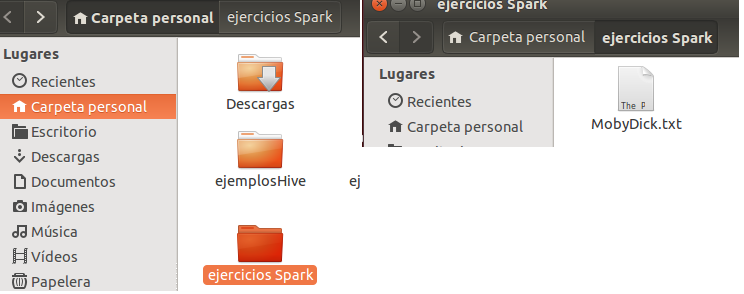
EJERCICIOS DE SPARK

*1. Recuperar el libro de Moby Dick del proyecto gutenberg*

[*http://www.gutenberg.org/cache/epub/2701/pg2701.txt*](http://www.gutenberg.org/cache/epub/2701/pg2701.txt)

Creo una carpeta para guardar el fichero descargado:



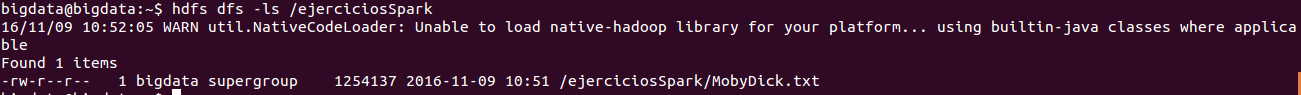
*2. Crear una carpeta ejercicioSpark en HDFS*

**

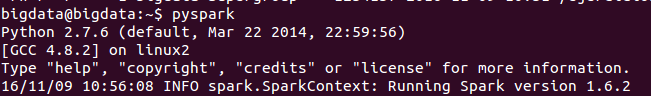
*3. Subir el fichero a HDFS a la carpeta anterior*

