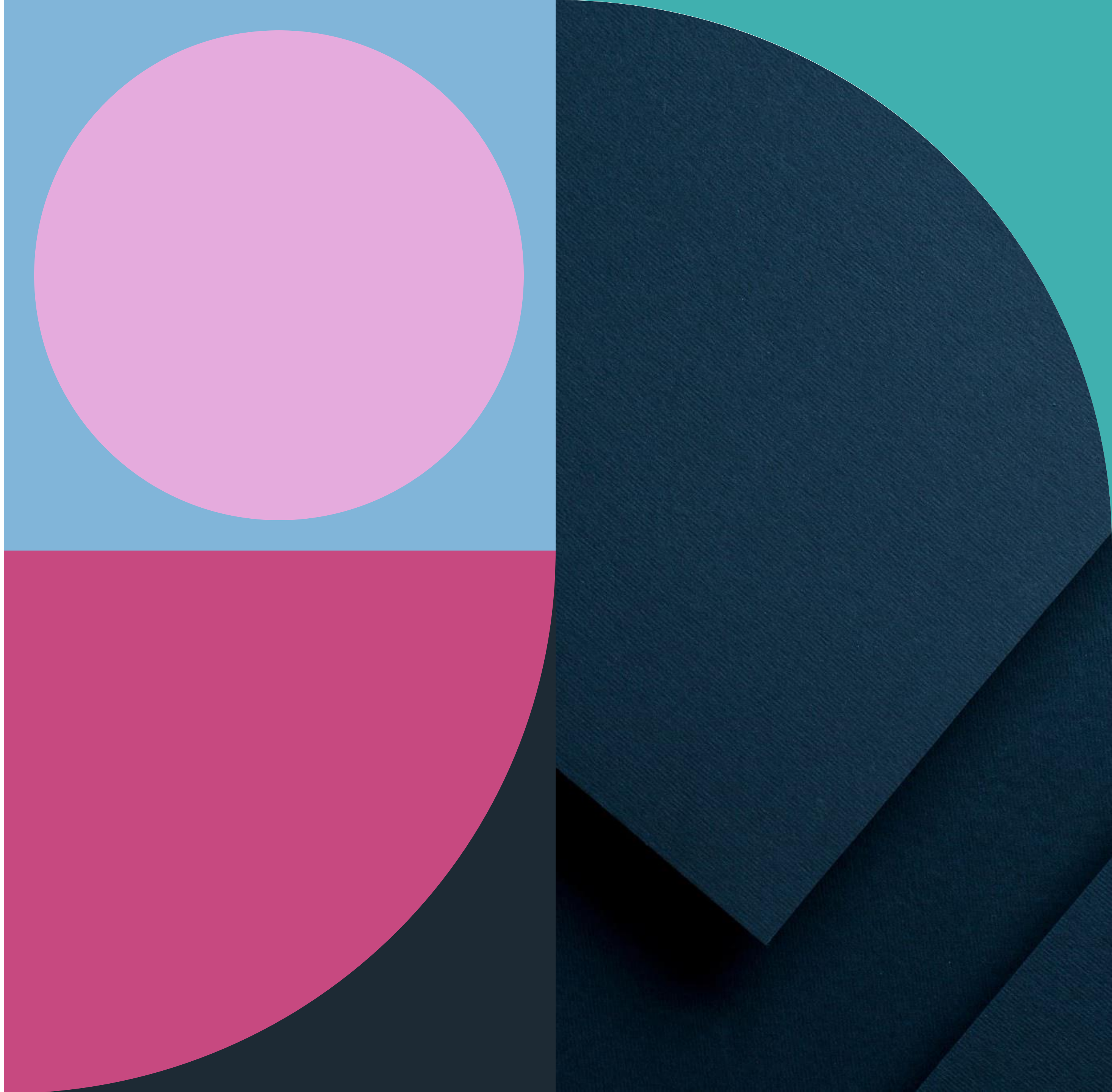
Software Systems Development

Lab 2 - Bash



Commands Covered in Lab-2

- ls
- head
- tail
- link
- diff
- diff3
- cmp
- comm
- file
- lpr
- awk

ls

- List information about files.

