

**Tishk International University**  
**Science Faculty**  
**IT Department**



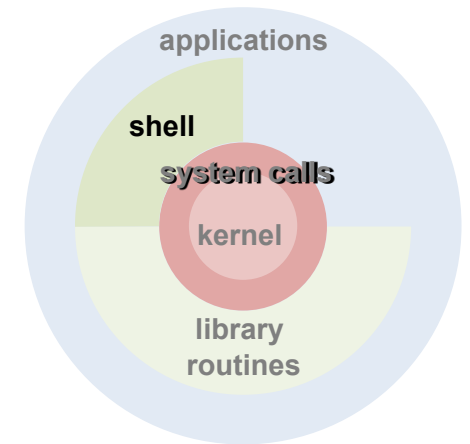
**Open Source OS (Linux)**

**Lecture02: Basic Commands**

**4th Grade -Fall Semester 2021-2022**

**Instructor: Alaa Ghazi**

# The shell



- **Command Interpreter:**
  - Same privileges as other program.
  - Multiple interpreters available: sh, csh, ksh, tcsh, **bash**...
  - Responds with the prompt: test@si:~\$ (normal account:\$, root account:#).
- **Session (login + passwd):**
  - Local Access:
  - Remote access: through network (telnet, rlogin, ssh...).

# The shell

- **Shell Types:**
  - Bourne shell “**sh**” (/bin/sh): old UNIX syntax (SysV).
  - C shell “**csh**” (/bin/csh): C-like syntax (BSD).
  - Bourne Again shell “**bash**” (/bin/bash): Similar to its antecessor, but extended with many features from csh.
  - Tcsh “**tcsh**” (/bin/tcsh): improved version of the original C shell.
  - In general, differences are not relevant for day-to-day use.
- **Shell Goal:** interactive dialog between user and system:
  - Through a huge amount of orders/commands and applications:

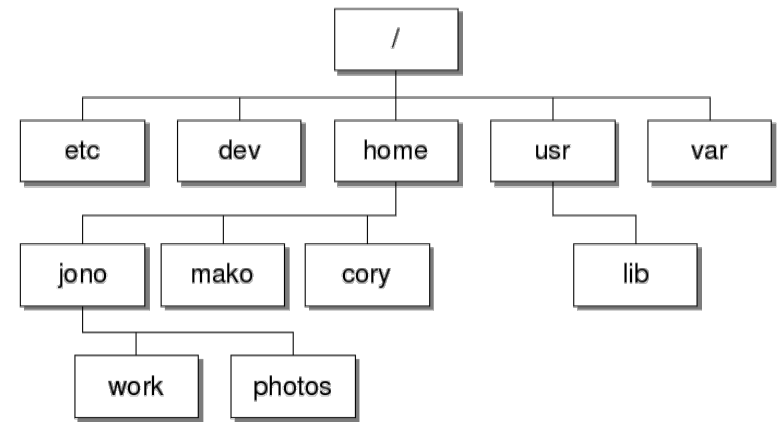
# The shell

- **Command structure:**

```
user@machine:~$ command -<options> [arguments]
```

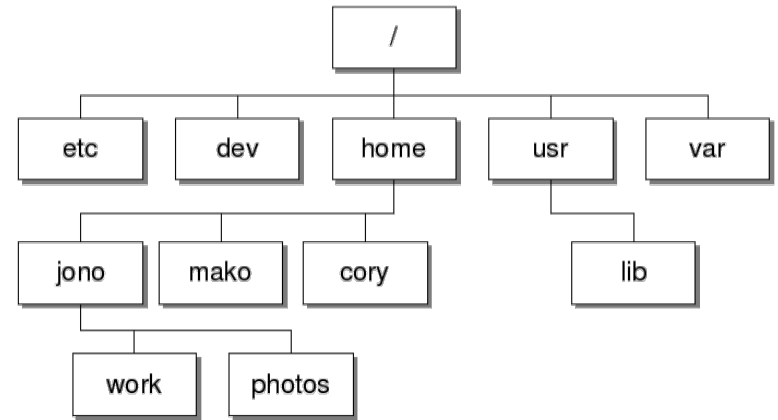
- **Options:** command pieces that modify the initial behavior.
  - **Arguments:** file name or any other kind of data needed by the command.
- **Man command (formats and displays manual pages):**
    - First command to learn. Displays on screen information about a command, programming function, configuration file, etc.
    - Syntax: `$ man -<options> [command]`:
      - **-a:** display all the manual pages that match “command”, not just the first one.
      - **-K:** search for the specified string in all man pages.

# File System



- Definition:
  - Logic structures and their corresponding methods employed by the Operating System to organize the files in the disk.
- Tree-like **Hierarchical** structure:
  - Efficient management of information (group related info into folders).
  - Folders separated by /
  - File access (path):
    - Absolute: `cd /home/pepe.`
    - Relative to current path (with “.” o “..”): `cd ../../../../usr/local.`
- Files starting with “.” are “hidden”.
- Security: protection of files against unauthorized accesses.

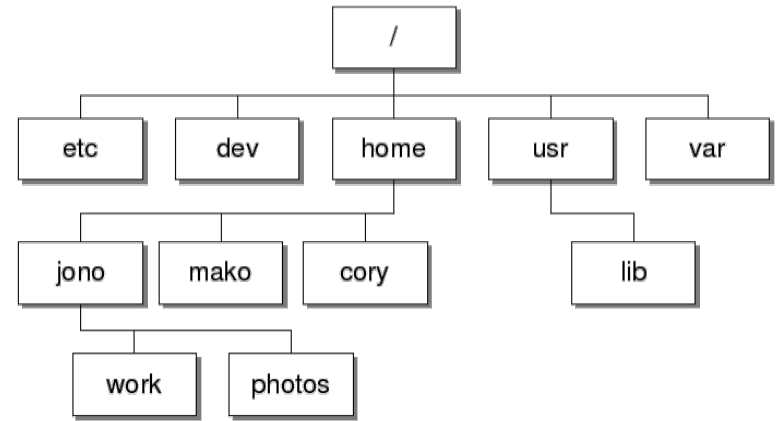
# File System



- Unit Mounting:
  - A storage device (usb, cd, etc.) can be associated with a particular position in the directory tree.
- Same treatment to files and I/O devices:
  - Same program can employ files and/or devices indifferently.
- Different locations of the file tree can be linked (**ln** command).
- Definition of a folder/file **path**:
  - Directories to be traversed, starting from root directory, in order to reach that folder/file.

# File System

<b>/</b>	Root directory.
<b>/bin</b>	Core operating system commands.
<b>/boot</b>	Kernel and files needed to load the kernel.
<b>/dev</b>	Device entries for disks, printers, pseudo-terminals, etc.
<b>/etc</b>	Critical startup and configuration files.
<b>/mnt</b>	Temporary mount points, mounts for removable media.
<b>/lib</b>	Libraries, shared libraries and parts of the C compiler.
<b>/home</b>	Default home directories for users.
<b>/opt</b>	Optional software packages (not consistently used).
<b>/root</b>	Home directory for the superuser.
<b>/sbin</b>	Command needed for minimal system operability.
<b>/proc</b>	Information about all running processes.
<b>/tmp</b>	Temporary files.
<b>/usr</b>	Hierarchy of secondary files and commands.
<b>/usr/local</b>	Software installed by users.
<b>/var</b>	System specific data and configuration files.



# File System (Commands)

- Large amount of shell command to interact with FS.
- Navigating through the file system:
  - Command **pwd**: displays current.
  - Command **cd**: change to a different directory.
  - Command **mkdir**: create a new folder.
- File Manipulation:
  - Command **ls**: list folder contents in alphabetical order.
  - Command **cp**: copy files.
  - Command **mv**: move files (or rename).
  - Command **rm**: remove files or folders.



# File System (Commands)

- File Manipulation (cont.):
  - Command **ln**: create a link between two files.
  - Command **whereis**: locate the path of a cmd's binary/src code/manual.
  - Commands **locate**/**find**: locate a file in the directory tree.
- File Contents:
  - Commands **cat**/**more**/**less**: show the contents of a file.
  - Command **wc**: count the number of bytes/words/lines in a file.
  - Commands **head**/**tail**: display in stdout the first/last lines of a file.
  - Command **grep**: display the lines of a file that match a text pattern.
  - Command **tar**: add the contents of a file tree to a single file.
  - Command **cut**: remove specific sections of each line of a file.
  - Command **sort**: arrange file lines in specific order (alphabetical).
  - Command **vi**: text editor in the terminal (present in every UNIX system).

# User Management (Commands)

- Detailed description in the APPENDIX.
- Basic user management:
  - Command **whoami**: displays username.
  - Command **who**: shows users connected to the system.
  - Command **passwd**: change user password.
  - Command **useradd**: adds a user in the system.
  - Command **userdel**: deletes a user from the system.
- File Permission management:
  - Command **chmod**: modify file or directory permissions.
  - Command **chown/chgrp**: modify UID/GID of a file.
  - command **umask**: modifies default permissions assigned to new files.

# Environment Variables

- Group of shell session variables with a pre-defined value. Their value is obtained this way: `$ echo $VARIABLE`.
- Allow the configuration of certain aspects in the cmd interpreter.
- Two kinds:
  - **User variables:** internal to our shell session:
    - Can be listed with command `env`.
  - **System variables:** common to every shell and other programs and users:
    - Can be listed with command `set`.

# Environment Variables

- Some important internal variables:
  - **\$PATH**: indicates which are the directories where binaries can be found. Before executing a command, the shell searches in those directories.
  - **\$HOME**: root directory of current user.
  - **\$TERM**: kind of terminal we are employing to connect to the system.
  - **\$SHELL**: user shell. Ex. /bin/bash.
  - **\$TZ**: time zone. Has an influence on the timing format returned by date command. Any change in our files adjusts to the time zone specified by that variable.