IT WORKSHOP
GXESL 208

Lab Worksheet

Semester 2
Bachelor of Technology
Applied Electronics & Instrumentation Engineering
Electronics & Communication Engineering
College of Engineering
Trivandrum

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
COLLEGE OF ENGINEERING TRIVANDRUM
KERALA

2025

List of Experiments

Syllabus

Preface

APJ Abdul Kalam Technological University, Kerala has introduced the course `IT Workshop (GXESL 208)` in first year of the B.Tech Degree programme. This worksheet has been prepared for use in the workshop sessions of the course at Department of Electronics & Communication Engineering, College of Engineering Trivandrum.

These worksheets are in the process of active development and updated versions may be found on https://jim79.github.io/it-workshop/. Feedback/suggestions may be shared via email at *arun.varghese@cet.ac.in* or *joaquim@cet.ac.in*

Dr. Arun Varghese
Joaquim Ignatious Monteiro

Experiment 1

Lab Safety

This section discusses safety in the lab. Safety guidelines help protect individuals from accidents and injury. They also help to protect equipment from damage.

General Safety

Safe working conditions help prevent injury to people and damage to computer equipment. A safe workspace is clean, organized, and properly lighted. Everyone must understand and follow safety procedures.

Follow the basic safety guidelines to prevent cuts, burns, electrical shock, and damage to eyesight. As a best practice, make sure that a fire extinguisher and first-aid kit are available in case of fire or injury. Poorly placed or unsecured cables can cause tripping hazards in a network installation. Cables should be installed in conduit or cable trays to prevent hazards.

This is a partial list of basic safety precautions to use when working on a computer:

- Remove your watch and jewelry and secure loose clothing.
- Turn off the power and unplug equipment before performing service.
- Cover sharp edges inside the computer case with tape.
- Never open a power supply or a CRT monitor.
- Do not touch areas in printers that are hot or that use high voltage.
- Know where the fire extinguisher is located and how to use it.
- Keep food and drinks out of your workspace.
- Keep your workspace clean and free of clutter.
- Bend your knees when lifting heavy objects to avoid injuring your back.

Electrical Safety

Follow electrical safety guidelines to prevent electrical fires, injuries, and fatalities in the home and the workplace. Power supplies and CRT monitors contain high voltage.

Fire Safety

Follow fire safety guidelines to protect lives, structures, and equipment. To avoid an electrical shock and to prevent damage to the computer, turn off and unplug the computer before beginning a repair.

Fire can spread rapidly and be very costly. Proper use of a fire extinguisher can prevent a small fire from getting out of control. When working with computer components, be aware of the possibility of an accidental fire and know how to react. Be alert for odors emitting from computers and electronic devices. When electronic components overheat or short out, they emit a burning odor.
If there is a fire, follow these safety procedures:

- Never fight a fire that is out of control or not contained.
- Always have a planned fire escape route before beginning any work.
- Get out of the building quickly.
- Contact emergency services for help.
- Locate and read the instructions on the fire extinguishers in your workplace before you have to use them.

Experiment 2

Computer Hardware Familiarization

JOAQUIM IGNATIOUS MONTEIRO

Instructions

1. All students should refer to the e-resources shared on this web link https://jim79.github.io/it-workshop/ and come prepared for the workshop session.
2. All Students should write all the questions from worksheet in the rough record before coming to the workshop session.
3. Answers to the above questions are to be found out during the workshop session and written in the rough record.
4. Rough record has to be completed during the workshop session.
5. Student should get the rough record verified and signed by the faculty before leaving the lab.

Hardware Familiarization exercises

Students are to complete the tasks in following sections to familiarize with computer hardware.

2.1 CPU Box (Chassis)

CPU Box is the enclosure that houses all the computer components, providing protection and structural support.

Tasks:

1. List the typical materials used for constructing a CPU chassis.
2. Identify the form factor of the CPU case provided to you.
3. Identify the types of form factors supported by different chassis.
4. List the cooling mechanisms commonly integrated into a CPU chassis.

2.2 Motherboard

A motherboard is the main circuit board in a computer system. It connects all of the internal components, like the memory, processor, graphics card and other hardware. It also provides power to each component and allows them to communicate with each other.

Tasks:

1. Identify your motherboard model.
   Tip: Motherboard model is usually printed on the motherboard. It can be located in several possible locations; for example, it may be printed near the RAM slots, near the CPU socket, or between the PCI slots.

