Getting the best out of “awk”

Manodeep Sinha
Swinburne University

What will you learn?

- When you should use awk

What will you learn?

- When you should use awk
- When you should **not** use awk

What will you learn?

- When you should use awk
- When you should not use awk
- How to use awk

What is awk?

What is awk?

"man awk" — pattern-directed scanning and processing language

What is awk?

man awk — pattern-directed scanning and processing language

What is awk?

man awk — pattern-directed scanning and processing language

awk is a tool for processing text files

What is awk?

man awk — pattern-directed scanning and processing language

awk is a tool for processing text files

When should you use awk?

- If you need to select certain column data from text files

When should you use awk?

- If you need to select certain column data from text files
- If you need to perform basic math on column data

When should you use awk?

- If you need to select certain column data from text files
- If you need to perform basic math on column data
- If you need to change the formatting (e.g., tab-separated to comma-separated)

When should you use awk?

- If you need to select certain column data from text files
- If you need to perform basic math on column data
- If you need to change the formatting (e.g., tab-separated to comma-separated)
- If you need to count some specific occurrence of “pattern” within entire file

When should you **not** use awk?

When should you **not** use awk?

- If you need to replace text in-place within file (use “sed”)

When should you **not** use awk?

- If you need to replace text in-place within file (use “sed”)
- If your data are not in columns

When should you **not** use awk?

Why awk?

"The Enlightened Ones say that....

- *You should never use **C** if you can do it with a **script**;*
- *You should never use a script if you can do it with **awk**;*
- *Never use **awk** if you can do it with **sed**;*
- *Never use **sed** if you can do it with **grep**."*

The language is simple and Awk programs are generally very short.
Awk is useful when the overhead of more sophisticated approaches is not worth the bother.

source (*some years ago*): http://awk.info/?whygawk

Inspirations from this tutorial

https://linuxhandbook.com/awk-command-tutorial/

How to use awk?

How to use awk?

- awk ‘<pattern> <action, if pattern is True>’ <filename>

How to use awk?

- awk '<pattern> <action, if pattern is True>' <filename>
- Works on each line of the input

How to use awk?

- awk '<pattern> <action, if pattern is True>' <filename>
- Works on each line of the input
- The default action is: print entire line

Handy variables for awk

Handy variables for awk

- NR is the line number being processed

Handy variables for awk

- NR is the line number being processed
- NF is the total number of columns

Handy variables for awk

- NR is the line number being processed
- NF is the total number of columns
- $0 is entire line, $1 is first column, $2 is second…

Handy variables for awk

- NR is the line number being processed
- NF is the total number of columns
- $0 is entire line, $1 is first column, $2 is second…
- **What is $NF?**

Handy variables for awk

- NR is the line number being processed
- NF is the total number of columns
- $0 is entire line, $1 is first column, $2 is second…
- What is $NF?

https://raw.githubusercontent.com/swincas/cookies-n-code/master/tutorials/regular_expressions/messier_objects.txt

Simplest use (awk '<pattern> <action, if pattern is True>')

Simplest use (awk '<pattern> <action, if pattern is True>')

- What will: awk '1 { print $0 }' <filename> do?

Simplest use (awk '<pattern> <action, if pattern is True>')

- What will: awk '1 { print $0 }' <filename> do?
- How about: awk '1' <filename>?

Simplest use (awk '<pattern> <action, if pattern is True>')

- What will: awk '1 { print $0 }' <filename> do?
- How about: awk '1' <filename>?
- Any other tool that already performs this?

Simplest use (awk '<pattern> <action, if pattern is True>')

- What will: awk '1 { print $0 }' <filename> do?
- How about: awk '1' <filename>?
- Any other tool that already performs this?
- awk '0' <filename>?

Normal use (awk '<pattern> <action, if pattern is True>')

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 'NR > 1 { print }' <filename>?

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 'NR > 1 { print }' <filename>?

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 'NR > 1 { print }' <filename>?
- awk 'NR > 1 && NR < 3 { print }' <filename>?
- awk 'NR > 1 && NR < 3' <filename>?

