**CSC 343, Exam I Review**

## Exam I Notes

- You may bring one page of notes (front and back) to the exam. This page may be handwritten or typed.
- Computer access will not be permitted during the exam.
- Cell phones must be put away at all times.
- Don't hesitate to contact me if you have any questions!

## Exam I Outline

- Docker
    - Create a container from an image and execute a command
    - Create a new image from a container
    - Copy files to/from a container
    - Mount a directory to a container
- Linux
    - File navigation: listing files/directories, changing to a directory
    - Making a directory
    - Copying, removing, and moving files/directories
    - Viewing the contents of 1 or more files
    - The wordcount command
    - File redirection and pipes
    - Wildcards and globbing
- HDFS
    - Filestorage concepts: Blocks, Namenode vs. datanode
    - Accessing HDFS from the command line
        - Listing files
        - Making directories
        - Copying files to/from/within HDFS
        - Viewing contents of files on HDFS
- Python
    - Basic concepts – printing, *if-else* statements, *for* loops
    - Lists, strings, and slicing
- MapReduce and Hadoop MapReduce Streaming
    - Mapper input/output
    - Reducer input/output
    - Writing a mapper and reducer in Python

**Exam I Practice**

1. In a single *docker* command, create a container from the *centos* image that lists the files that are in the */tmp* directory
2. Copy the files from the /tmp directory of this container to a folder named *container* on the desktop of your computer
3. Create a container from the *centos* image such that your desktop directory is mounted to /home/desktop
4. Create a container from the *centos* image and do the following:
    a. Create the files *file1.txt*, *file2.txt*, and *file3.txt* (their content is up to you)
    b. Use globbing to list all *.txt* files
    c. Remove file3.txt
    d. Find the total number of combined lines in *file1.txt* and *file2.txt*. Your output should include *only* the combined total. For example, if there are two lines in *file1.txt* and 3 lines in *file2.txt*, your output should consist of only the number *5*.
5. Create the HDFS directory *hdfs:/user/cloudera/practice*
6. Using the command line, copy the files *file1.txt* and *file2.txt* to *hdfs:/user/cloudera/practice/*
7. Using the command line, list the files in *hdfs:/user/cloudera/practice/*
8. Using the command line, display the contents of *hdfs:/user/cloudera/practice/file1.txt*
9. Using the command line, find the number of lines in *hdfs:/user/cloudera/practice/file1.txt*
10. Will your commands in (7) – (9) work if the NameNode was running, but the DataNode was not? Why or why not? What if the DataNode was running but the NameNode was not? Note, you can start and stop the NameNode and DataNode using the commands

```
# stop namenode and datanode
service hadoop-hdfs-namenode stop
service hadoop-hdfs-datanode stop

# start namenode and datanode
service hadoop-hdfs-namenode start
service hadoop-hdfs-datanode start
```

11. Within a mapper, the code below can be used to get the name of the file being read from standard input. Note that the *if* statement is used to set *fileName* to “unknown” if Hadoop Streaming is not being used (e.g.., if testing using `cat file | python3.4 mapper.py`). Modify the *wordcount* mapper.py so that the final output of the reducer has the format *filename: number of words in file*. Note that no modifications are needed to the wordcount reducer in order to do this.

```
import os
fileName = os.getenv('mapreduce_map_input_file')

if fileName == None :
      fileName = "unknown"
```

12. Suppose that Hadoop Streaming using the MapReduce framework from (11) is executed on the three files below. A separate mapper is used for each file, and a single reducer is used.
    a. What is the output from each mapper?
    b. What is the input into the reducer
    c. What is the final output from the reducer?

| | | **Output from Mapper** |
|---|---|---|
| **First file** | **Data.txt** | |
| | Here is the data.<br>What will the mapper do? | |
| | | |
| **Second file** | **File1.txt** | |
| | This file contains words. | |
| | | |
| **Third file** | **Abstract.txt** | |
| | Abstract:<br><br>The abstract is here. | |

Input into reducer:

Output from reducer: