


UNIX UTILITIES !!!

SPA-02-BASIC UNIX UTILITIES-I 16-11-1999




Some UNIX Utilities

- cat
- cp
- date
- diff
- echo
- file
- finger
- grep
- head
- lp
- ls
- cc

- mesg
- man
- mv
- passwd
- more
- rm
- sed
- sort
- tail
- talk
- w
- perl

- vi
- who
- write
- > < >> | &
- find
- awk
- ps
- cpio
- tar
- dd
- wc


SPA-02-BASIC UNIX UTILITIES-I 16-11-1999



Some UNIX Utilitites grep

```
$ cat grepnotes
The grep(global regular expression print)
utility searches through a file to see if
it contains a specified string of
characters and displays each line that
contains the string.
$ grep 'contains' grepnotes
it contains a specified string of
contains the string.
$
```

SPA-02-BASIC UNIX UTILITIES-I 16-11-1999




Some UNIX Utilitites grep

```
$ who
rik          ttyp2          Apr 1410:59
bf980110     ttyp3          Apr 14 10:33
cf971002     ttyp4          Apr 14 10:41
bf980121     ttyp5          Apr 14 11:00

$ who|grep 'rik'
rik          ttyp2          Apr 14 10:59
$ who|grep 'cf'
cf971002     ttyp3          Apr 14 11:12
$
```

SPA-02-BASIC UNIX UTILITIES-I 16-11-1999



Some UNIX Utilitites

head

\$ cat headpages

The head utility displays the first ten lines of a file. By attaching a number as an option it can display as many lines.

\$ head -1 headpages

The head utility displays the first ten


\$ head -2 headpages

The head utility displays the first ten lines of a file. By attaching a

\$

SPA-02-BASIC UNIX UTILITIES-I

16-11-1999



Some UNIX Utilitites

sed

\$ cat sedpages

The sed utility looks at top of a file. By attaching a number as an option it can display as many lines.

\$ sed 1q sedpages


The sed utility looks at top of a file. By

\$ sed 2q sedpages

The sed utility looks at top of a file. By attaching a number as an option it can

SPA-02-BASIC UNIX UTILITIES-I

16-11-1999



Some UNIX Utilitites

tail

\$ cat tailpages

The tail utility displays the last ten lines of a file. By attaching a number as an option it can display as few lines.

\$ tail -1 tailpages


an option it can display as few lines.

\$ tail -2 tailpages

lines of a file. By attaching a number as an option it can display as few lines.

SPA-02-BASIC UNIX UTILITIES-I

16-11-1999



Some UNIX Utilitites


sort

\$ who

df980213	tty0p1	Apr 14 11:30
df980256	tty0p3	Apr 14 11:38
df980223	tty1p0	Apr 14 11:28
df980231	tty1p7	Apr 14 11:30
df980248	tty1p10	Apr 14 11:30
df980252	tty1p11	Apr 14 11:38
df980245	tty0p15	Apr 14 11:24
rik	ttyp2	Apr 14 10:59
cf971011	ttyp4	Apr 14 11:21

SPA-02-BASIC UNIX UTILITIES-I

16-11-1999



Some UNIX Utilitites


Pipe

sort

```

$ who|sort
cf971011    ttyt4      Apr 14 11:21
df980213    tty0p1     Apr 14 11:30
df980223    tty1p0     Apr 14 11:28
df980231    tty1p7     Apr 14 11:30
df980245    tty0p15    Apr 14 11:24
df980248    tty1p10    Apr 14 11:30
df980252    tty1p11    Apr 14 11:38
df980256    tty0p3     Apr 14 11:38
rik         ttyt2      Apr 14 10:59
  
```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Some UNIX Utilitites

Filter


sort

```

$ who|sort|tail -2
df980256    tty0p3     Apr 14 11:38
rik         ttyt2      Apr 14 10:59

$ who|sort|sed 2q
cf971011    ttyt4      Apr 14 11:21
df980213    tty0p1     Apr 14 11:30
$
  
```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Permissions

Long Listing of a File

```

$ ls -l
-rwxr-xr-x 1 rik staff 42496 Mar 26 18:40 SPA
  
```

1

2

3

4

5

6

7

permissions link owner group size date & time name

permissions

a

r

w

x

r

w

x


r

w

x

type owner group others

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Permissions

type

for owner or group or others

character	meaning
-	ordinary file
b	block file
c	character file
d	directory
l	symbolic link


r read (1) or - (0)

w write (1) or - (0)

x execute(1) or - (0)

- Represent as Octal numbers
- Ex: 666 indicates (110 110 110) -> read, write perm. for all
- Usage of chmod : \$ chmod 666 file

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Relative Permissions

State the following with Relative Permissions:


