

Using Linux Commands

Lab 5

Creating files and directories

- Use the command mkdir to **create a new** directory to the **current** working directory.
- **\$ mkdir directoryName**

Creating files and directories

1. Go to your home directory. Type ***cd***.
2. Make sure that you got to your home directory, type ***pwd***
3. Create a new directory called **test** in your home directory:

```
$ mkdir test
```

4. Check the permissions of the directory by typing:

```
$ ls -ld test
```

```
drwxr-xr-x  2  chris   sales   1024   May 18  21:30  test
```

5. Suppose that you want to prevent everyone else from using or viewing the files in this directory:

```
$ chmod 700 test
```

6. Make the test directory your current directory :

```
$ cd test
```

Creating files and directories

- **To create a directory(test) in the Desktop, we have two ways:**

- ❑ `mkdir /home/chris/Desktop/test`
- ❑ `cd /home/chris/Desktop`
`mkdir test`

- **NOTE:**

`$ ls -ld test`

provide a long listing of the test directory, without showing the contents of the test directory.

- ❑ The -d option tells ls not to list the contents of the test directory; just show us the listing for the directory itself.

Moving, copying, and deleting files

- The command **mv** moves or renames files.

- The simplest form of use is:

mv oldfilename newfilename.

it will rename the file a.kwd to a new name b.kwd

```
$ mv a.kwd b.kwd → Renaming
```

- It will also move a file to a directory. If you use it in the form:

mv filename directoryname

it will move the file into the named directory keeping its old name

```
$ mv a.kwd /mnt/collection1
```

```
$ mv /mnt/collection1/ a.kwd ~
```

Moving, copying, and deleting files

Copying files is similar to moving them but you use `cp` instead of `mv`

`cp filename directoryname`

```
$ cp /mnt/collection1/ a.kwd /home/ahmed/Desktop
```

```
$ cp a.kwd /mnt/collection1
```

```
$ cp /mnt/collection1/ a.kwd ~
```

To remove a file from the current directory

`rm filename`

```
$ rm a.kwd
```

```
$ rm *
```

* → remove all files in the current directory

Create empty files



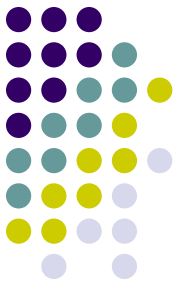
- To create an empty file in the current directory use the command **touch**
- **\$ touch file1 file2 file3**

```
$ touch apple banana grape grapefruit watermelon
```

```
$ ls -l
```

Using file-matching metacharacters

- ***Metacharacters*** help to match one or more files without typing each filename completely.
 - ***** This matches any number of characters(zero or more characters).
 - **?** This matches any one(single) character.
 - **[...]** This matches any one of the characters between the brackets, which can include a dash-separated rang of letters or numbers.



Using file-matching metacharacters

This matches any number of characters *

\$ ls a*

apple

\$ ls g*

grape grapefruit

\$ ls g*t

grapefruit

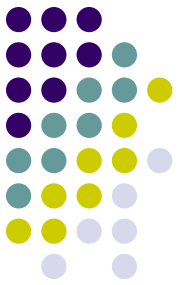
\$ ls *e*

apple grape grapefruit watermelon

\$ ls *n*

banana watermelon

Using file-matching metacharacters



This matches any one(single) character ?

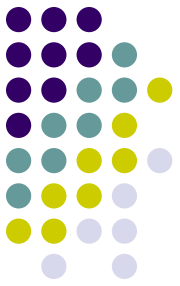
```
$ ls ???e
```

```
Apple  grape
```

```
$ ls g???e*
```

```
grape  grapefruit
```

Using file-matching metacharacters



This matches any one of the characters between the brackets [...]

```
$ ls [abw]*
```

```
apple banana watermelon
```

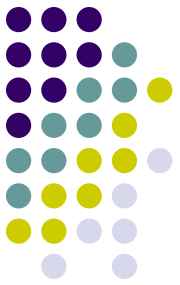
```
$ ls [agw] * [ne]
```

```
apple grape watermelon
```

```
$ ls [a-g] *
```

```
apple banana grape grapefruit
```

Using file-redirection metacharacters



- *< Direct the contents of a file to the command*

```
$ mail root < ~/.bashrc
```

the contents of the .bashrc file in the home directory are sent in a mail message to the computer's root user.

- *> Direct the output of a command to a file, overwriting any existing file*

```
$ echo "I finished the project on $(date)" > ~/projects
```

- *>> Direct the output of a command to a file, adding the output to the end of existing file*

```
$ echo "I finished the project on $(date)" >> ~/projects
```