BUC73105 · TOPIC 2

What is actually inside the machine?

System Components · hardware, operating systems, software

WHAT IS INSIDE THE COMPUTER

GROUP 1
HARD WARE

1. WHAT IS HARDWARE
2. GIVE EXAMPLES
3. EXPLAIN FUNCTIONS

GROUP 2
OPERATING SYSTEMS

1. WHAT IS AN Ops
2. GIVE EXAMPLES
3. WHAT Ops DOES

GROUP 3
SOFTWARE

1. WHAT IT IS
2. GIVE EXAMPLES
3. WHAT EACH SOFTWARE DOES

GROUP 4
HOW DO THE 3 WROK TOGETHER?

REQUIRED TO GIVE REAL LIFE EXAMPLES OF WHERE ITS APPLIED WITH PICTURES AND VIDEOS IF POSSIBLE
5 MINUTES TO PRESENT FINDINGS

WHERE WE HAVE GOT TO

Last topic you learned who. Now you learn what.

Topic 1 gave you	The person: what an administrator does, and the difference between system and network administration.
Topic 2 gives you	The thing they administer: the machine, its parts, and the operating system that runs it.
Why this order	You cannot manage a host in Topic 4 if you cannot name what is inside it and how it starts.

This topic is heavier on content than the last one. That is on purpose — everything after it depends on it.

BY THE END OF TOPIC 2

Three things you will be able to do

1

Explain what 'The System' is

Describe a computer system as hardware, software, network, users and policy working together — not just a box.

2

Describe handling hardware

Name every major component, say what it does, how it fails, and how to handle it safely and correctly.

3

Describe operating systems

Explain what an OS does, how a machine boots, and how Windows and Linux differ in practice.

Assessed by: component identification exercises, written descriptions, and a spoken defence of your work.

PART A

What is ‘The System’?

Before the parts, the whole. What are you actually administering?

- Why the system is never just the box on the desk
- The five layers, from silicon to the person using it
- Where your system ends and somebody else's begins
- Four shapes a system can take

LEARNING OUTCOME 1

The system is not the box

Ask most people what a computer system is and they point at the metal box. An administrator sees five things.

HARDWARE

The physical parts. What you can drop.

SOFTWARE

The operating system and everything running on it.

NETWORK

What connects it to everything else.

PEOPLE

The users, and what they actually do with it.