Normal use (awk '<pattern> <action, if pattern is True>')

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 'NF { count+=1 } END { print count }' <filename>?

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 'NF { count+=1 } END { print count }' <filename>?
- Any other tool that does that?

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 'NF { count+=1 } END { print count }' <filename>?
- Any other tool that does that?

- awk 'BEGIN { sum = 0} NF { sum += $2 } END { print sum }'

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 'NF { count+=1 } END { print count }' <filename>?
- Any other tool that does that?
- awk 'BEGIN { sum = 0} NF { sum += $2 } END { print sum }'
- awk 'array[$0]++' <file>?

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 'NF { count+=1 } END { print count }' <filename>?
- Any other tool that does that?
- awk 'BEGIN { sum = 0} NF { sum += $2 } END { print sum }'
- awk 'array[$0]++' <file>?
- awk 'array[$0]++' <file>?

Normal use (awk '<pattern> <action, if pattern is True>')

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 1 '{ print 3e5*$1/70.0 }' <filename>?

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 1 '{ print 3e5*$1/70.0 }' <filename>?

- awk 1 '{ print 1/(1 + $1) }' <filename>?

Normal use (awk '<pattern> <action, if pattern is True>')

- awk 1 '{ print 3e5*$1/70.0 }' <filename>?
- awk 1 '{ print 1/(1 + $1) }' <filename>?
- awk 1 '{ print 1/$1 - 1.0 }' <filename>?

Customising awk

Customising awk

- awk ‘<pattern> <action, if pattern is True>’ FS=‘’ OFS=‘’ <filename>

Customising awk

- awk ‘<pattern> <action, if pattern is True>’ FS=‘’ OFS=‘’ <filename>
- “FS” (field separator) — is “whitespace” by default

Customising awk

- awk ‘<pattern> <action, if pattern is True>’ FS=‘’ OFS=‘’ <filename>
- “FS” (field separator) — is “whitespace” by default
- “OFS” (output FS) — same default as FS

Customising awk

- awk ‘<pattern> <action, if pattern is True>’ FS=‘’ OFS=‘’ <filename>
- “FS” (field separator) — is “whitespace” by default
- “OFS” (output FS) — same default as FS
- **How would you parse “messier_objects.txt”?**

awk ‘<pattern> <action, if pattern is True>’ FS=‘’ OFS=‘’ <filename>

awk ‘<pattern> <action, if pattern is True>’ FS=‘’ OFS=‘’ <filename>

- awk 1 FS='|' OFS=' ** ' messier_objects.txt?

awk ‘<pattern> <action, if pattern is True>’ FS=‘’ OFS=‘’ <filename>

- awk 1 FS='|' OFS=' ** ' messier_objects.txt?
- awk ‘{ print $1, $2 }’ FS='|' OFS='**' messier_objects.txt?

awk ‘<pattern> <action, if pattern is True>’ FS=‘’ OFS=‘’ <filename>

- awk 1 FS='|' OFS=' ** ' messier_objects.txt?
- awk ‘{ print $1, $2 }’ FS='|' OFS='**' messier_objects.txt?
- awk ‘{ print $1, $NF }’ FS='|' OFS='**' messier_objects.txt?

awk ‘<pattern> <action, if pattern is True>’ FS=‘’ OFS=‘’ <filename>

- awk 1 FS='|' OFS=' ** ' messier_objects.txt?
- awk ‘{ print $1, $2 }’ FS='|' OFS='**' messier_objects.txt?
- awk ‘{ print $1, $NF }’ FS='|' OFS='**' messier_objects.txt?
- awk '$1=$1' FS=, OFS=';' <filename>

Custom printing with awk

Custom printing with awk

- C-style printf function

Custom printing with awk

- C-style printf function

- awk '+$1 { printf("%10s | %04d\n", $5, $2) }' FS=, <file>?

Inspirations from this tutorial

https://linuxhandbook.com/awk-command-tutorial/

Conclusions on awk

- When you should use awk

Conclusions on awk

- When you should use awk
- When you should **not** use awk

Conclusions on awk

- When you should use awk
- When you should not use awk
- How to use awk