

## A quick awk tour

Michael Doob  
Department of Mathematics  
University of Manitoba  
[mdoob@ccu.umanitoba.ca](mailto:mdoob@ccu.umanitoba.ca)

Using awk, nawk, gawk  
input line by line → AWK → output

Using awk, nawk, gawk  
input line by line → AWK → output

AWK: `awk [-f -F -v] 'AWK-COMMAND' FILE`

- ▶ `-f` define scriptfile
- ▶ `-F[char]`: define field separator
- ▶ `-v var=val`: predefine variable var

Using awk, nawk, gawk  
input line by line → AWK → output

AWK: `awk [-f -F -v] 'AWK-COMMAND' FILE`

- ▶ `-f` define scriptfile
- ▶ `-F[char]`: define field separator
- ▶ `-v var=val`: predefine variable `var`

AWK-COMMAND: `/pattern/ { awk commands }`

- ▶ `/pattern/`: apply if pattern matches input (may be empty)
- ▶ `awk commands`: (may be empty)

[FILE]: apply to file(s) instead of input stream

### Initial processing

- ▶ input line set to \$0
- ▶ NF set to number of fields
- ▶ NR set to record number
- ▶ fields set to \$1,\$2, ..., \$NF

### Awk from the command line

- ▶ print first two fields in reverse order  
`awk '{print $2 $1}'`

### Awk from the command line

- ▶ print first two fields in reverse order  
`awk '{print $2 $1}'`
- ▶ words per line (NF: number of fields)  
`awk '{print NF}'`

### Awk from the command line

- ▶ print first two fields in reverse order  
`awk '{print $2 $1}'`
- ▶ words per line (NF: number of fields)  
`awk '{print NF}'`
- ▶ number of lines (NR: number of records)  
`awk 'END {print NR}'`



### Awk from the command line

- ▶ print first two fields in reverse order  
`awk '{print $2 $1}'`
- ▶ words per line (NF: number of fields)  
`awk '{print NF}'`
- ▶ number of lines (NR: number of records)  
`awk 'END {print NR}'`
- ▶ frequency of words per line  
`awk '{print NF}' | sort -n | uniq -c`

### Awk from the command line

- ▶ print first two fields in reverse order  
`awk '{print $2 $1}'`
- ▶ words per line (NF: number of fields)  
`awk '{print NF}'`
- ▶ number of lines (NR: number of records)  
`awk 'END {print NR}'`
- ▶ frequency of words per line  
`awk '{print NF}' | sort -n | uniq -c`
- ▶ Find users using bash as login shell  
`awk -F: '$7 /bash/ {print $1,$7}' /etc/passwd`

### Awk from the command line

- ▶ print first two fields in reverse order  
`awk '{print $2 $1}'`
- ▶ words per line (NF: number of fields)  
`awk '{print NF}'`
- ▶ number of lines (NR: number of records)  
`awk 'END {print NR}'`
- ▶ frequency of words per line  
`awk '{print NF}' | sort -n | uniq -c`
- ▶ Find users using bash as login shell  
`awk -F: '$7 /bash/ {print $1,$7}' /etc/passwd`
- ▶ Imitate wc  
`awk '{nword=nword+NF; nchar=nchar+length($0)+1}  
END {print NR,nword,nchar}'`

### Awk using script files

- ▶ `awk -f scriptfile ...`
- ▶ `#!/usr/bin/awk -f`

### Awk using script files

- ▶ `awk -f scriptfile ...`
- ▶ `#!/usr/bin/awk -f`
- ▶ Example 1: (print a file backwards)  
`#!/usr/bin/awk -f`  
`BEGIN {while(getline) a[NR]=$0};`  
`END {while(NR) print a[NR--]}`

### Awk using script files

- ▶ `awk -f scriptfile ...`
- ▶ `#!/usr/bin/awk -f`
- ▶ Example 1: (print a file backwards)  
`#!/usr/bin/awk -f`  
`BEGIN {while(getline) a[NR]=$0};`  
`END {while(NR) print a[NR--]}`
- ▶ Example 2: (prints words on a line backwards)  
`#!/usr/bin/awk -f`  
`{ for (k=NF; k>0; k--) printf("%s ",$k);`  
`printf("\n") }`