Syntax :

ls [*Options*]... [*File*]...

Useful Flags :

-a : List all entries including files starting with ' . '

-l : Use a long listing format.

-h : Print sizes in human readable format

head

- Output the first part of files, prints the first part (10 lines by default) of each file.

Syntax :

head [*options*]...[*file*]…

Useful Flags :

-n : Output first ' N ' lines.

-*NUMBER* : Return the first *NUMBER* of lines from the file..

-v : Print file name headers also.

tail

- Output the last part of files, print the last part (10 lines by default) of each FILE.

Syntax :

tail [*options*]... [*file*]...

Useful Flags :

-n : Output last ' N ' lines.

-c : Output the last K bytes.

diff

- Display the differences between two files, or each corresponding file in two directories.
- It yields a list of modifications that require to be made in the first file to match the second file

Syntax :

diff [*options*] *FILES*

Useful Flags :

-a : Treat all files as text.

-b : Ignore changes in the amount of white space.

-i : Ignore case differences in file contents.

diff3

- Show differences among three files.
- It yields a list of modifications that require to be made in the first file to match the second file

Syntax :

diff [*options*] *FILES*

Useful Flags :

-a : Treat all files as text.

-b : Ignore changes in the amount of white space.

-i : Ignore case differences in file contents.

cmp

- Compare two files, and if they differ, tells the first byte and line number where they differ.
- It yields a list of modifications that require to be made in the first file to match the second file

Syntax :

cmp *options... FromFile* [*ToFile*]

Useful Flags :

-c : Print the differing characters.

-l : Print the (decimal) offsets and (octal) values of all differing bytes.

-s : Only return an exit status indicating whether the files differ..

comm

- Compare two sorted files line by line. Output the lines that are common, plus the lines that are unique.

Syntax :

comm [*options*]... *File1 File2*

Useful Flags :

-1 : Suppress lines unique to *file1*.

-2 : Suppress lines unique to *file2*.

-3 : Suppress lines that appear in both files.

file

- Determine file type.

Syntax :

```
file [option] [filename]
```

Useful Flags :

-b : Do not prepend filenames to output lines (brief mode).

-F : Use the specified string as the separator between the filename and the file result returned. Defaults to ':' .

lpr

- Print files. Send a print job to the default system queue.

Syntax :

```
lpr [-Pprinter] [-#num] [-C class] [-J job] [-T title] [-U user] [-i [numcols]]
    [-1234 font] [-wnum] [-cdfghlnmprstv] [name ...]
```

link

- Create a link to a file

Syntax :

link *FILE1 FILE2*

Useful Flags :

--help : Display Help.

--version : Output version information .

cut

- Divide a file into several parts (columns)

Syntax :

cut [*OPTION*]... [*FILE*]...

awk

- Awk is a basic programming language.
- Awk sees input as an array.
- When awk scans over a text file, it treats each line as a record.
- Each record is broken into fields.
- Tracks using NR (number of records) and NF (number of fields) built-in variables.
- Format : pattern or keyword{ actions }

awk (Cont.)

- $ awk '{print}' hamlet.txt
- $ awk '{print $NF}' geeks.txt hamlet.txt
- $awk '{print NR,$0}' hamlet.txt
- $awk '{ if($5 == "of") print $0;}' hamlet.txt
- $awk 'BEGIN { for(i=1;i<=6;i++) print "square of", i, "is",i*i; }'

awk (Cont.)

- Functions in AWK :

```
$ awk 'BEGIN{print substr("International Institute of Information and Technology", 9, 8)}'
```

```
$ awk 'BEGIN{print index("International ", "na"); print index("Technology", "is")}'
```

```
$ awk 'BEGIN{print length("Hello World")}'
```

Thank You!!