Trabajando con el filesystem

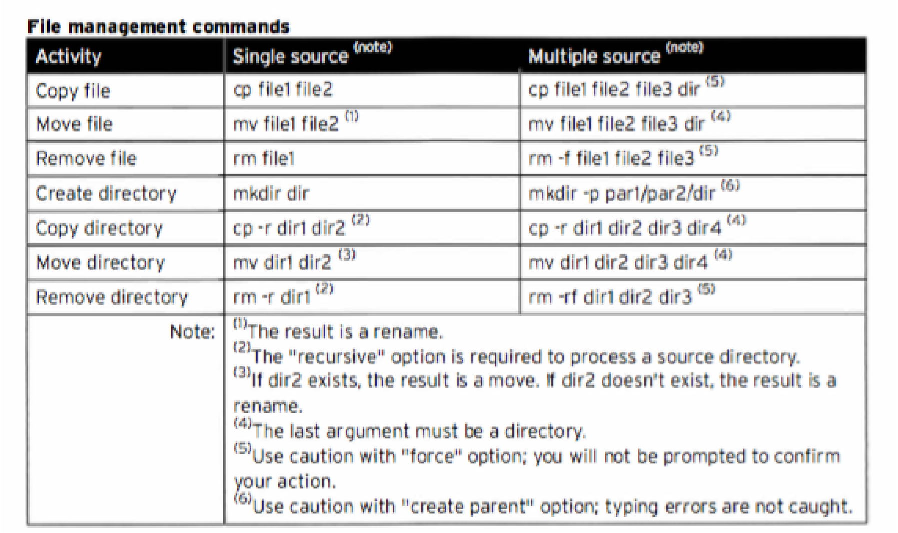
Seminario de Desarrollo de Software - Casa Central.

Maximiliano Osorio
mosorio@inf.utfsm.cl

Universidad Técnica Federico Santa María

5 de septiembre de 2017

Comandos



File management commands

Activity	Single source (note)	Multiple source (note)
Copy file	cp file1 file2	cp file1 file2 file3 dir (5)
Move file	mv file1 file2 (1)	mv file1 file2 file3 dir (4)
Remove file	rm file1	rm -f file1 file2 file3 (5)
Create directory	mkdir dir	mkdir -p par1/par2/dir (6)
Copy directory	cp -r dir1 dir2 (2)	cp -r dir1 dir2 dir3 dir4 (4)
Move directory	mv dir1 dir2 (3)	mv dir1 dir2 dir3 dir4 (4)
Remove directory	rm -r dir1 (2)	rm -rf dir1 dir2 dir3 (5)
Note:	(1)The result is a rename. (2)The "recursive" option is required to process a source directory. (3)If dir2 exists, the result is a move. If dir2 doesn't exist, the result is a rename. (4)The last argument must be a directory. (5)Use caution with "force" option; you will not be prompted to confirm your action. (6)Use caution with "create parent" option; typing errors are not caught.	

mkdir

mkdir crea un directorio o subdirectorios.

```
[mosorio@ssh ~]$ mkdir glob
```

Si ruta completa no existe

```
[mosorio@ssh ~]$ mkdir super/tuper
mkdir: cannot create directory ‘super/tuper’:
   No such file or directory
```

Solución:

```
[mosorio@ssh ~]$ mkdir -p super/tuper
```

Ejemplo

```
[mosorio@ssh ~]$ mkdir -p Videos/Watched
[mosorio@ssh ~]$ mkdir Documents
[mosorio@ssh ~]$ cd Documents
[mosorio@ssh Documents]$ mkdir ProyectX
   ProyectY
[mosorio@ssh Documents]$ mkdir -p Thesis/
   Chapter1 Thesis/Chapter2 Thesis/Chapter3
[mosorio@ssh Documents]$ cd
[mosorio@ssh ~]$ ls -R Videos/ Documents
...
```

Ejemplo

```
[mosorio@ssh ~]$ cd Videos/
[mosorio@ssh Videos]$ touch v1.ogg
[mosorio@ssh Videos]$ cp v1.ogg v3.ogg
[mosorio@ssh Videos]$ ls
v1.ogg  v3.ogg  Watched
```

Ejemplo

```
[mosorio@ssh Documents]$ touch thesis_c1.pdf
   thesis_c2.pdf
[mosorio@ssh Documents]$ cp thesis_c1.pdf
   thesis_c2.pdf Thesis/ ProyectX/
cp: omitting directory ‘Thesis/’
[mosorio@ssh Documents]$ ls -R
...
[mosorio@ssh Documents]$ cp -r Thesis/ ProyectX
```