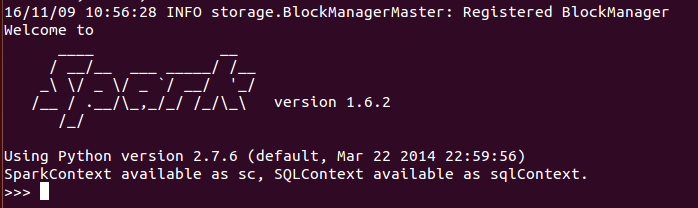


*4. Comprobar que el libro está correctamente subido*



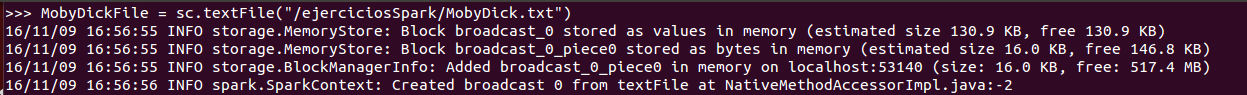
*5. Acceder a pyspark*



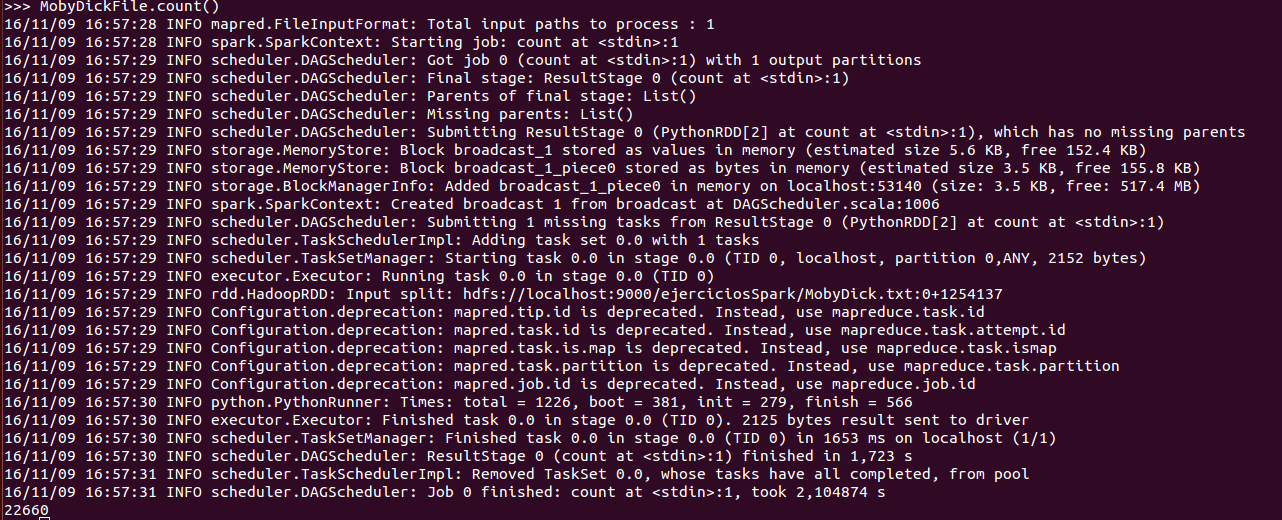


*6. Indicar los comandos y el resultado de contar el número de líneas que tiene el fichero*

Primero cargamos el fichero en un RDD referenciando el archivo hdfs:



Ejecutamos la instrucción “count()” para hacer el conteo de líneas:



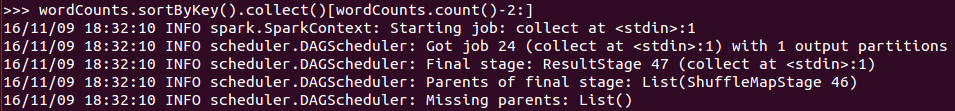
El resultado son 22660 lineas

*7. Ejecutar un word count e indicar ordenadas alfabéticamente las dos últimas palabras que aparecen y el número de repeticiones*

Guardamos las distintas palabras y el número de repeticiones en un RDD:

**

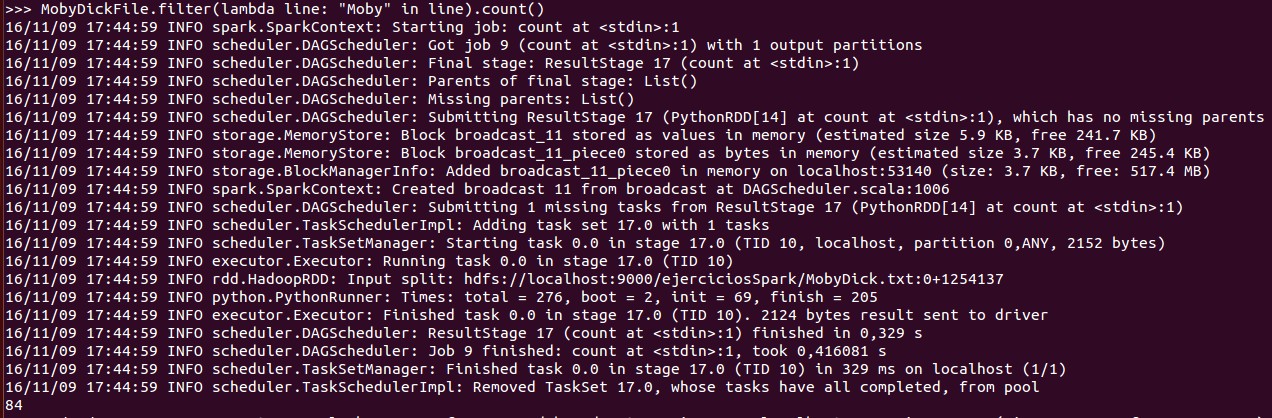
Ordenamos el RDD por la clave (la palabra) y mostramos el resultado por pantalla. Este resultado se obtiene con la función collect() que devuelve un vector. Como sólo queremos las dos últimas posiciones, lo indicamos al final:





*8. Indicar las instrucciones y el valor devuelto del número de líneas en las que aparece la palabra Moby en el libro*

Filtramos por la palabra “Moby” y hacemos un conteo:



*9. Describe con tus palabras las diferencias de usar Spark frente a Map Reduce*

La principal diferencia entre Spark y Map Reduce es que Spark se ejecuta en memoria, mientras que Map Reduce se ejecuta sobre discos. Debido a esta característica, la velocidad de procesamiento de Spark es superior a la de Map Reduce.