2. Google your motherboard model number and list the technical specifications of motherboard, including form factor, chipset, and socket type.

3. Identify the types and number of RAM slots available.

4. Describe the input/output ports provided on the motherboard.

5. Identify components on the motherboard provided and draw a layout (similiar to the figure given below) in your rough record.

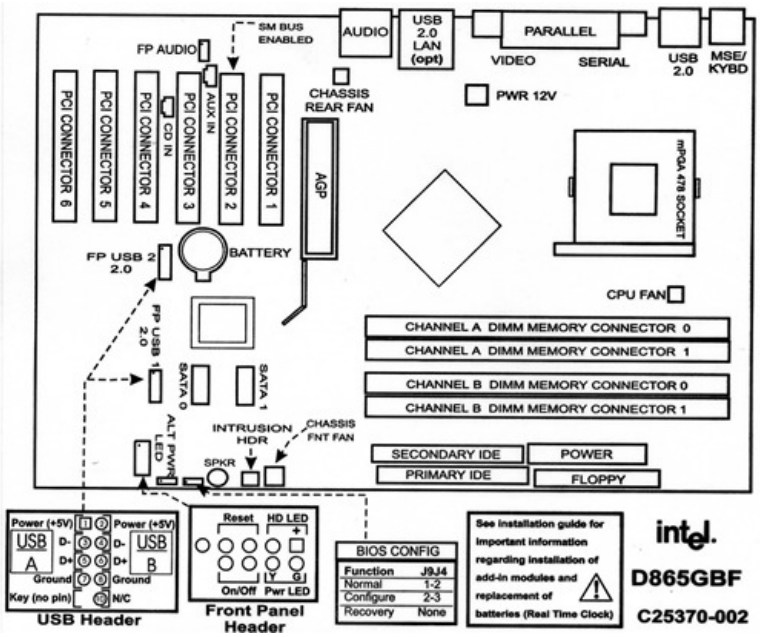

Figure 2.1:

6. Refer to the website https://motherboarddb.com/motherboards/ choose a motherboard manufactured after the year 2010 and list the following features :

- Manufacturer
- Year of manufacture
- Form factor
- Chipset
- Memory
- Number and type of USB ports
- Video outputs
- Network ports
- Audio ports
- Audio chipset
- Expansion slots
- Power connectors

2.3 CPU and Chipset

The Central Processing Unit (CPU) performs most of the processing inside a computer, while the chipset manages data flow between the processor, memory, and peripherals.

Tasks:

1. List the key specifications of a CPU, including clock speed, core count, and cache size.

2. Explain the terms power consumption and thermal design power (TDP) of a CPU.

3. Refer to the website https://www.techpowerup.com/cpu-specs/ choose a CPU manufactured after the year 2010 and list the following features :

    - Manufacturer
    - Year of manufacture
    - Socket
    - Process Size
    - Frequency
    - Number of Cores
    - Cache
    - Memory Support
    - TDP
    - Minimum Power
    - Market
    - Production Status

2.4 Storage Devices

2.4.1 Hard Disk Drive

Hard Disk Drive (HDD) is an electro-mechanical data storage device that stores and retrieves digital data using magnetic storage with one or more rigid rapidly rotating platters coated with magnetic material. HDDs are a type of non-volatile storage, retaining stored data when powered off. Modern HDDs are typically in the form of a small rectangular box.

Tasks:

1. List the technical specifications of the hard disk provided to you including storage capacity, RPM, and interface type.
2. Explain the differences between SATA, SAS, and IDE hard disks.
3. Refer to the website https://smarthdd.com/database/ choose a Hard Disk Drive (HDD) and list the following features :
    - Manufacturer
    - Model
    - Capacity
    - Interface
    - Maximum interface speed
    - Maximum read speed

2.4.2 Solid State Drive

A solid-state drive (SSD) is a type of solid-state storage device that uses integrated circuits to store data persistently.

1. Refer to the website https://smarthdd.com/database/ choose a Solid State Drive (SSD) and list the following features :
    - Manufacturer
    - Model
    - Capacity
    - Interface
    - Maximum interface speed
    - Maximum read speed
2. Compare the features of the HDD's and SSD's.

2.5 Interface Cards

Expansion cards like graphics cards, sound cards, and network cards provide additional functionalities to the computer.

