

Abbreviations

d: directory
f: filename
ptn: pattern

UNIX

Environment Control

cd d Change to directory d
cd / Change to directory root
cd .. Change to directory parent
mkdir d Create new directory d
rmdir d Remove EMPTY directory d
rm -R d Remove directory d
mv d1 d2 Rename directory d1 as d2
alias name="cmd" Create command alias
unalias name Remove command alias name

Input, Output, Communication, & Help

read var Read line from standard input
to variable var
echo string Display the string
man name UNIX manual entry for name

Process Control

Ctrl/c Interrupt processes
Ctrl/z Suspend current process
ps Print process status stats
jobs Print list of jobs
cmd & Run command cmd in background
fg %n Resume foreground job n
bg %n Resume background job n
stop %n Suspend background job n
kill n Kill process n
kill -9 n Remove process n
sleep n Sleep for n seconds
exit Exit from shell

Environment Status

uname Print Unix name
whoami Display current user
id Display current user
groups Display current groups
pwd Print working directory
ls [f...] List files in directory
ls -l [f...] List files in detail
ls -a [f...] List hidden files too
history Display recent commands
! n Submit recent command n
date Print date & time
date +%y.%m.%d Print year.month.day

Expressions:

expr n1 + n2
expr n1 * n2

File Manipulation

cat f1 f2 List contents of files
cp f1 f2 Copy file f1 into f2
mv f1 f2 Rename file f1 as f2
mv f1 [f2...] d Move files f1 [f2...] to directory d
rm f Delete (remove) file f
rm -i f Ask whether to delete (remove) file f
ln f1 f2 Make a hard link f2 to file f1
ln -s f1 f2 Make a soft link f2 to file f1
sort (-n) f Alphabetically (Numerically) sort file f
head -nb f Output the first b line(s) of f
tail -nb f Output the last b line(s) of f
cut -c n1-n2 Cut from char. n1 to char. n2 from the line
tr 's1' 's2' Change string s1 for string s2
grep -e 'ptn' f Output lines that match ptn
wc -l -w -c f Count line,- word,- char of f
more f List file f contents by screen
chown owner(:group) f Change user(:group) name in f
chgrp name f Change group name in file f
chmod [u g o a] [+ -][rwx] f Change permission of f
chmod nnn f Use numeric permission
cmp f1 f2 Compare two files
diff f1 f2 List file differences
find d -name f -print Find files f from directory d
find d -name f -exec cmd {} ';' ... and execute cmd

Unix scripts

Parameters:
\$1-9: positional parameters.
\$\$: actual PID.
\$# : number of parameters.
\$@ : all of the parameters.
\$* : all of the parameters.
\$? : Return value of the last command.
\$! : PID of the last background process.
shift shift positional parameters to the left

Functions:

fcn_name() {
...
}
case structure:
case word in
pattern 1) commands 1;;
pattern 2) commands 2;;
...
pattern n) commands n;;
esac

while cycle:

while command_1
do command_2
done

if-then-else:

if command1
then command_list1
elif command2
then command_list2
else command3
fi

for cycle:

for name in set
do
command list
done

test ([]) expressions:

Numerical expressions:
[\$a * \$b]
*: -lt (<) -gt (>) -le (<=)
-ge (>=) -eq (=) -ne (!=)

String expressions:

[\$a = \$b] [\$a != \$b]
[-n \$b] not null string [-z \$b] null string
File expressions:
[-f FILE] the regular file exists
[-d DIRECTORY] the directory exists
[-L LINK] the link exists

Other commands

>, >> Redirect standard output
< Redirect standard input
<< here document
| pipe
tee replicate standard output
`` substitute standard output

Perl scripts

blocks: if-else structure:
{ if (term)
command_1; block_1
... else
command_n; block_2
}

for cycle: while cycle:
for (init; test; incr) while (expression)
block block

Logical operators:

! not
&& and
|| or

Logical expressions

Numerical String
== eq (equal to)
> gt (greater than)
< lt (less than)
>= ge (greater than or equal to)
<= le (less than or equal to)
!= ne (not equal to)

File test expressions

-f the regular file exists
-d the directory exists
-z the file is empty
-s size of the file
-r file is readable
-w file is writeable

Variables

\$scalar @list

Input/output

open(HANDLE, "<f") file open/read
open(HANDLE, ">f") file open/overwrite
open(HANDLE, ">>f") file open/append
close(HANDLE) file close
print HANDLE x print x to HANDLE
<HANDLE> read from HANDLE

Predefined I/O handlers

STDIN, STDOUT, STDERR

Other commands

chomp remove end char (newline)
split split scalar to list
join join list into scalar
sort sort list
reverse reverse list (or scalar)
push add elements to end of list
pop take elements from end of list

Regular expressions

Metacharacters:

. Match any character (except newline)
^ Match the beginning of the line
\$ Match the end of the line
| Alternation
() Grouping
[] Character class
\ Quote the next metacharacter
? Match 0 or 1 times
* Match 0 or more times
+ Match 1 or more times
{n,m} Match x times, where n ≤ x ≤ m

Other special characters:

\n new line character
\r carriage return char
\t tab character
\f form feed character

Predefined classes:

\w word char
\W not word char
\d digit char
\D not digit char
\s white space
\S not white space

Match operator

m/pattern/modifiers

Find/replacement operator

s/pattern/replacement/modifiers

Modifiers (optional):

i case insensitive match
g global (iterative) match