

AWK COMMAND CHEAT SHEET

Definition:

‘awk’ is a powerful text processing tool used for data extraction and manipulation. It operates per line, allowing you to specify patterns and actions on lines matching those patterns.

Basic Syntax:

\$ awk options 'selection_criteria {action }' input-file > output-file

AWK Command Usefulness

- Changing data file
- Producing formatted reports

Common awk Variables:

- \$0:** Used to specify the whole line
- \$1:** Specifies the first field
- \$2:** Specifies the second field

Examples:

Printing specific columns
awk '{print \$2 "\t" \$3}' file.txt

Printing all lines in a file
awk '{print \$0}' file.txt

Printing columns that match a specific pattern
awk '/a/ {print \$3 "\t" \$4}' file.txt

Saving output of ‘awk’ to a different file
awk '/a/ {print \$3 "\t" \$4}' file.txt > Output.txt

Common awk Options:

Options	Description
-F [separator]	Used to specify a file separator. The default separator is a blank space.
-f [filename]	Used to specify the file containing the awk script. Reads the awk program source from the specified file, instead of the first command-line argument.
-v	Used to assign a variable.

Program:

BEGIN {<initializations>}
<pattern 1> {<program actions>}
<pattern 2> {<program actions>}
...
END {< final actions >}

Common Functions:

index(s,t)
length(s)
rand
substr(s,index,len)
srand
int(x)
split(s,a,fs)
match(s,r)
sub(r,t,s)
gsub(r,t,s)
system(cmd)
tolower(s)
toupper(s)
getline

Operators:

Arithmetic	Shorthand	Comparison
+	+=	==
-	-=	!=
*	*=	<
/	/=	>
%	%=	<=
++		>=
--		

Regular expression

/regex/	Line matches
!/regex/	Line not matches
\$1 ~ /regex/	Field matches
\$1 !~ /regex/	Field not matches

Built-in awk Variables:

NR: Number of Records
NF: Number of Fields
OFS: Output Field Separator
FS: Input Field Separator
ORS: Output Record Separator
RS: Input Record Separator
FILENAME: Name of the file

Common Specifiers:

c	ASCII character
d	Decimal integer
e,E,F	Floating-point format
o	Unsigned octal value
s	String
%	Literal %

Escape Sequences:

\b	Backspace
\f	Form feed
\n	Newline
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab

Condition:

awk -F: '\$3>30 {print \$1}' /etc/passwd

Regex Metacharacters:

\ ^ \$. [] | () * + ?