Tasks:

1. List the technical specifications of a graphics card, including VRAM, clock speed.
2. Identify the ports available on interface cards, such as HDMI, DisplayPort, or audio jacks.
3. Describe the cooling mechanisms for interface cards.
4. Refer to the website https://www.techpowerup.com/gpu-specs/ choose a GPU manufactured in the year 2003 and list the following features :
    - Manufacturer
    - Year of manufacture
    - GPU Name
    - GPU Clock
    - Memory Size
    - Graphics Features
    - Bus Interface
    - Production Status
5. Refer to the website https://www.techpowerup.com/gpu-specs/ choose a GPU manufactured after the year 2020 and list the following features :
    - Manufacturer
    - Year of manufacture
    - GPU Name
    - GPU Clock
    - Memory Size
    - Graphics Features
    - Bus Interface
    - Production Status

2.6 Card Slots

Slots on the motherboard such as PCI and PCIe where interface cards are inserted.

Tasks:

1. List the types of card slots available on a motherboard and their respective uses.

2.7 Cables

Various cables like SATA, IDE, and power cables that connect internal components and peripherals.

Tasks:

1. Refer to the website https://www.cablestogo.com/learning/connector-guides/internal and list the types of cables used in a computer and their specific purposes.
2. Describe the data transfer rates of different generations of SATA cables.

2.8 SMPS (Switch Mode Power Supply/PSU)

Converts AC power from the mains to DC power used by the computer's internal components.

Tasks:

1. List the power ratings and efficiency certifications of SMPS units.
2. Identify the types of connectors provided by an SMPS for various components.
3. Describe the cooling mechanisms and protections (e.g., overvoltage) in an SMPS.
4. Refer to the website https://www.cybenetics.com/index.php?option=power-supplies choose a Power Supply Unit (PSU) and list the following features :
    - Manufacturer
    - Form factor
    - Wattage
    - Efficiency rating

2.9 NIC (Network Interface Card)

Allows the computer to connect to a network and communicate with other devices.

Tasks:

1. List the technical specifications of a NIC, including speed and connection type.
2. Describe the difference between wired and wireless NICs.
3. Refer to the website https://www.scan.co.uk/shop/computer-hardware/network-cards-accessories/rj45-network-cards choose a network card and list the following features :

- Manufacturer
- Interface
- Supported data rates

2.10 Various Ports

Description: Includes USB, HDMI, Ethernet, and audio ports used to connect external devices.

Tasks:

1. Refer to the website https://newnex.com/usb-connector-type-guide.php and list the technical specifications of USB ports, including version and data rates.
2. Refer to the website https://www.xenarc.com/different-types-of-monitor-ports.html and list the common display ports in a computer and their typical applications.

2.11 I/O Devices

Description: Input devices like keyboards and mice, and output devices like monitors and printers.

Tasks:

1. List the specifications of common input devices, such as DPI for mice or key travel for keyboards.
2. Refer to the website https://www.displaydb.com/brands and list the following features of computer monitors :
    - Brand
    - Model
    - Size
    - Panel
    - Refresh rate
    - Screen aspect ratio
    - Screen resolution
3. List the connectivity options available for computer printers.

2.12 Buses

Description: Electrical pathways like data bus, address bus, and control bus that transfer data between components.

Tasks:

1. Explain the function of the address bus, data bus and control bus in a computer system.

2.13 Firmware

Description: Software programmed into read-only memory, providing low-level control for the device's hardware.

Tasks:

1. List the features of BIOS and UEFI firmware.
2. Explain how firmware updates improve hardware functionality.
3. Describe the role of firmware in initializing hardware during startup.

Experiment 3

Familiarizing basic Unix/Linux commands

DR.ARUN VARGHESE

Instructions:

- Each command in this handout are to by typed at the terminal (not copy-pasted)
- Write down each command and its output(shorten, if needed) in your rough record
- You will be asked to submit your rough record and obtain signature from faculty

The Shell

- The shell is a program that takes keyboard commands and passes them to the operating system to carry out.
- Almost all Linux distributions supply a shell program from the GNU Project called `bash`.
- The name is an acronym for **bourne-again shell**, a reference to the fact that `bash` is an enhanced replacement for `sh`, the original Unix shell program.
- We interact with the shell through the terminal.

3.1 pwd

- Print Working Directory: displays the current working directory
- When we first log in to our system (or start a terminal emulator session), our current working directory is set to our home directory.
- Each user account is given its own home directory, and it is the only place a regular user is allowed to write files.

