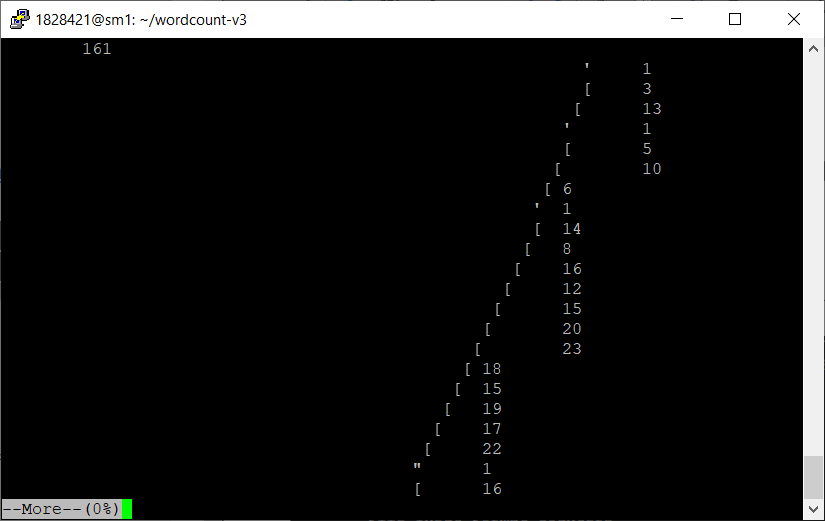
1. Word Count V3:







1. Exercise:



* 1. Sample Programs:
  2. Sudoku:
  3. Shell Script:

1. Python code: