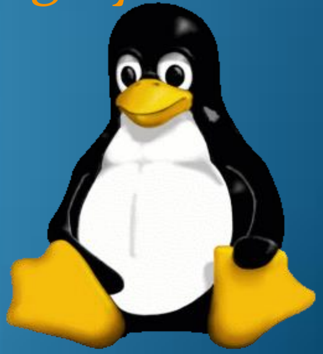
Operating System
lab2

Lab1: Introduction to Shell Scripts
Instructor: Eng\ Samar Shaaban
E-mail: ssa10@fayoum.edu.eg
Email subject: **fci-os-2021**
Github Repo: https://github.com/SamarShabanCS/Operating-Sytem-2021
Slack workspace: https://fu-fci-os-2021.slack.com



Agenda

- Linux Kernel
- Linux Shell
- Basic Linux Commands
- File Permissions
- Shell Scripts.

Linux Kernel

- The **Linux kernel** is the operating system kernel used by the Linux family of Unix-like systems. It is one of the most prominent examples of free and open source software.

Linux Shell

- *Shell is a user program or its environment provided for user interaction. Shell is a command language interpreter that executes commands read from the standard input device (keyboard) or from a file.*

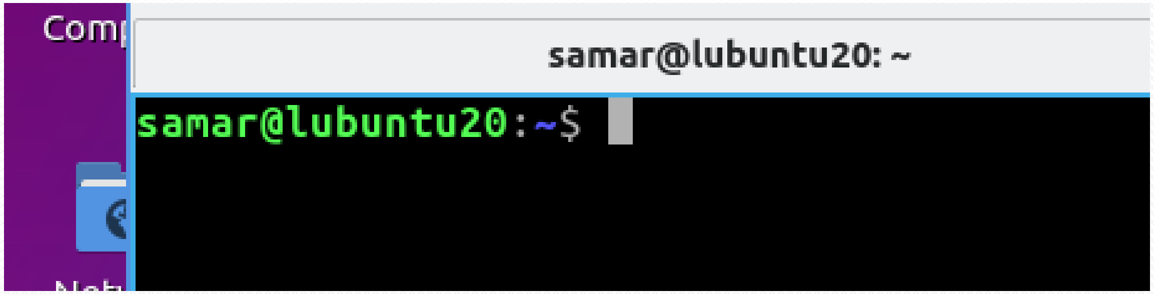
- *Shell is not part of system kernel, but uses the system kernel to execute programs, create files etc.*

- *There are several shells available like Bourne shell, C shell, Korn shell, etc. Each shell differs from the other in Command interpretation. The most popular shell is bash.*

Linux Shell

- ***shell prompt***

➢ a character at the start of the command line which indicates that the shell is ready to receive the commands. The character is usually a '%' (percentage sign) or a $(dollar sign).



```
samar@lubuntu20:~$
```

Linux Shell

- Linux commands are executable binary files located in directories with the name bin (for binary). Many of the commands that are generally used are located in the directory /usr/bin.

- Environment variables: Shell has built in variables which are called environment variables. For e.g. the user who has logged **$USER**

- When the command name is entered, the shell checks for the location of the command in each directory in the **PATH** environment variable. If the command is found in any of the directories mentioned in PATH, then it will execute. If not found, will give a message <u>Command not found</u>.

Basic Linux Commands

File Permissions

- Access permissions define which users have permission to access a file or directory. Permissions are three types, read, write and execute.

- **Chmod command**.
  - ❑ *Permissions*
    - *u - User who owns the file.*
    - *g - Group that owns the file.*
    - *o - Other.*
    - *a - All.*
    - *r - Read the file.*
    - *w - Write or edit the file.*
    - *x - Execute or run the file as a program.*

Permission examples

chmod g+r file	Allow group to read
chmod u+w file	Allow user to write
chmod a+x file	Allow everyone (user, group, and other) to execute
chmod o-r file	Disallow others to read
chmod ug+r file	Disallow others to read
chmod g+w,o-r file	Allow group write, disallow other read

- **Octal numerical representation**

r	**4**
w	**2**
x	**1**

- For example: converting rwxr-x--- to octal:
  7 = 4+2+1 = r + w + x
  5 = 4+1 = r + x (not write)
  0 = no rights
  So rwxr-x--- is 750 in octal.
- Octal values can be given to chmod, so chmod 750 file

Shell Scripts

- *Shell scripts* are short *programs* that are written in a *shell programming language* and *interpreted* by a *shell process.*

- A feature of bash and other shells used on Unix-like operating systems is that each contains a built-in programming language, referred to as a shell programming language or *shell scripting language*, which is used to create shell scripts.

- **A First Script**
  - The script clears the monitor screen of all previous lines and then writes the text ”*Hello World.*” on it.

```
#!/bin/bash
clear
echo “Hello World."
```

  - Save the text file with ***.sh***

- *After writing shell script set execute permission for your script using syntax:*

  ```
  chmod permission your-script-name
  ```

  - ❑ *Examples:*

    ```
    $ chmod +x your-script-name
    $ chmod 755 your-script-name
    ```

- *Execute your script using syntax:*

  ```
  bash your-script-name
  sh your-script-name
  ./your-script-name
  ```

  - ❑ *Examples:*

    ```
    $ bash bar
    $ sh bar
    $ ./bar
    ```