

Introduction

I know the tools and the syntax. But I don't use it every day and when I have to use it, I'm starting nearly at the beginning. I hope the following (partial or merged) helps me to get an idea and start faster to find a solution. Maybe you too?!

Simple try it and look what you get :)

Made with current versions in 2025 (Ubuntu LTS <https://ubuntu.com>; GNU utilities for Windows <https://unxutils.sourceforge.net>).

AWK (octal, hex)

From: yin & yang
To: yin and yang

Lnx: echo -e yin \0046 yang | awk '{gsub(/\x26/, "and"); print \$0}'

Win: echo.exe yin \046 yang | gawk.exe "{gsub(/\x26/, \"and\"); print \$0}"

SED (octal, hex)

From: yin & yang
To: yin and yang

Lnx: echo -e yin \0046 yang | sed 's/\x26/and/'

Win: echo.exe yin \046 yang | sed.exe "s/\x26/and/"

AWK (Conditions)

From: 1
2
3
4
5
To: Line 2: 2
Line 3: 3
Line 5: 5

Lnx: echo -e '1\n2\n3\n4\n5' | awk '{if(\$0 > 1 && \$0 <= 5 && NR != 4) print "Line " NR ": " \$0}'

Win: echo.exe 1\n2\n3\n4\n5 | gawk.exe "{if(\$0 > 1 && \$0 <= 5 && NR != 4) print \"Line \" NR \": \" \$0}"

AWK (Substring)

From: www.google.de
To: www.google.com
google
go-away

Lnx: echo www.google.com | awk 'BEGIN{FS="."}{print \$0; print \$2; print substr(\$2, 1, 2)"-away"}'

Win: echo.exe www.google.com | gawk.exe "BEGIN{FS=\".\"}{print \$0; print \$2; print substr(\$2, 1, 2)}\"-away\"}"

SED (Substring)

From: www.google.de
To: go-away

Lnx: echo www.google.com | sed 's/^\.\.\ (. .)\.\.\.* / \1-a way/'

Win: echo.exe www.google.com | sed.exe "s/^\.\.\ (. .)\.\.\.* / \1-a way /"

AWK (System call)

From: To:
Time: Time: hh:mm

Ln: echo "Time: " | awk '{system("date +%R" | getline m); print \$0 m}' 2>/dev/null
Win: echo.exe Time: | gawk.exe '{system("\time /t" | getline m); print \$0 m}' 2>NUL

AWK (Substitution, mult.)

From: To:
E4-54-E8-6D-31-AA [subst cnt] e4:54:e8:6d:31:aa

Ln: echo E4-54-E8-6D-31-AA | awk '{print gsub("-", "."); print tolower(\$0)}'
Win: echo.exe E4-54-E8-6D-31-AA | gawk.exe "{print gsub(\"-\", \".\"); print tolower(\$0)}"

SED (Substitution, mult.)

From: To:
E4-54-E8-6D-31-AA e4:54:e8:6d:31:aa

Ln: echo E4-54-E8-6D-31-AA | sed 's/-/:/g; s/^(.*)\$/\L&/'
Win: echo.exe E4-54-E8-6D-31-AA | sed.exe "s/-/:/g; s/^(.*)\$/\L&/"

AWK (printf)

From: To:
dummy1-dummy2 Six \ Two = 3.00
123456789 123456 1234567

Ln: echo dummy1-dummy2 | awk 'BEGIN{FS="-"; equ="="; bsl="\\"}{V1="Six"; V2="Two"; V3=3; printf(" %+9s %s %-6s %s %7.2f\n", V1, bsl, V2, equ, V3)}END{print " 123456789 123456 1234567"}'
Win: echo.exe dummy1-dummy2 | gawk.exe "BEGIN{FS=\"-\"; equ=\"=\"; bsl=\"\\\"}{V1=\"Six\"; V2=\"Two\"; V3=3; printf(\" %+9s %s %-6s %s %7.2f\n\", V1, bsl, V2, equ, V3)}END{print \" 123456789 123456 1234567\"}"

VIM (Sort file, system call)

From: To:
bravo alpha
delta bravo
alpha bravo
bravo charlie
charlie delta
echo echo
foxtrot echo
echo echo
echo foxtrot

Ln/Win: :1,\$!sort



VIM (Remove duplicate lines from sorted file)

From:	To:
alpha	alpha
bravo	bravo
bravo	charlie
charlie	delta
delta	echo
echo	foxtrot
echo	
echo	
foxtrot	

LnxWin: `:%s/^\(.*\)\(\n\1\)+$/\1/`

VIM (Rearrangement, assign states to regions)

From:	To:
AMER <i>tab</i> Canada	AMER <i>tab</i> Canada
APAC <i>tab</i> China	APAC <i>tab</i> China <i>tab</i> Japan
APAC <i>tab</i> Japan	EMEA <i>tab</i> Lebanon <i>tab</i> Senegal <i>tab</i> Spain
EMEA <i>tab</i> Lebanon	
EMEA <i>tab</i> Senegal	
EMEA <i>tab</i> Spain	

LnxWin: `:%s/^\(.. \|/\(..\$\)\(\n\1\)+\t(.*$\)\1\t2\t4/
[Repeat until "Pattern not found"]`

Some Bash script

```
#!/bin/bash
hh=`date +%H`
echo "Current hour:" $hh
read -p 'Hour to search for (x=Exit)?:' hour
# \< > to suppress hits of a single digit in a two digit hour
echo $hh | grep "\<$hour\>" >/dev/null
RetCode=$?
if [ $hour == "x" ]; then
    exit
elif [ $RetCode -eq 0 ]; then
    echo "$hh eq $hour Hit"
elif [ $RetCode -eq 1 ]; then
    echo "$hh neq $hour noHit"
fi
a=`expr $hour + 1`
echo -e "$hour + 1 =" $a "\n"
```



Some Batch script

```
@echo off
REM Some echo. commands would make it nicer ...
cls
set hh=%time:~-11,2%
if %hh% LSS 10 set hh=0%time:~-10,1%
echo Current hour: %hh%
set /P hour=Hour to search for (x=Exit?):
:: Brackets to identify blank input.
if [%hour%]==[x] goto :EOF
echo %hh% | find "%hour%" >NUL
if errorlevel 1 echo %hh% neq %hour% noHit
if not errorlevel 1 echo %hh% equ %hour% Hit
for /f %%d in ('date /T') do set dat=%%d
echo %dat%
if %dat% NEQ "01.01.1970" set /a sum=1+1
echo 1+1=%sum%
pause
```

Some Vim script

```
" Execute: vim -S this.script
:function! IN()
:  call inputsave()
:  let @i = input('Hour to search for?: ')
:  call inputrestore()
:endifunction
:let hh=strftime("%H")
:echo "Current hour:" hh
:call IN()
:let hour = @i
:if hh == hour
:  echo "\n" hh "equ" hour "Hit"
:elseif hh != hour
:  echo "\n" hh "neq" hour "noHit"
:endif
:let a = hour + 1
:echo hour "+ 1 =" a
```