File System Hierarchy

- Like Windows, Linux organizes its files in a hierarchical directory structure.
- The first directory in the file system is called the root directory (denoted by a /).
- Linux always has a single file system tree. Storage devices are attached (or more correctly, mounted) at various points on the tree

3.2 ls

- List the contents of of a directory
- List the contents of the root directory: `ls /`
- Listing the contents of the current directory: `ls`
- We can specify the directory to list: `ls /usr`
- We can specify multiple directories: `ls ~ /usr`
- The ~ represents the home directory.
- Use the `-l` option to reveal more detail: `ls -l /usr`
- `ls` can also be used to know details of files: `ls -l filename`
- Example output:

```
-rw-r-r- 1 user1 workshop 573 Sep 29 22:00 foo
```

- `-t` option sorts by time modified: `ls -t`
- We can combine options: `ls -lt`
- `-a` option lists all files, even those with names that begin with a `.` (period), which are normally hidden: `ls -a`
- The `ls` command has many other options

3.3 cd

- Change our working directory:
- `cd <pathname>`
- We can specify pathnames in one of two different ways: as absolute pathnames or as relative pathnames.

- Absolute pathnames begin with the root directory (represented by `/`):For example, `cd /usr/bin; pwd`
- A relative pathname starts from the working directory.
- Special notations to represent relative positions:
  - `.` refers to the current directory
  - `..` refers to the parent directory
- From `/usr/bin` change to `/usr`: `cd ..`
- Change back to `/usr/bin`: `cd ./bin`
- We can omit the `./`: `cd bin`

Exercise:

1. `cd` to `/sys/bus/cpu/devices`
2. List the contents of the folder using `ls`
3. From `devices`, `cd` to `/sys/bus/cpu` using relative path
4. From `cpu`, list the contents of `/sys/devices` using relative path
5. Change back to your home directory

3.4 file

- Print a brief description of a file's contents: `file filename`
- `file /usr/bin/apt`

There are many kinds of files. In fact, one of the common ideas in Unix-like operating systems such as Linux is that "everything is a file".

3.5 less

- View text file's contents: `less filename`
- `less /etc/passwd`
- To exit less, press q

The less program was designed as an improved replacement of an earlier Unix program called more. The name less is a play on the phrase "less is more".

3.6 mkdir and rmdir

- Create and remove directories
- `mkdir directory...`
- `mkdir dir1`: Create the directory `dir1`.
- `mkdir dir1 dir2 dir3`: Create directories `dir1`, `dir2`, and `dir3`.
- **rmdir** : deletes an empty directory.

Exercise:

1. Create a folder named `ec` in your home folder.
2. Under `ec`, create four folders named `y1`, `y2`, `y3`, `y4` without changing from your home directory.
3. Under `y1`, create folders `ec1` and `ec2`.
4. Verify your folder structure using the command: `tree ec`.
5. Delete all folders created.

3.7 cp

Copy files and folders

Command	Results
cp file1 file2	Copy file1 to file2. If file2 exists, it is overwritten with the contents of file1. If file2 does not exist, it is created.
cp -i file1 file2	Same as previous command, except that if file2 exists, the user is prompted before it is overwritten.
cp file1 file2 dir1	Copy file1 and file2 into directory dir1. The directory dir1 must already exist.
cp dir1/* dir2	Using a wildcard, copy all the files in dir1 into dir2. The directory dir2 must already exist.
cp -r dir1 dir2	Copy the contents of directory dir1 to directory dir2. If directory dir2 does not exist, it is created and, after the copy, will contain the same contents as directory dir1. If directory dir2 does exist, then directory dir1 (and its contents) will be copied into dir2.

3.8 mv

Move and rename files

Command	Results
mv file1 file2	Move `file1` to `file2`. If `file2` exists, it is overwritten with the contents of `file1`. If `file2` does not exist, it is created. In either case, `file1` ceases to exist.
mv -i file1 file2	Same as the previous command, except that if `file2` exists, the user is prompted before it is overwritten.
mv file1 file2 dir1	Move `file1` and `file2` into directory `dir1`. The directory `dir1` must already exist.
mv dir1 dir2	If directory `dir2` does not exist, create directory `dir2` and move the contents of directory `dir1` into `dir2` and delete directory `dir1`. If directory `dir2` does exist, move directory `dir1` (and its contents) into directory `dir2`.