Ejemplo

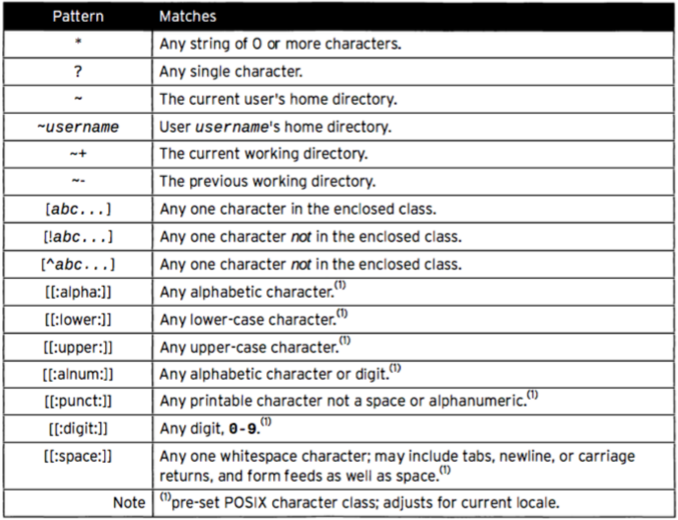
```
[mosorio@ssh Documents]$ mv ProyectX ProyectZ
[mosorio@ssh Documents]$ rm thesis_c2.pdf
[mosorio@ssh Documents]$ rm Thesis/
rm: cannot remove ‘Thesis/’: Is a directory
[mosorio@ssh Documents]$ rm -r Thesis/
[mosorio@ssh Documents]$ rm -ri Proyect
ProyectY/ ProyectZ/
[mosorio@ssh Documents]$ rm -ri ProyectY/
rm: remove directory ‘ProyectY’? Y
```

Brace expresion

```
bash$ echo a{d,c,b}e
ade ace abe
```

```
mkdir /usr/local/src/bash/{old,new,dist,bugs}
```

Expresiones regulares



Pattern	Matches
*	Any string of 0 or more characters.
?	Any single character.
~	The current user's home directory.
~*username*	User *username*'s home directory.
~+	The current working directory.
~-	The previous working directory.
[abc...]	Any one character in the enclosed class.
[!abc...]	Any one character *not* in the enclosed class.
[^abc...]	Any one character *not* in the enclosed class.
[[:alpha:]]	Any alphabetic character.[1]
[[:lower:]]	Any lower-case character.[1]
[[:upper:]]	Any upper-case character.[1]
[[:alnum:]]	Any alphabetic character or digit.[1]
[[:punct:]]	Any printable character not a space or alphanumeric.[1]
[[:digit:]]	Any digit, **0-9**.[1]
[[:space:]]	Any one whitespace character; may include tabs, newline, or carriage returns, and form feeds as well as space.[1]
Note	[1]pre-set POSIX character class; adjusts for current locale.

Preparando...

```
[mosorio@ssh ~]$ mkdir glob
[mosorio@ssh ~]$ cd glob/
[mosorio@ssh glob]$ ls
[mosorio@ssh glob]$ touch alfa bravo charlie
   delta echo able baker cast dog easy
[mosorio@ssh glob]$ ls
able   alfa   baker   bravo   cast   charlie   delta
    dog   easy   echo
```

Match file name using path name expansion

```
[mosorio@ssh glob]$ ls *a*
able  alfa  baker  bravo  cast  charlie  delta
    easy
[mosorio@ssh glob]$ ls [ac]*
able  alfa  cast  charlie
[mosorio@ssh glob]$ ls ????
able  alfa  cast  easy  echo
[mosorio@ssh glob]$ ls ?????
baker  bravo  delta
```