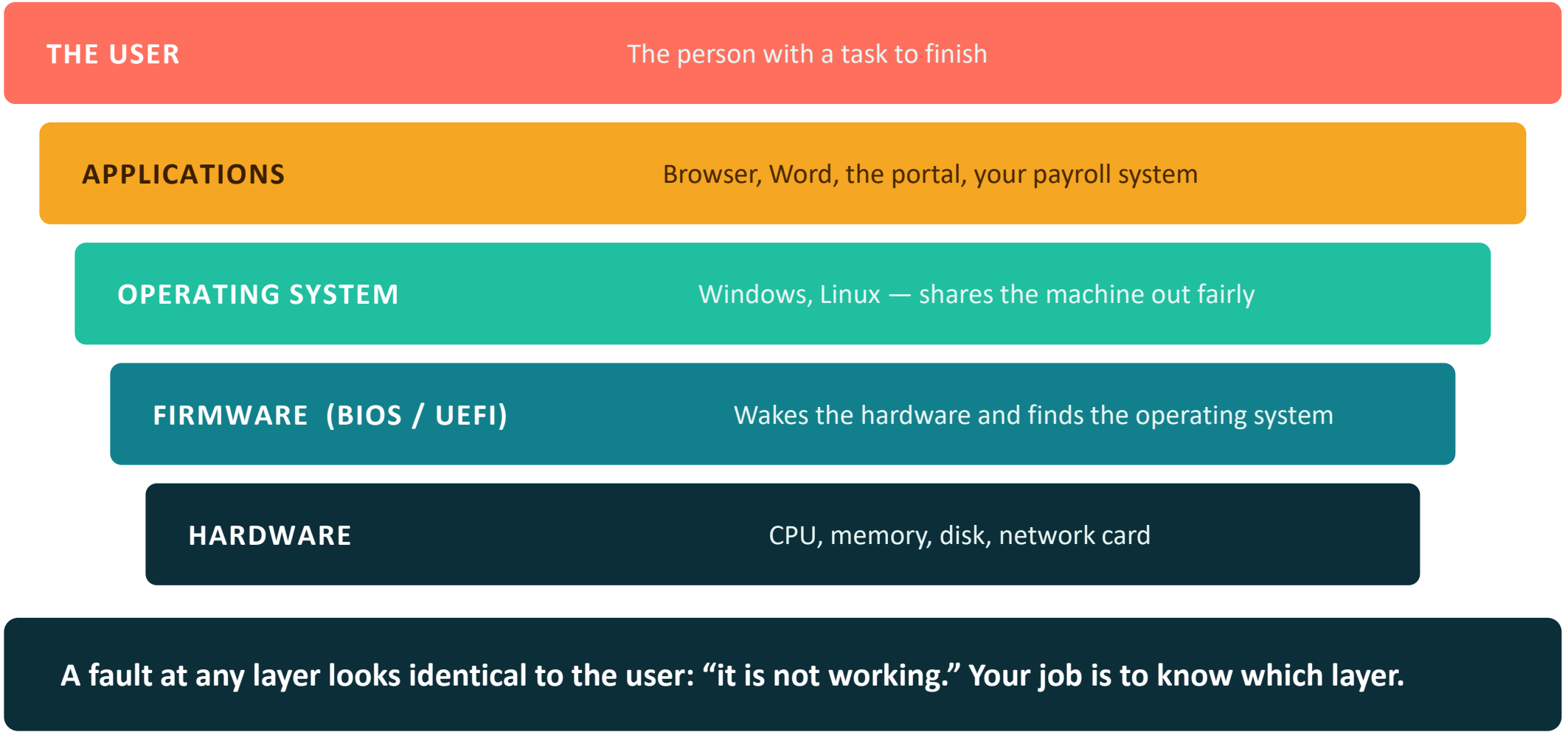
POLICY

The rules. What is allowed, and who decided.

Remove any one of the five and the system stops being useful. A perfect server with no policy is a disaster waiting.

FROM SILICON TO THE PERSON

Every system is a stack of layers



A QUESTION EVERY ADMINISTRATOR MUST ANSWER

Where does your system end?

The machine	Yours. You install, patch, secure it.	*Nobody else may change it*
The network in the building	Usually yours.	*Shared with the network admin*
The internet link	Not yours — the ISP's.	*You can only report and escalate*
The cloud service	Not yours. Google or Microsoft run it.	*You own only the accounts and data*
The user's own phone	Not yours — unless policy says so.	*This is where BYOD arguments start*

Agree your boundary before an incident, not during one. “Not ours” must be policy, not an excuse.

SYSTEMS COME IN SHAPES

Four ways a system is put together

STANDALONE

One machine, doing its own work. A shop till. Simple, but nothing is shared and nothing is backed up centrally.

CLIENT – SERVER

Many machines asking one central machine. Your campus portal. Easy to control, but the server is a single point of failure.

PEER TO PEER

Machines sharing directly with each other. A small office with a shared folder. Cheap, and impossible to manage at scale.

VIRTUAL / CLOUD

Machines that exist as software, here or in somebody else's data centre. Flexible, fast to build, and easy to lose track of.

Most Ugandan small offices are peer to peer and wish they were client–server.

GUIDED PRACTICE 8 MIN

Draw the system you are sitting in

On paper. Five boxes, stacked, exactly like the layers slide — but filled in with YOUR machine.

The user	Write your own name. You are part of the system.
Applications	Name the three programs you use most.
Operating system	Which one, and which version? Be exact.
Firmware	BIOS or UEFI — write “not sure” if you do not know. That is honest.
Hardware	Make and model of your machine.

Then answer

Which layer do you understand least?

Post that one word in the chat — it tells me what to teach hardest today.

PART B

Handling hardware

The physical layer. What is inside, what it does, how it fails, and how to touch it safely.

- The parts inside every computer, and the job each one does
- Storage, and why one disk is never enough
- What makes a server different from the machine on your desk
- The rules for handling hardware without destroying it

LEARNING OUTCOME 2 · OPEN THE BOX

Eight parts, in every computer you will ever meet

1
CPU
Does the thinking

2
RAM
Short-term memory

3
Storage
Long-term memory

4
Motherboard
Joins everything

5
Power supply
Feeds everything

6
Network card
Talks to the world

7
Cooling
Stops it dying

8
Graphics
Draws the screen

You must be able to name all eight, unprompted, and say what each one does. That is examinable.

THE THREE THAT DECIDE HOW FAST IT FEELS

CPU, memory, storage

WHAT IT IS	WHAT IT DOES	HOW YOU KNOW IT IS THE PROBLEM
CPU	Executes instructions. Measured in GHz and cores.	*Everything is slow at once, fan is loud, CPU at 100%*
RAM	Holds what is being used right now. Emptied at power off.	*Slow only when many programs are open; disk thrashing*
Storage	Keeps data when the power goes. HDD or SSD.	*Slow to open files and to boot; "disk full" errors*

The analogy that works: the CPU is the cook. RAM is the counter space. Storage is the store room.
A fast cook with no counter space still produces food slowly.