3.9 rm

- Remove(delete) files and directories
- `rm item ...`
- To delete directories, use the `-r` (or `-recursive`) option.

Exercise:

1. In your home directory, create a folder named `temp`.
2. Change directory to `temp`.
3. Create two directories `dir1` and `dir2` in `temp`.
4. Copy the `passwd` file from the `/etc` folder to the `temp` folder using the `-v` option.
5. Once again, copy the `passwd` file from `/etc` to `temp`, this time using the `-i` option.
6. Rename the `passwd` file in `temp` to `fun`.
7. Move `fun` to `dir1`.
8. Move `dir1` to `dir2`.
9. Delete `dir2`.

10. Delete `temp`.

Wildcards Using wildcards (which is also known as globbing) allows you to select filenames based on patterns of characters.

Pattern	Matches
*	All files
g*	Any file beginning with g
b*.txt	Any file beginning with b followed by any characters and ending with .txt
Data???	Any file beginning with Data followed by exactly three characters

3.10 alias

- Aliases are commands that we can define ourselves, built from other commands.
- `alias name='string'`
- `alias foo='cd /usr; ls; cd -'`
- `alias rm='rm -i'`
- To remove an alias, the `unalias` command is used, like so:
  - `unalias foo`

3.11 type

- Display a command's type
- `type command`
- Examples:
- `type type`
- `type ls`
- `type cp`

3.12 which

- Display an executable's location
- **which** command
- Examples:
- `which ls`
- `which` works only for executable programs, not builtins or aliases
- `which cd`

3.13 man

- Display manual page
- `man mkdir`
- Many executable programs support a `-help` option that displays a description of the command's supported syntax and options
- `ls -help`

3.14 IO Redirection

stdout, stderr, and stdin:

- Keeping with the Unix theme of "everything is a file," programs such as `ls` actually send their results to a special file called standard output (often expressed as `stdout`) and their status messages to another file called standard error (`stderr`).
- By default, both standard output and standard error are linked to the screen and not saved into a disk file.
- In addition, many programs take input from a facility called standard input (`stdin`), which is, by default, attached to the keyboard.
- **Redirecting stdout:**
- `ls /usr/bin > output.txt`
- `less output.txt`
- **Overwrite:**

- `> output.txt`
- `less output.txt`
- **To append:** »
- **Redirecting stderr:**
- `ls /bin/usr`
- `ls /bin/usr 2> output.txt`
- **For redirecting stdin**: <

3.15 cat

- Shows the contents of a file, all at once
- `cat filename`
- Example: `cat /etc/os-release`
- Because `cat` can accept more than one file as an argument, it can also be used to join files together.

3.16 grep

- Used to find a specific string within files
- `grep pattern filename`
- Let us create a text file to search:
- `ls /usr/bin > out.txt`
- Search for pattern `zip` in the file:
- `grep zip out.txt`
- This gives all programs in `/usr/bin` that had the name `zip` in it.

3.17 wc

- Used to display the number of lines, words, and bytes contained in files.
- The `-l` option limits its output to report only lines.
- Example: Count the number of executable programs in `/usr/bin`:
- `ls /usr/bin > out.txt`
- `wc -l out.txt`

3.18 |(pipe)

- Using the pipe operator `|`, the output of one command can be piped into the input of another.
- The pipe can be used to link commands together to perform more complex tasks that would otherwise take multiple steps (and possibly writing information to disk).
- Example: Count the number of executable programs in `/usr/bin`:
- `ls /usr/bin | wc -l`
- Example: Find all programs in `/usr/bin` that has the name `zip` in it:
- `ls /usr/bin | grep zip`

The Difference Between > and |

- The redirection operator > connects a command with a file, while the pipeline operator `|` connects the output of one command with the input of a second command.
- `command1 > file1`
- `command1 | command2`

3.19 echo

- prints its text arguments on standard output
- `echo this is a test`
- Using `echo` to create a simple text file:
- `echo 'sample text' > myfile.txt`

3.20 touch

- modifies the time-stamp of a file
- `touch filename`
- If the file does not exist, `touch` creates the file.
- So touch can be used to create files, though it is not its primary purpose

3.21 Text Editors

- Text editors fall into two basic categories: graphical and text-based.
- `gedit` is a graphical editor.
- Popular text-based editors are `nano`, `vi`, and `emacs`.
- The `nano` editor is a simple, easy-to-use editor while `vi` and `emacs` are more powerful editors.
- `nano myfile`
- The file will open or one will be created if it doesn't exist
- **Exercise:** Use `nano` to open a file, make edits, save it and exit.