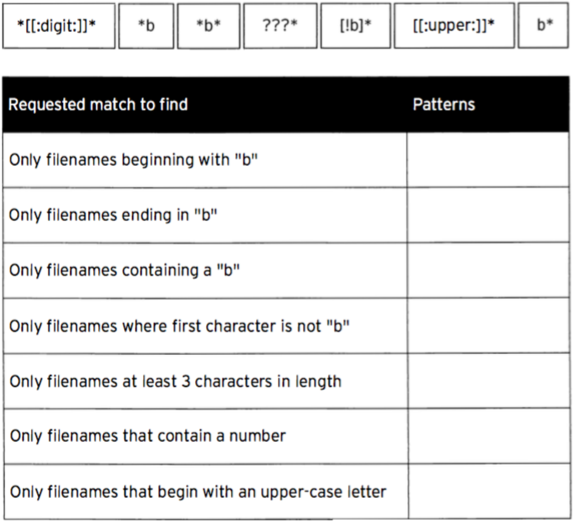
Match file name using path name expansion

```
[mosorio@ssh glob]$ ls ~/glob/
able  alfa  baker  bravo  cast  charlie  delta
    dog  easy  echo
[mosorio@ssh glob]$ echo ~/glob/
/home/mosorio/glob/
```

Match file name using path name expansion

```
[mosorio@ssh glob]$ echo {Sunday,Monday,Tuesday
   ,Wednesday}.log
Sunday.log Monday.log Tuesday.log Wednesday.log
[mosorio@ssh glob]$ echo file{1..3}.log
file1.log file2.log file3.log
[mosorio@ssh glob]$ echo file{a..c}.log
filea.log fileb.log filec.log
[mosorio@ssh glob]$ echo file{a,b}{1,2}.log
filea1.log filea2.log fileb1.log fileb2.log
[mosorio@ssh glob]$ echo file{a{1,2},b,c}.log
filea1.log filea2.log fileb.log filec.log
```

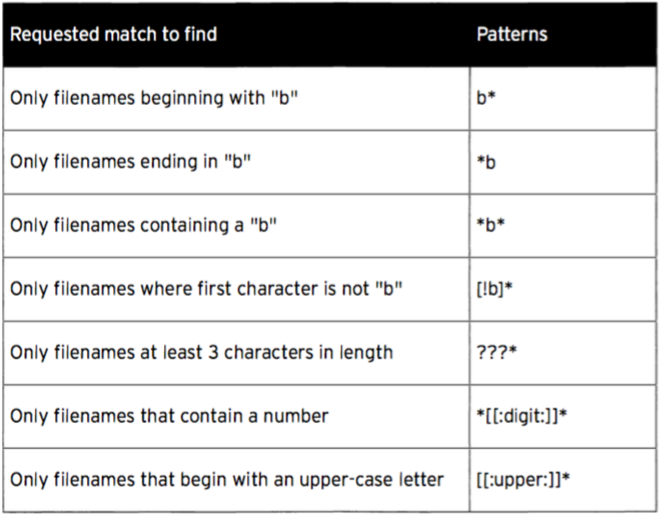
Preguntas



[[:digit:]]	*b	*b*	???*	[!b]*	[[:upper:]]*	b*

Requested match to find	Patterns
Only filenames beginning with "b"	
Only filenames ending in "b"	
Only filenames containing a "b"	
Only filenames where first character is not "b"	
Only filenames at least 3 characters in length	
Only filenames that contain a number	
Only filenames that begin with an upper-case letter	

Respuestas



Requested match to find	Patterns
Only filenames beginning with "b"	b*
Only filenames ending in "b"	*b
Only filenames containing a "b"	*b*
Only filenames where first character is not "b"	[!b]*
Only filenames at least 3 characters in length	???*
Only filenames that contain a number	*[[:digit:]]*
Only filenames that begin with an upper-case letter	[[:upper:]]*

Subtitución de comandos

```
‘command‘
$(command)
[mosorio@ssh glob]$ echo $(hostname)
ssh.inf.utfsm.cl
[mosorio@ssh glob]$ echo ‘hostname‘
ssh.inf.utfsm.cl
```

Consejo

Utilizar backticks no es recomendado dado que se puede confudir con las comillas simples y no puede colocar backticks dentro de backticks.

Subtitución de comandos

- En bash shell hay caracteres reservados.
- Para escapar estos comandos se utiliza el backslash

```
[mosorio@ssh glob]$ host=$(hostname)
[mosorio@ssh glob]$ echo $host
ssh.inf.utfsm.cl
[mosorio@ssh glob]$ echo "$host"
ssh.inf.utfsm.cl
[mosorio@ssh glob]$ echo "\$host"
$host
```

Subtitución de comandos

```
[mosorio@ssh glob]$ echo "La variable sera
   evaluada $(hostname)?"
La variable sera evaluada ssh.inf.utfsm.cl?
[mosorio@ssh glob]$ echo 'La variable sera
   evaluada $(hostname)?'
La variable sera evaluada $(hostname)?
```