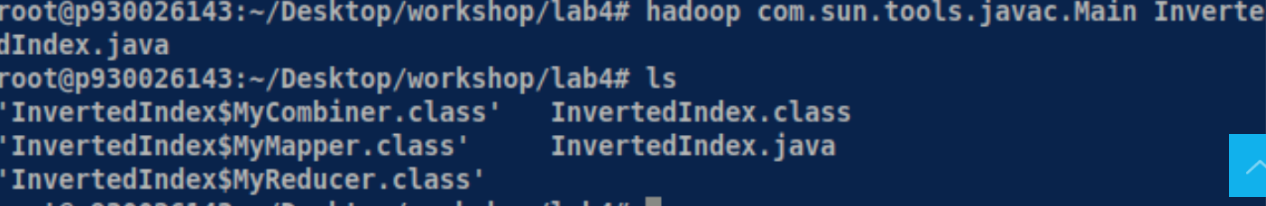
**Lab 2 Run Map Reduce Java Program on Ubuntu**

**Part 1: Compile InvertIndex.java**

Step 1: download or write a .java file in your home directory and name it “InvertedIndex.java”. Then type the common:

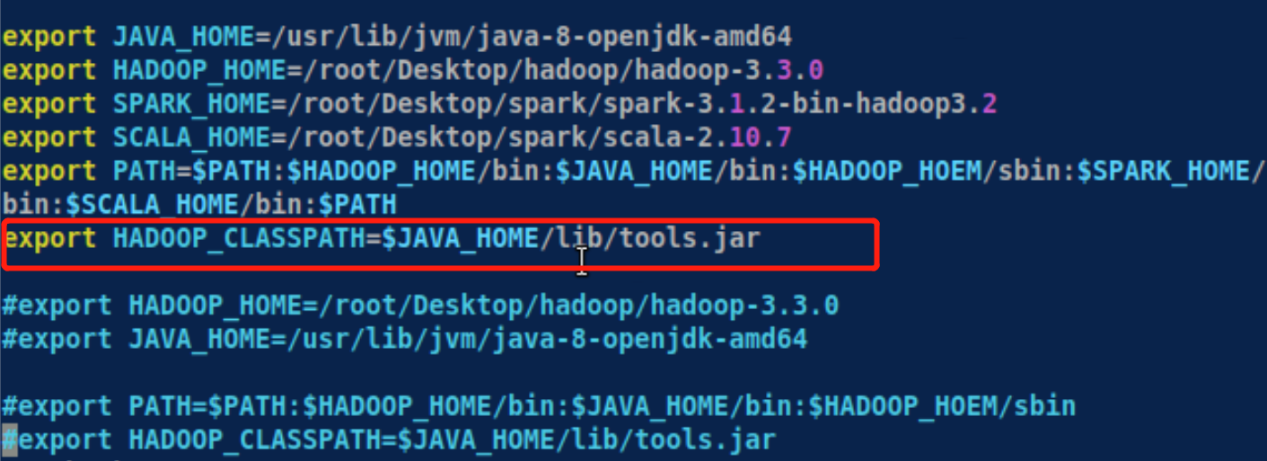


If you get an error, you should check your bashrc file in the HOME path.

Check if it is set correctly. Make sure it has the following four lines:

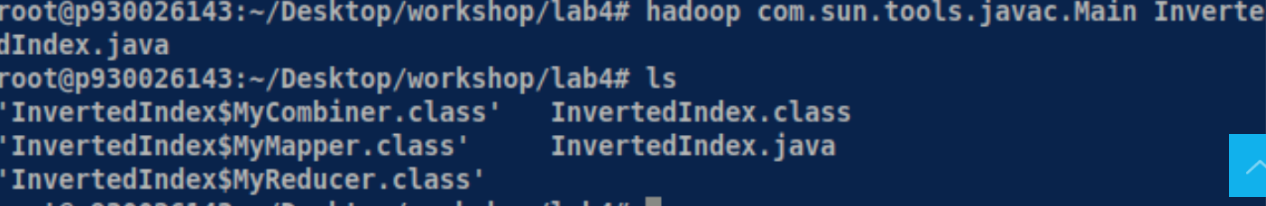
（The yellow part depends on where you install your Hadoop and java）

-3-3.0



After you finish it, you can type this common to validate the bashrc file.

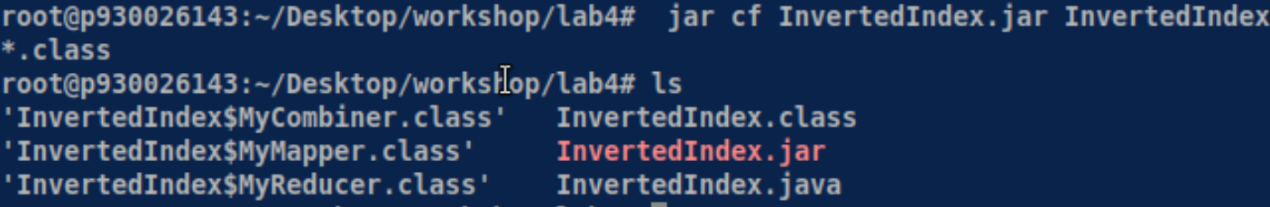
And run the common again:



You can find that there are three more .class files in the folder:

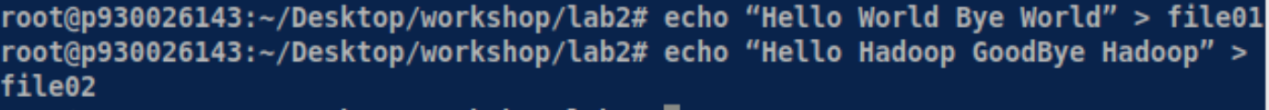
Step 2: Create a jar

Type the following command: You will find InvertedIndex.jar is created if everything is fine.



Step 3: create two text files for program.

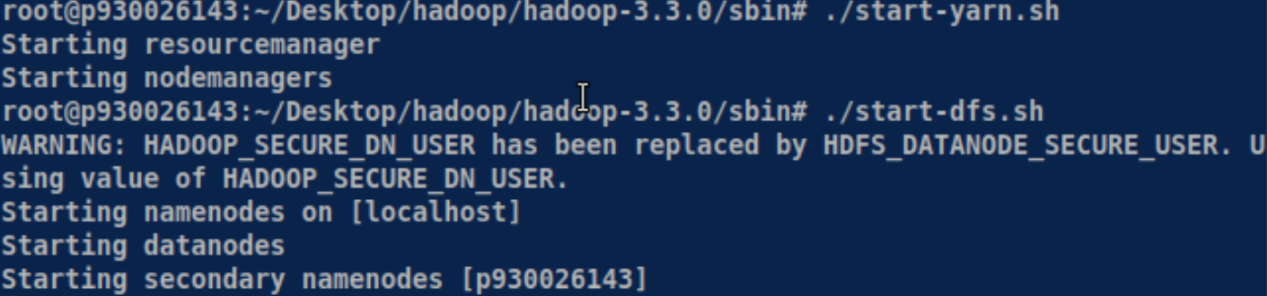
We can use Linux redirect feature to create two text files. You can also use vim to create two text files, for example:



Step 4: Start Hadoop system

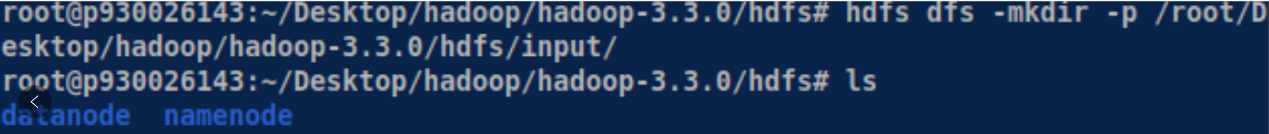
$ start-dfs.sh

$ start-yarn.sh



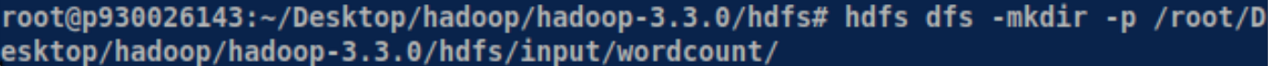
Step 5: Create an input directory

-3-3.0



Step 6: Create a subdirectory under input

----



Step 7: Upload file01 and file02 from home directory to hdfs directory:

---



It will get a long message and you should wait about half minutes and the end with the message above the picture.

(If you are unfortunate getting the error messages, you should delete your datanodes and create a new one and repeat the steps from step1 to here)

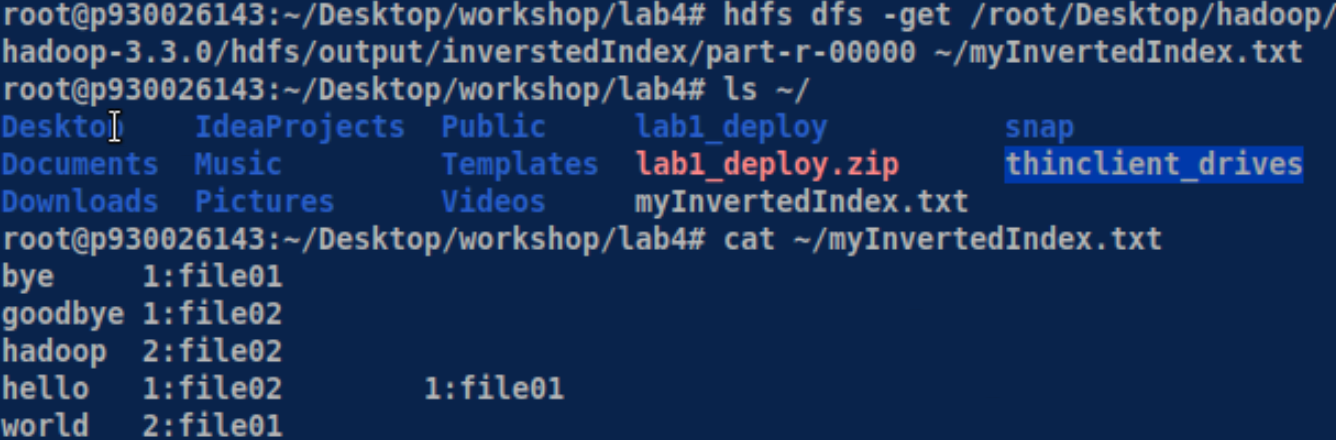
Step 8: Run the application

hadop-3-3.0 -3-3.0



And you can pull it to your local virtual machine by the common:

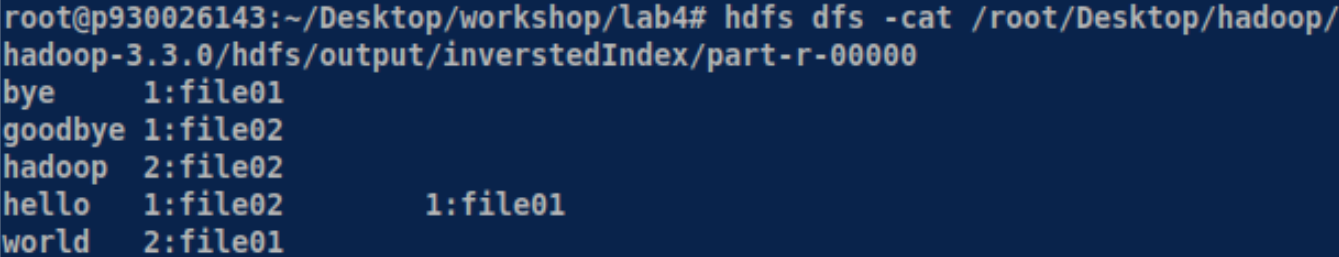
hadop-3-3.0--



You can check the result is same as the part-r-0000 file, which means it is pulled into your local machine successfully.

Step 9: Check the output result

---

**

If you can see the result, you have successfully finished part one of lab4.

Step 10: Stop Hadoop

**Part 2: Compile InvertedIndex.java**

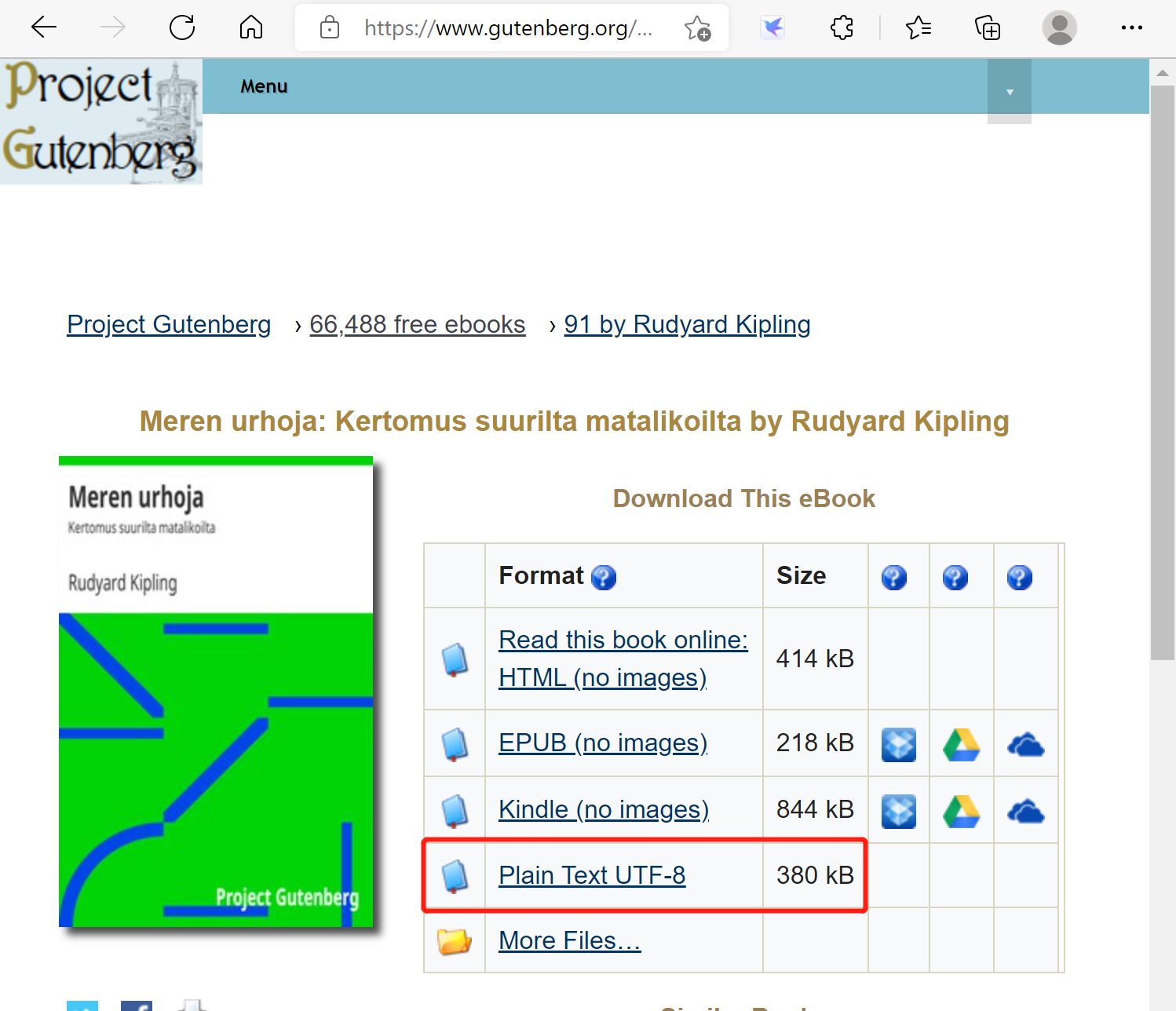
Write a Hadoop MapReduce program that change the result of the Part 1. The number of each word appears in each article should be sorted by descending order and output in the appropriate inform.

Step 1:

Download any at least 20 articles from http://www.gutenberg.org/ebooks/

(Gutenberg is a free-book database website. Please download at least 20

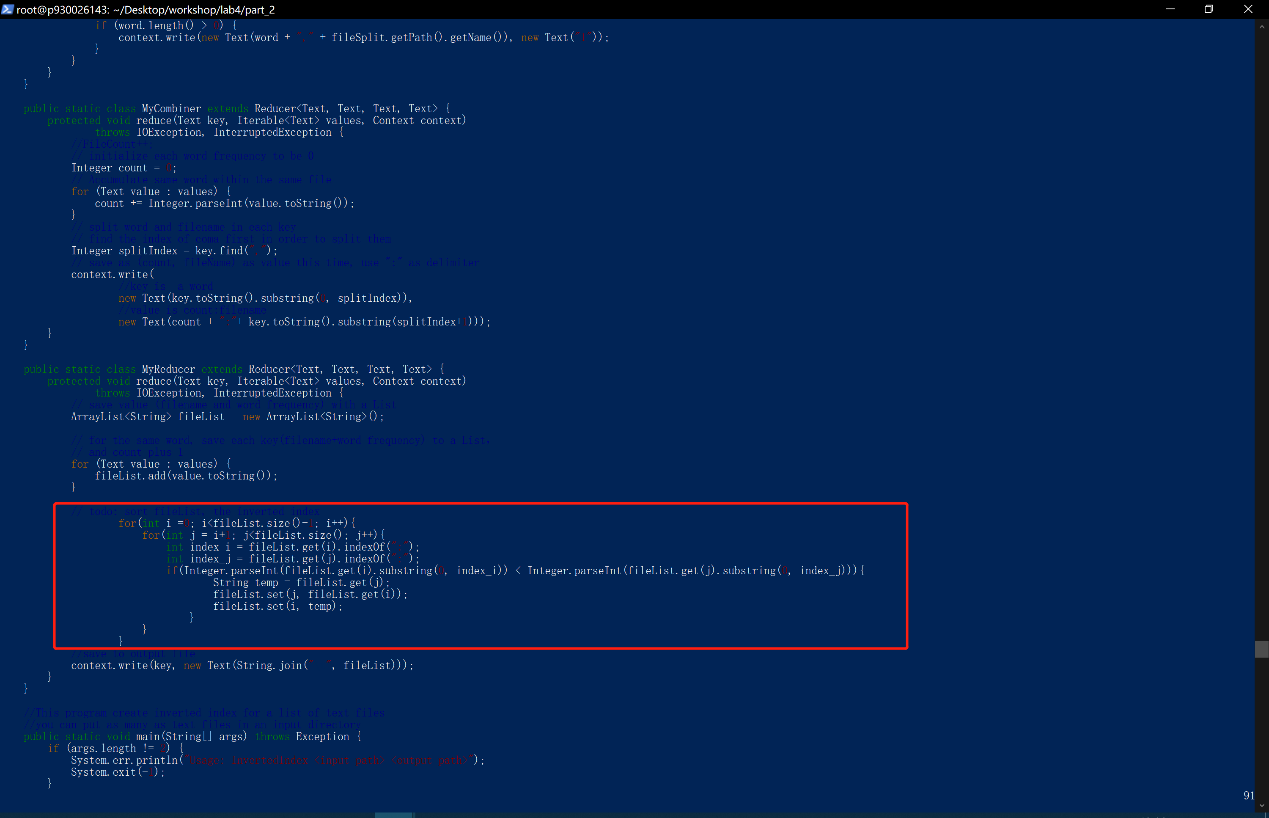
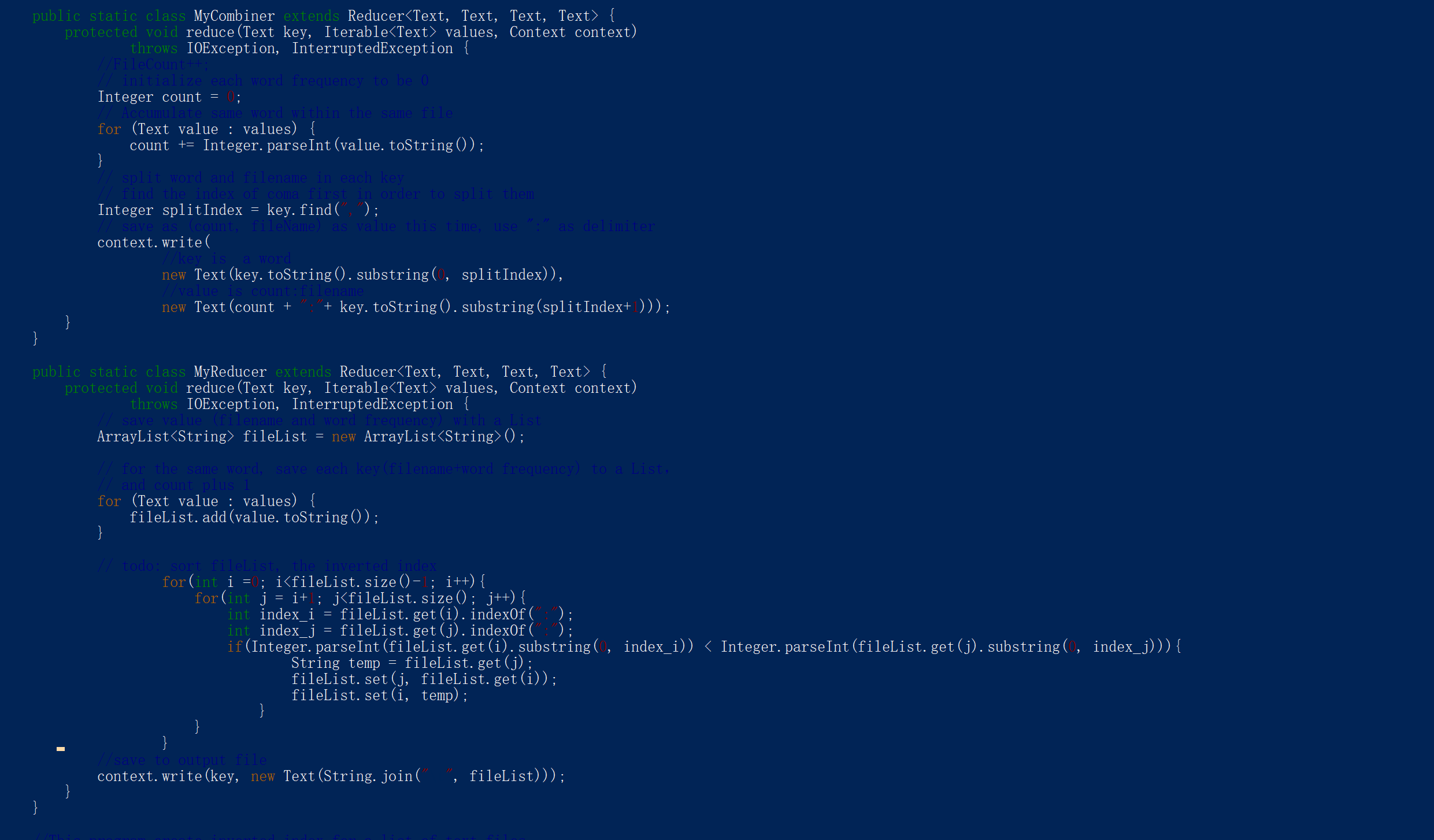
books in plain text UTF-8 format).