- who** you are giving permissions to
- what **operations** to be done (add,subtract or set)
- what the **permissions** are

who	description
a	all users
g	owner's group
o	all others
u	just the user

operator	description
+	adds the mode
-	removes mode
=	sets mode

Permission	
x	sets execute
r	sets read
w	sets write

1-1999




Example of Permissions

```

$ cat whoson
Date
echo Users Currently Logged In
who
$
$ ls -l whoson
-rw-r----- 1 rik staff 40 Apr 14 12:33 whoson
$ whoson
sh: whoson: Execute permission denied.
$ chmod u+x whoson
$ ls -l whoson
-rwxr----- 1 rik staff 40 Apr 14 12:33 whoson

```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Example of Permissions

```

$ whoson
Tue Apr 14 12:38:25 MAL 1998
Users Currently Logged In
bf980108    tty1p4      Apr 14 12:34
df980246    tty1p5      Apr 14 12:07
bf980107    tty1p6      Apr 14 12:34
df971024    tty1p11     Apr 14 12:14
df971110    tty0p14     Apr 14 12:34
df980267    tty2        Apr 14 12:34
rik         tty4        Apr 14 12:26
df980216    tty5        Apr 14 12:35
cf971005    tty7        Apr 14 12:30

```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Example of Permissions

```

$ pwd
/users/staff/rik/unix
$ ls -l
total 10
-rw-r----- 1 rik staff 22 Apr 11 16:45 colors.1
-rw-r----- 1 rik staff 16 Apr 11 16:45 colors.2
-rwxr----- 1 rik staff 123 Apr 11 16:37 memo
-rw-r----- 1 rik staff 49 Apr 11 16:39 months
-rwxrwxrwx 1 rik staff 41 Apr 11 17:11 whoson

```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Example of Permissions

```

$ chmod a+rw colors.1

$ ls -l colors.1
-rw-rw-rw- 1 rik staff 22 Apr 11 16:45 colors.1
$ chmod o-rw whoson
$ ls -l whoson
-rwxrwx--x 1 rik staff 41 Apr 11 17:11 whoson
  
```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Directory access permissions

```

$ pwd
/users/staff/rik
$ chmod a+rwx /users/staff/rik/unix
$ cd unix
$ ls -l
total 10
-rw-rw-rw- 1 rik staff 22 Apr 11 16:45 colors.1
-rw-r----- 1 rik staff 16 Apr 11 16:45 colors.2
-rwxr----- 1 rik staff 123 Apr 11 16:37 memo
-rw-r----- 1 rik staff 49 Apr 11 16:39 months
-rwxrwx--x 1 rik staff 41 Apr 11 17:11 whoson
$ ls -ld /users/staff/rik/unix
drwxrwxrwx 2 rik staff 1024 Apr 11 17:11 /users/staff/rik/unix
  
```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Links


There are two types of links.
 A **hard link** is a pointer to a file.
 A **symbolic link** is an indirect pointer to a file.
 It is a directory entry that contains the path name of the pointed-to file.

Hard Links to a file

```

$ cat file_a
This is file A.
  
```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Hard Link

```

$ ln file_a file_b
$ cat file_b
This is file A.
$ vi file_b
:
$ cat file_b
This is file B after the change.
$ cat file_a
This is file B after the change.
  
```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Hard Link

```

$ cat file_c
This is file C.
$ cp file_c file_d
$ cat file_d
$ This is file C.
$ vi file_d
:
$ cat file_d
$ This is file D after the change.
$ cat file_c
This is file C.

```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Hard Link

```

$ ls -l f*


-rw-r----- 2 rik staff 34 Apr 18 11:54 file_a
-rw-r----- 2 rik staff 34 Apr 18 11:54 file_b
-rw-r----- 1 rik staff 17 Apr 18 11:56 file_c
-rw-r----- 1 rik staff 33 Apr 18 11:58 file_d

$ ls -l f*

144417 file_a 144417 file_b 144436 file_c 144479 file_d

```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Symbolic Link

Symbolic Links

To make a symbolic link to a file or a directory, use `ln` with the `-s` option.


Let us try to create a symbolic link `/tmp/memo1` to the file `memo`.

```

$ ln -s memo /tmp/memo1
$ ls -l memo /tmp/memo1
lrwxr-x--- 1 rik staff 4 Apr 18 12:24 /tmp/memo1 -> memo
-rwxr----- 1 rik staff 123 Apr 11 16:37 memo

```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Symbolic Link

```

$ ln -s /users/staff/rik/unix /tmp/new1
$ cd /tmp/new1
$ pwd
/tmp/new1
$ ls -l memo
-rwxr----- 1 rik staff 123 Apr 11 16:37 memo
$ cp memo memo2
$ rm memo
$ cat memo1
cat: Cannot open memo1: No such file or directory
$ ls -l memo1
lrwxr-x--- 1 rik staff 4 Apr 18 13:25 memo1 -> memo
$ rm memo1

```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Regular Expression (RE)

- A Regular Expression (RE) is a pattern used in the search mechanism.
- It consists of:
 - (a) A string of **normal alphanumeric** characters.
 - (b) A string with normal alphanumeric characters plus **Meta-characters** or special characters.
- Regular Expressions are used in **grep, egrep, sed, awk, find, perl** and numerous other utilities.

SPA-02-BASIC UNIX UTILITIES-I

16-11-1999



Metacharacters

- ⌘ **.** matches any single character
- ⌘ **^** matches the beginning of a line
- ⌘ **\$** matches the end of a line
- ⌘ A **regular expression** followed by an asterisk (*) matches zero or more occurrences of the preceding regular expression

SPA-02-BASIC UNIX UTILITIES-I

16-11-1999



Metacharacters (cont..)

- ⌘ **[]** defines a set of characters
 - ☑ It matches any single character in the set.
 - ☑ Examples: **[A-Z]**, **[aeiou]**
 - ☑ Note: Characters *****, **^**, **\$**, and **** lose their special meaning inside the square brackets.
 - ☑ If the first character in the bracket is **^**, then it matches any single character *not* in the set.

SPA-02-BASIC UNIX UTILITIES-I

16-11-1999




Metacharacters for Extended REs

- ⌘ Available for **egrep**
- ⌘ A regular expression (RE) followed by a **+** matches one or more matches of the regular expression.
- ⌘ An RE followed by a **?** matches zero or one match of the regular expression.
- ⌘ **r1|r2** will match if there is a match for r1 or for r2

SPA-02-BASIC UNIX UTILITIES-I


16-11-1999



Rules of Matching

⌘ Quoted parentheses `()` and `()` can be used for grouping RE's.


SPA-02-BASIC UNIX UTILITIES-I 16-11-1999



Examples: RE matches

<code>thing</code>	thing anywhere in the line
<code>^thing</code>	thing at the beginning of the line
<code>thing\$</code>	thing at the end of the line
<code>^thing\$</code>	Line that contains only thing
<code>[tT]hing</code>	thing or Thing anywhere in the line
<code>thing[0-9]</code>	thing followed by a digit anywhere
<code>thing[^0-9]</code>	thing followed by any character other than a digit
<code>thing.*thing</code>	thing followed by any number of characters followed by thing

SPA-02-BASIC UNIX UTILITIES-I 16-11-1999




Data File for grep operations

Create the following file (Using vi editor)

thing oh thing of all things!
please do remember one Thing ...
though there are many facets in a thing
there's but thing1, 2 or 3 ...
in a true facet of life!
so make haste to give that one ...
thing

thing


SPA-02-BASIC UNIX UTILITIES-I 16-11-1999



grep operations on thing

```
$ grep 'thing' thing
$ grep '^thing' thing
$ grep 'thing$' thing
$ grep '^thing$' thing
$ grep '[tT]hing' thing
$ grep 'thing[0-9]' thing
$ grep 'thing[^0-9]' thing
$ grep 'thing.*thing' thing
```


SPA-02-BASIC UNIX UTILITIES-I 16-11-1999



Full Regular Expressions - egrep

RE	Matches	Examples
ab+c	a followed by one or more b 's followed by a c	yabcw,abbc57
ab?c	a followed by zero or one b followed by a c	back, abcdef
ab ac	either ab or ac	ab, ac, abac

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Other usages of grep

```

% List the names of all subdirectories in the current directory
ls -l | grep '^d'

% List the files others can read
ls -l | grep '^.....r..'
  
```


SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Major options for grep

- % **-c** displays only a count of matching lines
- % **-i** ignores the upper and lower case distinctions in pattern matching
- % **-l** lists only the names of files containing matching lines
- % **-n** precedes each matching line with its line number in the file
- % **-v** displays all lines that do not match

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Examples related to System Administration

```

% Check whether vijay is logged on
% who | grep vijay

% List all filenames that does not end in h
% ls | grep -v '.*h'
  
```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Examples on grep usage


File testa	File testb	File testc
aaabb	aaaaa	AAAAA
bbbcc	bbbbb	BBBBB
ff-ff	cccc	CCCC
ccdd	dddd	DDDD
dddaa		

```

$ grep bb testa
$ grep -v bb testa
$ grep -n bb testa

```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Examples on grep usage (cont..)


```

$ grep bb *
$ grep -i bb *
$ grep -i BB *
$ grep -c bb *
$ grep -n '^' testa

Others...
$ grep 'st.*ing' test2
$ vi `grep -l 'Sampson' *`

```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Stream Editing sed

- sed [-n] [-e script] [-f scriptfile] {inputfile}*

% Edits an input stream according to a given set of editing commands (called sed scripts).

% The editing command may be given on the

- command line (default)
 - or in a scriptfile (using -f option)


% For multiple editing commands on the command line

- precede each editing command with -e flag

% -n option is to output only edited lines

- the default is to output every line


SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



sed commands

address a text	Append text after the line specified by the address
addressRange c/ text	Replace the text specified by addressRange with text
addressRange d	Delete the text specified by addressRange
address i/ text	Insert text after the line specified by address
address r fileName	Append the contents of fileName after the line specified by address
addressRange s/expr/ str/	Substitute the first occurrence of RE expr by the string str
addressRange s/expr/ str/g	Substitute every occurrence of the RE expr by the string str

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



sed commands (cont..)

⌘ addressRange **p**

- ☑ Print the specified line(s), usually used with the -n option

⌘ **q**


- ☑ Quit after printing the current line

⌘ **!**

- ☑ Don't do

⌘ There is more... Refer man pages!!!

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Address Specification


⌘ An address is

- ☑ a line number, or
- ☑ a regular expression
 - ☑ enclosed within two slashes

⌘ If no address is specified

- ☑ The command is applied to all the lines in input

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Address Range


⌘ An addressRange can be a single address

- ☑ The command is applied all matching lines

⌘ or, two addresses separated by a comma

- ☑ The command is applied to the first line matching the
 - first address
- ☑ and, to all subsequent lines
 - ☑ until a line is found that matches the second address (inclusive).
- ☑ The steps are repeated until the end of input file is
 - reached.


SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Data file for sed examples

2	25	114	register	sfile
5	20	188	sphere	
12	29	176	trapeg	
1	25	110	sphere	
10	40	193	whereis	
29	114	671	total	

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Example 1

⌘ Substitution

\$ sed 's/sphere/SPHERE/' sfile

2	25	114	register
5	20	188	SPHERE
12	29	176	trapeg
1	25	110	SPHERE
10	40	193	whereis
29	114	671	total

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Example 2

⌘ Substitution within an address range

\$ sed '1,3s/sphere/SPHERE/' sfile

2	25	114	register
5	20	188	SPHERE
12	29	176	trapeg
1	25	110	sphere
10	40	193	whereis
29	114	671	total

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Example 3

⌘ Specifying address using RE

\$ sed '/sphere/s/1/ONE/' sfile

2	25	114	register
5	20	ONE88	sphere
12	29	176	trapeg
ONE	25	110	sphere
10	40	193	whereis
29	114	671	total

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Example 4

⌘ Global substitution

\$ sed '/sphere/s/1/ONE/g' sfile

2	25	114	register
5	20	ONE88	sphere
12	29	176	trapeg
ONE	25	ONEONE0	sphere
10	40	193	whereis
29	114	671	total

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Example 5

⌘ Specifying address range using RE

\$ sed '/r.*g/,/r.*g/s/1/ONE/g' sfile

2	25	ONEONE4	register
5	20	ONE88	sphere
ONE2	29	ONE76	trapeg
1	25	110	sphere
10	40	193	whereis
29	114	671	total

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Example 6

⌘ Specifying address range using RE

\$ sed '/sphere/,/trapeg/s/1/ONE/g' sfile

2	25	114	register
5	20	ONE88	sphere
ONE2	29	ONE76	trapeg
ONE	25	ONEONE0	sphere
ONE0	40	ONE93	whereis
29	ONEONE4	67ONE	total

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Example 7

⌘ Substitution using a pattern

\$ sed '/sp.*e/s//CONE/' sfile

2	25	114	register
5	20	188	CONE
12	29	176	trapeg
1	25	110	CONE
10	40	193	whereis
29	114	671	total

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



Example 8

⌘ Do default action and quit early

\$ sed 2q sfile


2	25	114	register
5	20	188	sphere

⌘ Printing only certain lines

\$ sed -n 3p sfile

2	25	114	register
5	20	188	sphere
12	29	176	trapeg

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999




Example 9

⌘ Don't print lines matching the address

```
$ sed -n '/9!/p' sfile
```

2	25	114	register
5	20	188	sphere
1	25	110	sphere

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



More examples using sed


```
Line one
The second line.  new
The third.
This is line four.
Five.
This is the sixth sentence.
This is line seven.
Eight and last.
```

```
/This/ i\
BEFORE.
```

```
2,4 c\
INSERT THESE\
LINES FOR\
SELECTED ONES.
change_demo
```

```
2 a\
AFTER.
append_demo
```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999



More examples using sed

```
$ sed '/line/ p' new
$ sed -n '/line/ p' new
$ sed -n '3,6 p' new
$ sed '5 q' new
$ sed -f append_demo new
$ sed -f insert_demo new
$ sed -f change_demo new
```

SPA-02-BASIC UNIX UTILITIES-I
16-11-1999