3.22 printenv

- The shell maintains a body of information during our shell session called the environment.
- To see all the environment variables and their values:
- `printenv`
- `printenv` can also print the value of a specific variable:
- `printenv HOME`
- `printenv PATH`
- When we type a command in the shell, it searches the list of directories in the `PATH` variable.

3.23 history

- bash maintains a history of commands that have been entered. This list of commands is kept in your home directory in a file called $.bash\_history$
- To view command history:
- `history`
- In this listing, each command is numbered.
- If you want to reissue a command at line number 20:
- `!20`
- The `!` can also be followed by a string to repeat the last history list item starting with that string. Example: `!c`

3.24 chmod

- change permissions for a file or directory
- Octal or symbolic representation:
- `> foo.txt`
- `ls -l foo.txt`
- `chmod 600 foo.txt`
- `ls -l foo.txt`
- `chmod +x`

Exercise:

1. Create a file with some text in it.
2. Check its permissions.
3. Change its permissions to `r__r__r`.
4. Try to make changes to the file.
5. Try to delete the file using `rm`

3.25 Interrupting programs and signalling EOF

- Use `Ctrl-C` to stop running a command:
- `sleep 60`
- `Ctrl-C`
- `sleep 60; echo hi`
- `Ctrl-C`
- Use `Ctrl-D` to indicate EOF.

3.26 passwd

- To change a user password:
- `passwd`
- To change another user's password:
- `passwd user2`

3.27 useradd

- Creates a new user account
- To add a new user named `user2`:
- `sudo useradd -m user2`
- The `-m` option creates a home directory for the new user automatically.
- After creating a new user, you typically need to set a password for the user account using the `passwd` command:
- `sudo passwd user2`
- To switch to `user2`:
- `sudo -u user2 bash`
- `whoami`
- `cd`
- `exit`

- To delete `user2`:
- `sudo userdel user2`
- This does not delete the home directory `/home/user2`

3.28 uname

- Displays information about a Linux machine's operating system and hardware platform.
- `uname -a`
- Prints:
  - Kernel name
  - Hostname
  - Kernel release
  - Kernel version and the build date
  - Machine architecture name
  - CPU type
  - Hardware platform
  - Operating system

3.29 time

- Measures the execution time of a specified command or program and reports various metrics, including real, user, and system time.
  - Real Time: The actual elapsed time, from start to finish, including time spent waiting for I/O and other processes.
  - User Time: The amount of CPU time spent executing user-mode instructions within the process.
  - System Time: The amount of CPU time spent executing system-level instructions on behalf of the process.
- `time ls -l`
- `time mv file1 dir1`

3.30 du

- `du [options] [file/directory]`
- The `du` (disk usage) command is used to estimate and summarize file and directory space usage indicated by the given path name.
- If you do not specify a path name, `du` assumes the current directory.
- By default, `du` gives disk usage in KB.
- `-h` option gives the output in a more human-readable form (e.g., KB, MB):
- `du -h dir1`

3.31 Installing and maintaining software

High-level tools like apt permit a package to be downloaded from a repository and installed with full dependency resolution

Command	Description
`apt update`	Update all repository info
`apt install package_name`	Install a package
`apt upgrade`	Update all installed packages
`apt autoremove`	Remove packages that are no longer needed
`apt remove package_name`	Remove an installed package
`apt purge package_name`	Remove an installed package and delete configuration files
`apt search string`	Search repositories for a package
`apt show package_name`	Show details for a package

- If a package file has been downloaded from a source other than a repository, it can be installed directly
- `sudo dpkg -i package_name`

3.32 Comperssing Files

- The `gzip` program is used to compress one or more files. When executed, it replaces the original file with a compressed version of the original.
  - `ls -l /etc > foo.txt`
  - `ls -l foo.txt`
  - `gzip foo.txt`
  - `ls -l foo.*`

- The corresponding `gunzip` program is used to restore compressed files to their original, uncompressed form.
  - `gunzip foo.txt.gz`
- The `bzip2` program is similar to `gzip` but uses a different compression algorithm that achieves higher levels of compression at the cost of compression speed.
- In most regards, it works in the same fashion as `gzip`.
- A file compressed with `bzip2` is denoted with the extension `.bz2`.

