A\* capital A followed by 0 or more char (note: need $ls –d flag or do $echo)

? single char

[] single character that is one of the characters in the brackets.

[a-z][A-Z][0-9]

[[:alpha:]] upper, lower, digit, alnm, blank, space(whitespace), punct

[!a-z] a single character whose value is not greater than or equal to a and less than or equal to z

? and \* will not match a leading . in a name (indicating a hidden file)

NOTE: you can’t copy a file on top of a dir or or a dir on top of a file

cp file1 file2 🡪 copy file1 to file2. File2 is overwritten if it exists.

cp file1 file2 file3 fileN dir 🡪 copy one of more files into an existing dir

cp –r dir1 dir2 🡪 copy dir1 and all of its contents to dir2. If dir2 does not exit, the new copy of dir1 is named dir2. If dir2 exists, the copy of dir1 is placed in dir2.

cp –r dir1….dirN dir2 🡪 copy one or more directories and their contents into the existing dir2

mv file1 file2 🡪 file2 is deleted if it exists. Then file1 is renamed file2

mv file or dir1….fileordirN dir 🡪 move one or more existing files or dir into an existing dir. If a dir is moved, everything is moved, everything beneath it goes with it.

mv dir1 dir2 🡪 if dir2 does not exist, rename dir1 to dir2. If dir2 exists, dir1 is moved into dir2

Owners – each piece of data on unix has one owner (whomever created it)

Groups – users on the system are organized into groups

* Most users in may groups (each piece of data is in one group)
* One of my groups is my default group (the one that data goes into)

u - if you are the owner

g - I am not the owner but a member of the data’s group

o -I am not the owner nor do I belong to a data’s group

-(file or d for dir)|r –x (u)|rwx (g)|r - - (o) 1 (ownder) (group #) ….. file1

R – read (ie cat the file)

W – write (ie change file content)

X – execute (ie. attempt to execute it as a program)

R – reading a dir which is an ls cmd

W – create, delete, rename contents in a directory

X – required (search permission – to go into the Inode)

Remember: to delete a file you do not need its permissions but only permissions to the directory

Every dir can find itself . and its parent ..

|  |  |  |
| --- | --- | --- |
| Operation | Perm to dir | Perm to file |
| Ls dir | R |  |
| Ls –l dir | Rx |  |
| Ls –l dir/file | X |  |
| Cat dir/file | X | R |
| Create new file in dir | Wx |  |
| Cat > dir/file | X | W |
| Rm dir/file | wx |  |
| Mv dir/file | Wx |  |

Step1: check owner, check if in group, if not those then you are in other!

Use ls –l 🡨 will list permissions

Cp

1. Does file exist?

Yes.

|  |  |  |  |
| --- | --- | --- | --- |
| Dir | File | Dir | File1 |
| X | R | X | w |

No.

|  |  |  |  |
| --- | --- | --- | --- |
| Dir | File | Dir | File1 |
| X | R | XW |  |

$chmod to change permissions; $chgrp to change group $chown to change owner

Symbolic Mode

Chmod [ugo][+ - =][rwx] filename or dir

Eg. u+r,g-w; a=r; a= ; gu+w; o+r \*; go-w \*

Absolute Mode

Rwx = 421

When you create data: permissions, ownder, group set to:

The owner is who created it; group is your default group

Check your info do $id

$umask (only takes away permissions)

Find cmd

$find dir/file –name “ “ –type[d or f]

$find . (everything recursively in the dir); use “ “ so it can read wildcards

-The r permission for dir governs if you can list the objects in it; you need r to have access to the names of objects

-to get more info you will also need to search the dir, which requires the (x) permission; also needed if you want to use options (ie. –l cmd or other flags)

-w permission for dir governs if you can update the dir (create, delete, rename/mv, objects)

-if you do not have the (x) perm for dir you cannot pass through it, making its subdirs inaccessible; you also need (x) perm to access the attributes (owner, group, perm) of an object in the dir