

```
es.desktop  lena_rotate.bmp  me.txt  op.c      Public  rc_c  rc_s  
mp          matchanal.py    Music   Pictures  qw.c   rc_hw  rc_t  
  
.jpg  datasets  lena.bmp      salt_n_pepper.bmp  se05  
lena1.bmp  saltandpeppernoise.jpeg  se04  xampp  
  
untwords.cpp  d2b_part2.cpp  homework_ds      possible_combin  
b.cpp        gp.cpp       largest_island.cpp  reverse_words_in  
  
h1.cpp  
h2.cpp  
  
p  
pp  
ementtriangle.cpp  
words.cpp  
pp  
art2.cpp  
p  
ork_ds  
st_island.cpp  
ble_combinations_backtracking.cpp  
se_words_in_sentence.cpp  
g.cpp  
g.cpp  
rray.cpp  
gle.cpp
```

Discovering the Magic of the Awk Command

If you're unfamiliar with Awk, it's a versatile command-line tool that can greatly simplify a variety of tasks. Keep reading to learn more!



The Flexible World of Awk

1

Powerful

Awk provides a simple and effective way to parse text and perform various tasks on the command line.

2

Easy to Learn

Despite its power, Awk is easy to learn, with concise syntax and plenty of online resources available.

3

Universal

Awk is available on practically all Unix-based systems, making it a useful tool to know how to use in any environment.



Getting Started with Awk

Syntax Basics

Awk programs are composed of patterns and actions. A pattern specifies which lines to process, and an action specifies what to do with them.

Variables

Awk provides several built-in variables that can be used in patterns and actions, such as \$0 (the current line) and NF (the number of fields in the current line).

File Input and Output

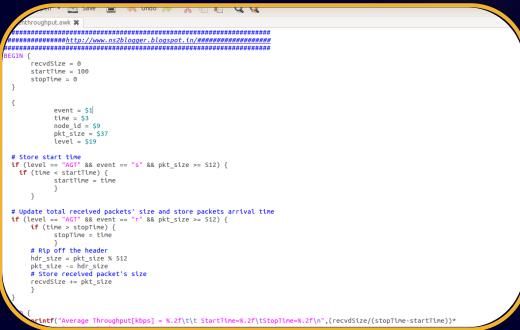
You can use awk to process files or to read input from the standard input stream. Output can also be redirected to a file or sent to another command through a pipe.

Advanced Usage

Awk can be used to perform more advanced tasks, such as text manipulation, calculations, and more, by using regular expressions and other advanced features.



Unlocking the Power of Awk



```
groupout.awk
#!/usr/bin/awk -f
# Set initial values
recvSize = 0
start = 0
end = 0
stopTime = 0
mode = "idle"
event = "idle"
pkt_size = 537
level = 559
# Store start time
if (level == "559" && event == "x" && pkt_size == 512) {
    if (time != startime) {
        startime = time
    }
}
# Update total received packets' size and store packets arrival time
if (level == "559" & event == "y" && pkt_size == 512) {
    if (time != stopTime) {
        stopTime = time
    }
}
# Rtp off the header
hdr_size = pkt_size - 512
pkt_size = pkt_size - hdr_size
# Store received packet's size
recvSize += pkt_size
# Calculate Average Throughput[Mbps]
print("Average Throughput[kbps] = " . (2 * (time - startime) / (stopTime - startime)) * (recvSize / (stopTime - startime)))
```

Text Manipulation

Use Awk to transform data easily, such as converting CSV files to XML or extracting specific fields from a log file.



Filtering Data

Awk can be used in combination with other tools like grep to filter text and find specific patterns quickly.



Pattern Matching with Regular Expressions

Awk provides powerful pattern matching functionality through regular expressions, opening up a wide range of possible use cases.



Made with Gamma



Examples of Awk in Action

① Summing Columns of Numbers

By using Awk, you can sum the values in a specific column of a CSV file easily.

② Filtering Log Files

Awk can be used alongside other tools like grep to filter log files and generate reports on errors or other interesting events.

③ Extracting Fields from Text Files

With its ability to work with regular expressions and text fields, Awk can be used to extract specific fields from a file.



Awk and Beyond

Advanced Topics

Awk can also be used to perform more advanced tasks, like processing binary files, working with databases, and even creating full-blown scripts.

1

Other Text Processing Tools

If you enjoyed using Awk, there are other tools to learn, like sed and grep, which can greatly expand your text-processing toolset.

2

Online Resources

There are countless Awk tutorials, guides, and online communities that can help you get even more out of this amazing command-line tool.

3

Final Thoughts

Awk is Powerful

Awk is a great tool for text processing on the command line, and can greatly simplify even complicated tasks.

But it's not perfect

While Awk can be very powerful, it won't always be the best tool for the job. Other command-line tools and programming languages may be better suited for certain tasks.

Have fun learning!

Whether you're new to Awk or are already an experienced user, there's always more to learn. So have fun, experiment, and use this amazing tool to unlock new insights from your data!