3.33 Archiving

- Archiving is the process of gathering up many files and bundling them together into a single large file. The `tar` program is the classic tool for archiving files.
- `tar <mode and options> <archive> <files or locations>`
  - c: create an archive
  - v: verbose
  - f: specify name of archive file
  - x: extract an archive
- `tar -cvf temp.tar temp`
- By default, `tar` extracts all the components to the current directory. To indicate where to extract the components, add the `-C` option and specify the path:
- `tar -xvfC <archive_name> <path>`
- **Creating a Tar Archive with Compression**:
  - `z`: use gzip compression
  - `j`: use bzip2 compression
  - `tar -cvzf temp.tgz temp`
- **Extracting a compressed archive**:
  - `tar -xvf temp.tgz`
- The `tar` command will auto-detect compression type and will extract the archive.

3.34 fdisk

- The `fdisk` command in Linux is used to manage disk partitions. It allows you to create, delete, resize, and modify partitions on your hard drive.
- Listing Partitions
  - `fdisk -l`
- Selecting a Disk
  - `sudo fdisk /dev/sdX`
  - Replace `sdX` with the appropriate disk identifier (e.g., `sda`, `sdb`).
- Creating a New Partition
  - Enter `n` to create a new partition.
  - Choose the partition type (primary or extended).
  - Specify the partition number and size.
- Deleting a Partition
  - Enter `d` to delete a partition.
  - Specify the partition number to delete.
- Saving Changes
  - Enter `w` to write the changes to the disk.
- Quitting Without Saving
  - Enter `q` to quit without saving any changes.
- Always be cautious when using `fdisk`, as making changes to disk partitions can result in data loss if not done carefully.

3.35 dmesg

- The `dmesg` command is a Linux utility that displays kernel-related messages retrieved from the kernel ring buffer.
- The ring buffer stores information about hardware, device driver initialization, and kernel module messages during system startup.
- The `dmesg` command is handy for diagnosing hardware and kernel-related issues.

SEMESTER S2

IT WORKSHOP

(Common to Group A&B)

Course Code	**GXESL208**	**CIE Marks**	50
Teaching Hours/Week (L: T:P: R)	0:0:2:0	**ESE Marks**	50
Credits	1	**Exam Hours**	2 Hrs. 30 Min.
Prerequisites (if any)	None	**Course Type**	Lab

Course Objectives:

1. To provide a basic understanding about computer hardware, software, and computer network.
2. To familiarize the learner with the web development process using HTML, CSS, and Javascript.

Details of Experiment

Expt. No	Experiment (Minimum 10 Experiments)
1	Practice Computer Hardware – Familiarization CPU Box, Motherboard, CPU & Chip-set, Interface cards, Card slots, Hard disk, Cables, SMPS, NIC, Various ports, etc. Computer Peripherals - I/O Devices. Storage devices, Interface cards – Buses – Firmware
2	Familiarization of Boot process
3	Familiarizing installation of Linux and Windows operating systems
4	Familiarizing basic Unix/Linux commands - ls, mkdir, cp, mv, grep, rmdir, chmod, useradd, passwd, history, dmesg, cpuinfo, uname, du, time, write, fdisk

5	Familiarizing networking hardware - RJ45, UTP, fibre, switch, NIC, router, Wireless Access Point (WAP), modem
6	Familiarizing basic networking commands - ifconfig, ping, traceroute, nslookup, ssh, scp, telnet, ftp
7	View network traffic using Wireshark/Packet tracer
8	Familiarizing the steps how to configure and establishing a network connecting
9	Shell programming in Linux(bash)
10	Create a web page and deploy on a local web server.
11	Use Javascript to validate forms.
12	Create an image slider using HTML, CSS, and JavaScript. Allow users to navigate between images using previous and next buttons.
13	Familiarisation of LaTeX - Basic only
14	Familiarisation of Development Environments - Visual studio code, Sublime Text, Atom
15	Introducing Repositories - Git / Bitbucket

Course Assessment Method

(CIE: 50 Marks, ESE 50 Marks)

Continuous Internal Evaluation Marks (CIE):

Attendance	Preparation/Pre-Lab Work, experiments, Viva and Timely completion of Lab Reports / Record. (Continuous Assessment)	Internal Exam	Total
5	**25**	**20**	**50**

End Semester Examination Marks (ESE):

Procedure/ Preparatory work/Design/ Algorithm	Conduct of experiment/ Execution of work/ troubleshooting/ Programming	Result with valid inference/ Quality of Output	Viva voce	Record	Total
10	**15**	**10**	**10**	**5**	**50**

Mandatory requirements for ESE:

- Submission of Record: Students shall be allowed for the end semester examination only upon submitting the duly certified record.

