**Some awk commands**

**Awk is useful for field stripping, like cut in a more powerful way. Search this document for other examples. See for example**[**gnulamp.com**](http://www.gnulamp.com/awk.html)**and**[**one-liners for awk**](http://student.northpark.edu/pemente/awk/awk1line.txt)**for some nice examples.**

awk '{ print $2, $1 }' file # Print and inverse first two columns

awk '{printf("%5d : %s\n", NR,$0)}' file # Add line number left aligned

awk '{print FNR "\t" $0}' files # Add line number right aligned

awk NF test.txt # remove blank lines (same as grep '.')

awk 'length > 80' # print line longer than 80 char)

**Some sed commands**

**Here is**[**the one liner gold mine**](http://student.northpark.edu/pemente/sed/sed1line.txt)**. And a good**[**introduction and tutorial to sed**](http://www.grymoire.com/Unix/Sed.html)**.**

sed 's/string1/string2/g' # Replace string1 with string2

sed -i 's/wroong/wrong/g' \*.txt # Replace a recurring word with g

sed 's/\(.\*\)1/\12/g' # Modify anystring1 to anystring2

sed '/<p>/,/<\/p>/d' t.xhtml # Delete lines that start with <p>

# and end with </p>

sed '/ \*#/d; /^ \*$/d' # Remove comments and blank lines

sed 's/[ \t]\*$//' # Remove trailing spaces (use tab as \t)

sed 's/^[ \t]\*//;s/[ \t]\*$//' # Remove leading and trailing spaces

sed 's/[^\*]/[&]/' # Enclose first char with [] top->[t]op

sed = file | sed 'N;s/\n/\t/' > file.num # Number lines on a file

**Regular Expressions**

**Some basic regular expression useful for sed too. See**[**Basic Regex Syntax**](http://www.regular-expressions.info/reference.html)**for a good primer.**

[\^$.|?\*+() # special characters any other will match themselves

\ # escapes special characters and treat as literal

\* # repeat the previous item zero or more times

. # single character except line break characters

.\* # match zero or more characters

^ # match at the start of a line/string

$ # match at the end of a line/string

.$ # match a single character at the end of line/string

^ $ # match line with a single space

^[A-Z] # match any line beginning with any char from A to Z

**Some useful commands**

**The following commands are useful to include in a script or as one liners.**

sort -t. -k1,1n -k2,2n -k3,3n -k4,4n # Sort IPv4 ip addresses

echo 'Test' | tr '[:lower:]' '[:upper:]' # Case conversion

echo foo.bar | cut -d . -f 1 # Returns foo

PID=$(ps | grep script.sh | grep bin | awk '{print $1}') # PID of a running script

PID=$(ps axww | grep [p]ing | awk '{print $1}') # PID of ping (w/o grep pid)

IP=$(ifconfig $INTERFACE | sed '/.\*inet addr:/!d;s///;s/ .\*//') # Linux

IP=$(ifconfig $INTERFACE | sed '/.\*inet /!d;s///;s/ .\*//') # FreeBSD

if [ `diff file1 file2 | wc -l` != 0 ]; then [...] fi # File changed?

cat /etc/master.passwd | grep -v root | grep -v \\*: | awk -F":" \ # Create http passwd

'{ printf("%s:%s\n", $1, $2) }' > /usr/local/etc/apache2/passwd

testuser=$(cat /usr/local/etc/apache2/passwd | grep -v \ # Check user in passwd

root | grep -v \\*: | awk -F":" '{ printf("%s\n", $1) }' | grep ^user$)

:(){ :|:& };: # bash fork bomb. Will kill your machine

tail +2 file > file2 # remove the first line from file

**I use this little trick to change the file extension for many files at once. For example from .cxx to .cpp. Test it first without the | sh at the end. You can also do this with the command rename if installed. Or with bash builtins.**

# ls \*.cxx | awk -F. '{print "mv "$0" "$1".cpp"}' | sh

# ls \*.c | sed "s/.\*/cp & &.$(date "+%Y%m%d")/" | sh # e.g. copy \*.c to \*.c.20080401

# rename .cxx .cpp \*.cxx # Rename all .cxx to cpp

# for i in \*.cxx; do mv $i ${i%%.cxx}.cpp; done # with bash builtins