Here I download 20 different articles to my “workshop” fold and rename them book01, book02, …, book19, book20.

Step 2:

Then, change the .java file in your home directory from Part 1.



The code in the red box is the modification (addition) from the InvertedIndex.java in the Part 1. Just sorting the number index in the top of each array by descending order.

Then type the common:

If you get an error, you should check your file in the HOME path.

Check if it is set correctly. Make sure it has the following four lines:

（The yellow part depends on where you install your Hadoop and java）

-3-3.0

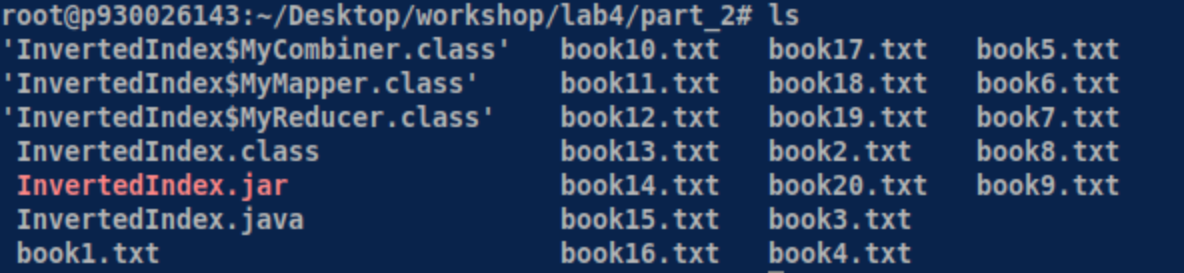
After you finish it, you can type this common to validate the file.

And run the common again:

You can find that there are three more .class files in the folder:

Step 3: Create a jar

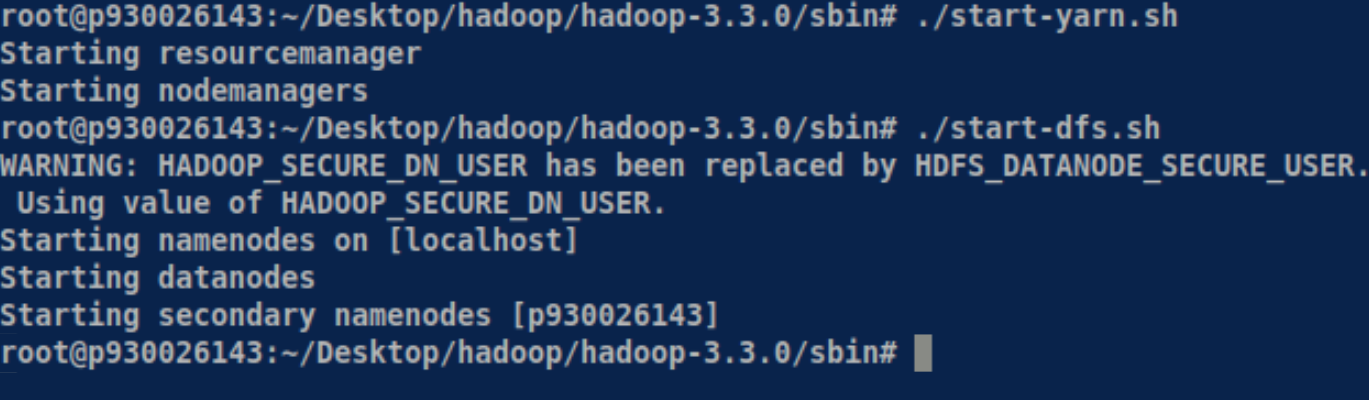
Type the following command: You will find IndexS.jar is created if everything is fine. And then just change a name and repeat the step 2 to step 9.



Step 4: Start Hadoop system

$ start-dfs.sh

$ start-yarn.sh



Step 5: Create a subdirectory under input

----

Step 6: Upload book01 to book20 from home directory to directory:

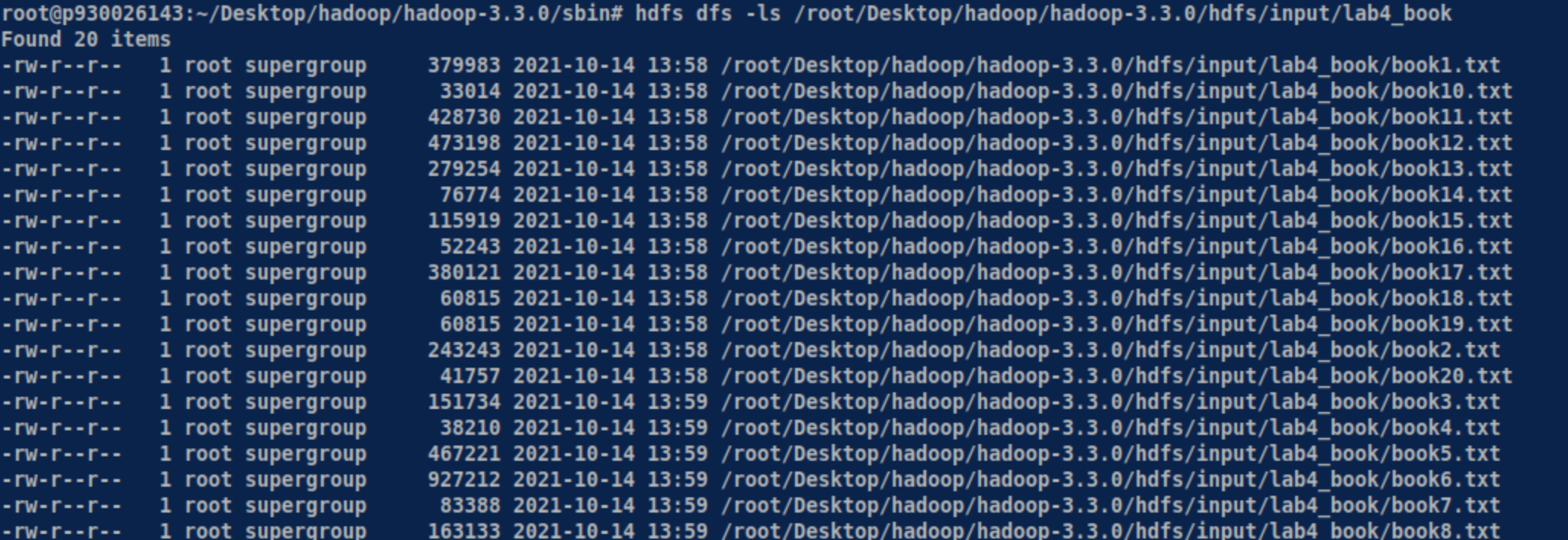
---

It will get a long message and you should wait about half minutes.

(If you are unfortunate getting the error messages, you should delete your data nodes and create a new one and repeat the steps from step1 to here)

Then you can check whether upload it successfully by this command:

---

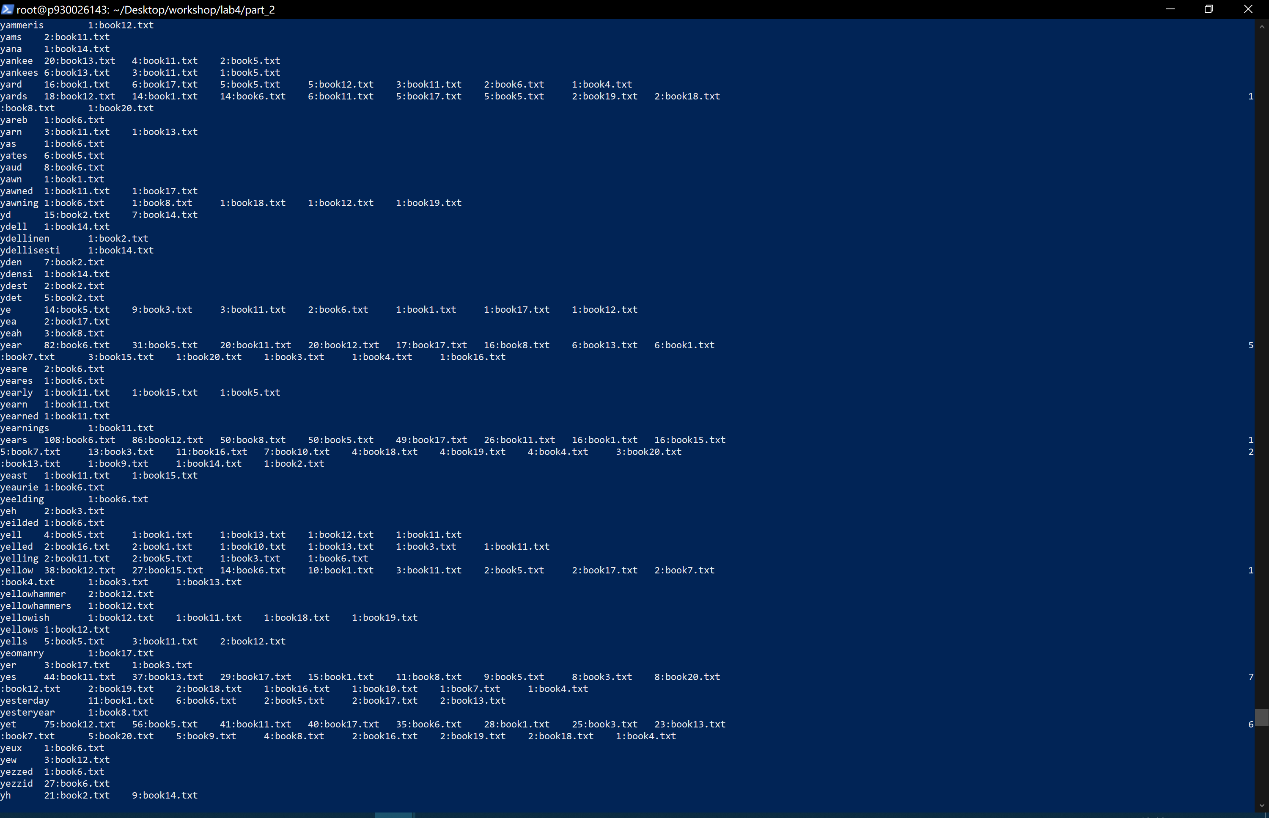


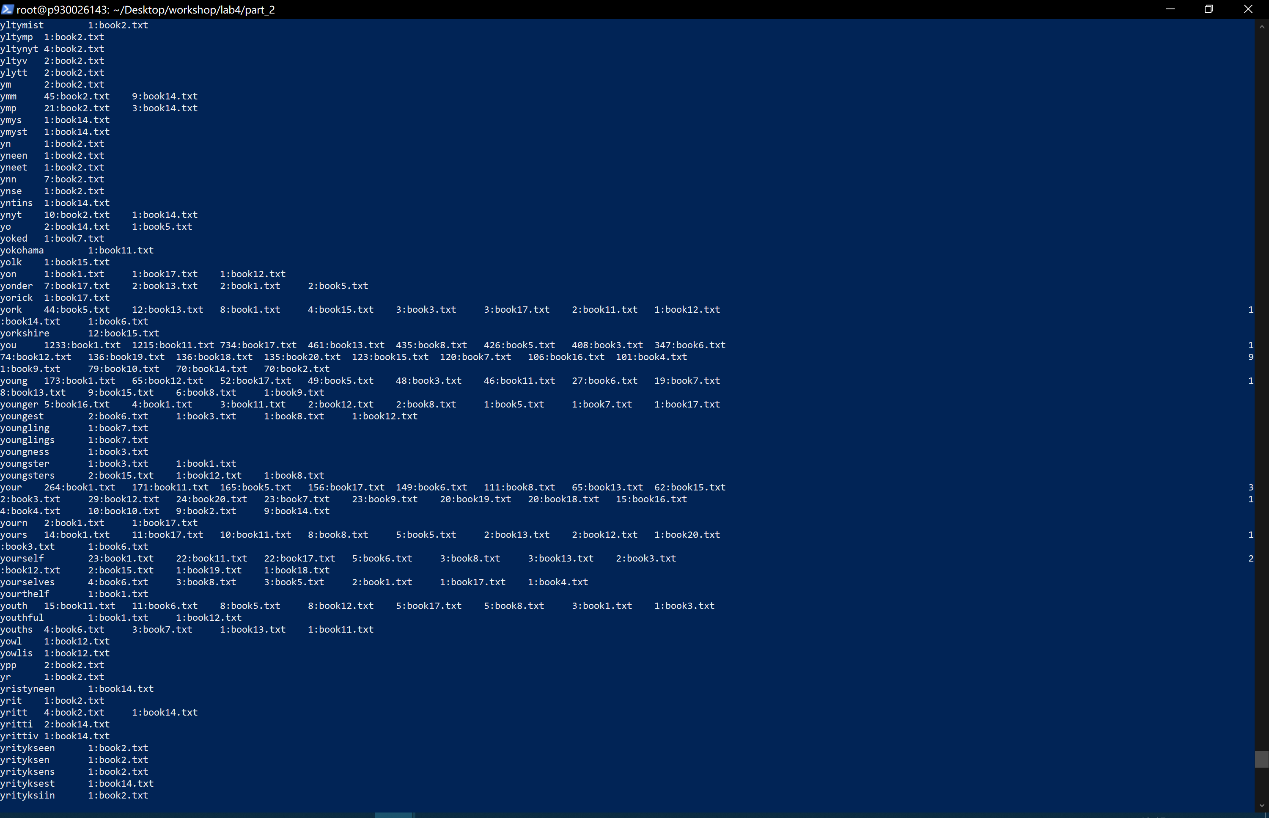
Step 7: Run the application

hadop-3-3.0 -3-3.0

Step 9: Check the output result

-----





If you can see the result, you have successfully finished part two of lab4. And you can check whether it is correct.

Step 10: Stop Hadoop