Top N – python

Complete Step-by-Step Tutorial to Do Word Count MapReduce Locally

---

Step 1: Create a working directory

Open your terminal and run:

```
bash
CopyEdit
mkdir wordcount
cd wordcount
```

You are now inside the `wordcount` directory.

Step 2: Create your input text file

Run:

```
bash
CopyEdit
nano input.txt
```

Paste the sample input text into `input.txt`:

```
nginx
CopyEdit
nginx

apple banana apple orange banana apple mango orange apple banana
grape mango mango banana orange orange apple
```

Save and exit (`Ctrl+O` then `Enter` to save, `Ctrl+X` to exit in nano).

Step 3: Create the mapper script (`mapper.py`)

Run:

```
bash
CopyEdit
nano mapper.py
```

Paste this code:

```
python
CopyEdit
#!/usr/bin/env python3
import sys
import re

for line in sys.stdin:
    words = re.findall(r'\b\w+\b', line.lower())
    for word in words:
        print(f"{word}\t1")
```

Save and exit (`Ctrl+O`, `Enter`, `Ctrl+X`).

Make it executable:

```
bash
CopyEdit
chmod +x mapper.py
```

Step 4: Create the reducer script (`reducer.py`)

Run:

```
bash
CopyEdit
nano reducer.py
```

Paste this code:

```
python
CopyEdit
#!/usr/bin/env python3
import sys
from collections import defaultdict

word_counts = defaultdict(int)

for line in sys.stdin:
    word, count = line.strip().split('\t')
    word_counts[word] += int(count)

sorted_words = sorted(word_counts.items(), key=lambda x: (-x[1], x[0]))
```

```
for word, count in sorted_words[:10]:
    print(f"{word}\t{count}")
```

Save and exit (`Ctrl+O`, `Enter`, `Ctrl+X`).

Make it executable:

```
bash
CopyEdit
chmod +x reducer.py
```

Step 5: Run the MapReduce simulation

Run the following command:

```
bash
CopyEdit
cat input.txt | ./mapper.py | sort | ./reducer.py
```

What it does:

- `cat input.txt` reads the input file
- `./mapper.py` maps words to `(word, 1)`
- `sort` sorts the output by word (simulate shuffle and sort phase)
- `./reducer.py` aggregates counts and prints top 10 words

Step 6: See the output

You should see something like:

```
nginx
CopyEdit
apple   6
banana  5
orange  4
mango   3
grape   1
nginx   1
```

Extra: Explanation of each file

- `input.txt` — your raw data (the text file)
- `mapper.py` — processes input lines and outputs `(word, 1)`
- `reducer.py` — sums counts for each word and prints top 10 sorted by frequency and then alphabetically

---

Summary of commands

```
bash
CopyEdit
mkdir wordcount
cd wordcount
nano input.txt          # paste input text
nano mapper.py          # paste mapper code
chmod +x mapper.py
nano reducer.py         # paste reducer code
chmod +x reducer.py
cat input.txt | ./mapper.py | sort | ./reducer.py
```