Awk Cheat Sheet

www.thegeekstuff.com

Awk		
Awk syntax: awk *[options] '[/pattern/] {action}' [input-file]*		
-F	Specify field separator	*awk -F: '{print $1}' /etc/passwd*
/pattern/	Execute actions only for matched pattern	*awk -F: '/root/ {print $1}' /etc/passwd*
BEGIN	BEGIN block is executed once at the start	*awk -F: 'BEGIN { print "uid"} { print $1 }' /etc/passwd*
END	END block is executed once at the end	*awk -F: '{print $1} END { print "-done-"}' /etc/passwd*

Awk built-in variables	
FS	Input field separator. Specify the input-file field delimiter. Default is space.
OFS	Output field separator. Fields printed out are separated using this value. Default is space.
RS	Record separator. Specify the input file record delimiter. Default is new line.
ORS	Output record separator. Records printed out are separated using this value. Default is new line.
NF	Number of fields in the current record.
NR	Number of the record. Indicates the total number of records processed at that point.
FILENAME	Contains current input-file that is getting processed. Set to “-” while processing from stdin/pipe.
FNR [n g]	File “Number of the record”. For multiple files, FNR is reset for every file.
SUBSEP [n g]	Subscript separator for array indexes. Default is "\034"
RSTART [n g]	Match function sets RSTART with the starting location of str1 in str2.
RLENGTH [n g]	Match function sets RLENGTH with length of the str1.
ARGC [n g]	Total number of arguments passed to awk script.
ARGV [n g]	Array containing all awk script arguments. Indexed from 0 through ARGC
ARGIND [g]	Index to ARGV to retrieve the current file name
OFMT [n g]	Awk uses this to decide how to print values. Default is "%.6g"
ENVIRON [g]	Array containing all environment variables and values. e.g. ENVIRON["PATH"]
IGNORECASE [g]	Default is 0. When set to 1, it is case insensitive for string and reg-ex comparisons.
ERRNO [g]	Contains error message of an I/O operation. e.g. while using getline function.

Awk operators	
String	Use an empty space for string concatenation. e.g. *str3 = str1 str2*

Awk Cheat Sheet

www.thegeekstuff.com

Assignment	x = x + 2 is same as x += 2. Similarly use -=, *=, /=, and %=
Conditional	>, >=, <, <=, ==, !=, && (for and), || (for or)
Reg-ex	~ for match operator. ~= for "no match operator"

Awk conditional statements and loops

if	if (conditional-expression) action
if else	if (conditional-expression) action1 else action2
while	while (condition) action
do while	do action while (condition)
for	for (initialization; condition; increment/decrement) action
break	Jump out of the innermost looping that encloses it.
continue	Skip over the rest of the loop body and start the cycle again.
exit	Stop executing the script and exit

Awk associative arrays

Basics	Assign element : arrayname[string]=value	e.g. *item[105]="Laser Printer"*
	Refer element : variable=arrayname[string]	e.g. *myvar=item[105]*
	Delete element : delete arrayname[index]	e.g. *delete item[105]*
	Browse elements : for (var in arrayname) actions	e.g. *for (x in item) print item[x]*
asort	asort function sorts the array values and stores them in indexes from 1 through n. Where n is the total number of elements in the array.	
asorti	Just like sorting array values, you can take all the array indexes, sort them, and store it in a new array using asorti function.	

Awk printf command

Format specifier	s = String	e = Exponential floating point	o = Octal
	c = Single character	f = Fixed floating point	x = Hexadecimal
	d = Decimal	g = Use either e or f	% = Percentage symbol
Special character	\n = Newline	\t = Tab	\v = Vertical tab
	\b = Backspace	\r = Carriage return	\f = Form feed
Number	Fixed Width. Number immediately after %. e.g. "%5s". By default it is right justified		
-	Fixed Width (left justified). Minus symbol immediately after % and before number. e.g. "%-5s"		

0	Leading zeros. Add a 0 (zero) before the number. e.g. "%05s"

Awk Functions	
index(str1,str2)	Give location of str2 in str1
length(str1)	Length of the string
split(str,arr,sep)	Splits the string (str) using separator (sep) and stores output in the array (arr)
substr(str,loc,len)	Extracts number of characters (len) from string (str), starting at location (loc)
sub(str1,str2,var) [n g]	In the input string (var), str1 is replaced with str2, and output is stored back in var
gsub(str1,str2,var) [n g]	Same as sub, but global. It does multiple substitutions on the same input string (var).
match(str1,str2) [n g]	Returns positive number when str1 is present in str2.
toupper(str) [g]	Converts str to upper-case
tolower(str) [g]	Converts str to lower-case
Numeric	init(n), log(n), sqrt(n), sin(n), cos(n), atan2(m,n)
Random Number	rand() - random number between 0 and 1 srand(n) – generate random number from n

Additional Awk Commands	
print	Print a record, or field, or value
|& [g]	Two way communication between awk command and external process
system(cmd)	Execute the Unix OS command (cmd) from awk
systime() [g]	Current time in epoch time. Combine with strftime [g]. e.g. print strftime("%c",systime())
getline	Read the next line from the input-file. Sets $0, NF, NR, FNR
getline var	Read next line from the input-file and store it in variable (var). Sets NR, FNR
getline < file [n g]	Read next line from another input-file. Sets $0, NF
getline var < file [n g]	Read next line from another input-file and store it in variable (var).
cmd | getline	Execute Unix OS command (cmd) and store output in $0

[n g] is NAWK and GAWK Only. [g] is GAWK Only.