Course Outcomes (COs)

At the end of the course the student will be able to:

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Experiment with the fundamental hardware components of a computer and how to interface them with software systems.	**K3**
CO2	Make use of the command line of Linux operating system and shell programming.	**K3**
CO3	Experiment with the data network communication scenarios using Wireshark.	**K3**
CO4	Develop basic websites using HTML, CSS & JavaScript and manage the versions.	**K3**

K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create

CO-PO Mapping Table

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3		3							3
CO2	3	3	3	3	3							3
CO3	3	3	3	3	3							3
CO4	3	3	3	3	3							3

1: Slight (Low),2: Moderate (Medium),3: Substantial (High), : No Correlation

Text Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Invitation to Computer Science	G.Michael Schneider, Judith Gersting	Cengage	2/e, 2020
2	LINUX for Developers: Jumpstart Your Linux Programming Skills	William Rothwell	Person	1/e, 2018
3	HTML, CSS, and JavaScript All in One, Sams Teach Yourself	Julie C. Meloni Jennifer Kyrnin	Pearson	1/e, 2018

Reference Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	The Architecture of Computer Hardware, Systems Software, & Networking: An Information Technology Approach	Irv Englander	Wiley	5/e, 2014
2	Mastering Git : Attain expert level proficiency with Git for enhanced productivity and efficient collaboration	Jakub Narębski	Packt	1/e, 2016
3	Web Design with HTML, CSS, JavaScript and Jquery	Jon Duckett	Wiley	1/e, 2014

Video Links (NPTEL, SWAYAM…)	
Sl. No.	**Link ID**
1	https://overthewire.org/wargames/bandit/
2	https://www.w3schools.com/

Continuous Assessment (25 Marks)

1. Preparation and Pre-Lab Work (7 Marks)

- Pre-Lab Assignments: Assessment of pre-lab assignments or quizzes that test understanding of the upcoming experiment.
- Understanding of Theory: Evaluation based on students' preparation and understanding of the theoretical background related to the experiments.

2. Conduct of Experiments (7 Marks)

- Procedure and Execution: Adherence to correct procedures, accurate execution of experiments, and following safety protocols.
- Skill Proficiency: Proficiency in handling equipment, accuracy in observations, and troubleshooting skills during the experiments.
- Teamwork: Collaboration and participation in group experiments.

3. Lab Reports and Record Keeping (6 Marks)

- Quality of Reports: Clarity, completeness and accuracy of lab reports. Proper documentation of experiments, data analysis and conclusions.
- Timely Submission: Adhering to deadlines for submitting lab reports/rough record and maintaining a well-organized fair record.

4. Viva Voce (5 Marks)

- Oral Examination: Ability to explain the experiment, results and underlying principles during a viva voce session.

Final Marks Averaging: The final marks for preparation, conduct of experiments, viva, and record are the average of all the specified experiments in the syllabus.

Evaluation Pattern for End Semester Examination (50 Marks)

1. Procedure/Preliminary Work/Design/Algorithm (10 Marks)

- Procedure Understanding and Description: Clarity in explaining the procedure and understanding each step involved.
- Preliminary Work and Planning: Thoroughness in planning and organizing materials/equipment.
- Algorithm Development: Correctness and efficiency of the algorithm related to the experiment.
- Creativity and logic in algorithm or experimental design.

2. Conduct of Experiment/Execution of Work/Programming (15 Marks)

- Setup and Execution: Proper setup and accurate execution of the experiment or programming task.

3. Result with Valid Inference/Quality of Output (10 Marks)

- Accuracy of Results: Precision and correctness of the obtained results.
- Analysis and Interpretation: Validity of inferences drawn from the experiment or quality of program output.

4. Viva Voce (10 Marks)

- Ability to explain the experiment, procedure results and answer related questions
- Proficiency in answering questions related to theoretical and practical aspects of the subject.

5. Record (5 Marks)

- Completeness, clarity, and accuracy of the lab record submitted