

CSCI-130 Linux Fundamentals - Week 6 (September 30, 2025) Assignment Submission

Assignment Details

1. Using the NASA web log, extract all the lines with the 'apollo' name in them.
2. Count how many lines for the apollo in the log file.
3. Count how many lines for the appolo in the log file (note misspelling)
4. Extract lines with the 'apollo' in the file name and print file field and status field (i.e. status 200,302 etc). Count how many 30x status codes are there, how many 50x status codes, and how many 40x status codes.

How to submit

Copy your relevant history entries. Add comments of what you did and what the result was. Save everything in the text file and submit it to the Canvas.

Assignment Submission Notes

Extracting lines with 'apollo'

```
cat NASA_access_log_Aug95 | awk '$7 ~ /apollo/' > apollo_lines.txt
```

Counting lines with 'apollo'

```
cat apollo_lines.txt | wc -l
```

Output:

```
162205
```

Using **awk** directly:

```
cat NASA_access_log_Aug95 | awk '$7 ~ /apollo/' | wc -l
```

Output:

```
162205
```

Counting lines with 'appolo' (misspelled)

```
cat NASA_access_log_Aug95 | awk '$7 ~ /appolo/' | wc -l
```

Output:

```
9
```

Extracting file and status fields for 'apollo' and counting status codes

```
cat NASA_access_log_Aug95 | awk '$7 ~ /apollo/ {print $7, $9}' > apollo_status.txt
```

Counting 30x status codes

```
cat NASA_access_log_Aug95 | awk '$7 ~ /apollo/ && $9 ~ /^30[0-9]$/ {print $7, $9}' | wc -l > apollo_30x_count.txt
```

Output:

```
9959
```

Counting 40x status codes

```
cat NASA_access_log_Aug95 | awk '$7 ~ /apollo/ && $9 ~ /^40[0-9]$/ {print $7, $9}' | wc -l > apollo_40x_count.txt
```

Output:

```
1698
```

Counting 50x status codes

```
cat NASA_access_log_Aug95 | awk '$7 ~ /apollo/ && $9 ~ /^50[0-9]$/ {print $7, $9}' | wc -l > apollo_50x_count.txt
```

Output:

```
2
```

Conclusion

This assignment involved using `awk` for pattern matching and field extraction from the NASA web log file. We successfully extracted lines containing 'apollo', counted occurrences, and analyzed HTTP status codes associated with those requests. The use of regular expressions in `awk` allowed for efficient filtering based on specific criteria. While `grep` could also be used for simple pattern matching, `awk` provided the flexibility needed for more complex field-based operations.