THE FOUR THAT DECIDE WHETHER IT STAYS ALIVE

Board, power, network, cooling

Motherboard	Connects every component together	*Rare, but fatal. Usually means replace the machine*
Power supply	Converts mains power to what the parts need	*Random shutdowns, will not power on, burning smell*
Network card	Connects the machine to the network	*No link light, no IP address, works on another port*
Cooling	Removes heat so the parts survive	*Loud fan, machine hot, sudden shutdowns under load*

In Uganda, power supplies and cooling fail most — unstable mains, heat and dust. Budget for both.

WHY A SERVER COSTS FIVE TIMES MORE

What makes server hardware different

Redundant power	Two power supplies. One fails, the machine keeps running while you replace it.
ECC memory	Memory that detects and corrects its own errors. Desktop RAM silently corrupts data.
Hot-swap drives	Pull a failed disk out and slide a new one in without switching the machine off.
Remote management	iLO, iDRAC or IPMI — a second tiny computer that lets you fix the server from home, even when it is off.
Rack form factor	Built to bolt into a rack, cooled front to back, designed to run for years without stopping.

You are not paying for speed. You are paying for not having to switch it off.

HARDWARE DOES NOT LIVE IN A VACUUM

The room around the machine

POWER

Unstable mains kills equipment. UPS for ride-through, generator for the long outage, surge protection always.

COOLING

Heat shortens component life. Airflow front to back, never block vents, air conditioning for a real server room.

DUST

The quiet killer in this climate. Blocked fans, overheating, failed power supplies. Clean on a schedule.

ACCESS

A locked door is a security control. If anyone can walk in and touch the server, nothing else you do matters.

This returns in Topic 4 as ‘physical considerations of the server room’. Start noticing rooms now.

DO IT NOW

12 MIN

Spec your own machine

Every component we just named is discoverable with a command. Find yours.

`systeminfo`

Windows — the whole picture: OS, RAM, model

`wmic cpu get name`

Windows — which processor

`lscpu` · **`free -h`** · **`df -h`**

Linux — processor, memory, disk

`Get-PhysicalDisk`

PowerShell — your disks and their health

Fill in your spec sheet

- Make and model
- Processor and cores
- RAM installed
- Disk size, and how full
- Disk type: HDD or SSD?
- Operating system and version

Post one line in the chat: your RAM, and how full your disk is. We will compare.

LEARNING OUTCOME 3

What is an operating system?

Not “Windows.” That is a brand. Here is the definition you will be assessed on.

The software that manages the hardware, shares it fairly between programs, and gives people a way to use it.

MANAGER

Decides which program gets the processor, the memory and the disk, and for how long.

TRANSLATOR

Programs do not talk to hardware directly. They ask the OS, which knows how each device works.

INTERFACE

Gives you a desktop or a command line so a human can actually operate the machine.

EXAMINABLE · LEARN ALL FIVE

The five jobs of an operating system

1 Process management	Starts, stops and schedules running programs	*A frozen program; 100% CPU*
2 Memory management	Gives each program memory and keeps them apart	*Out of memory; one app crashes another*
3 File management	Stores, names, finds and protects files	*Corrupt files; permission denied*
4 Device management	Talks to hardware through drivers	*Printer not working; no sound; no Wi-Fi*
5 Security and users	Who may log in, and what they may touch	*Wrong access; no audit trail*

Every fault in Topic 7 will be one of these five failing. Learn the list now, diagnose faster later.

THE MOST IMPORTANT LINE INSIDE AN OPERATING SYSTEM

Kernel space and user space

USER SPACE

Your programs live here. Browser, Word, the portal. If one crashes, the machine survives.

KERNEL SPACE

The core of the OS, with total control of the hardware. If something crashes here, the whole machine goes down.

Why it matters to you

A blue screen or a kernel panic is not an app problem. It is hardware, drivers or the OS itself — a completely different hunt.

POWER BUTTON TO LOGIN SCREEN

What happens when you switch it on

1	**POST**	Firmware checks the hardware is present and healthy. Beeps or lights if not.
2	**BIOS / UEFI**	Firmware wakes the board and decides which device to boot from.
3	**Bootloader**	Small program on the disk — GRUB on Linux, Windows Boot Manager — loads the kernel.
4	**Kernel loads**	The core of the OS starts, detects hardware, loads drivers.
5	**Init / systemd**	Starts every background service in order: network, logging, printing.
6	**Login**	The system is ready and asks who you are.

This sequence is how you diagnose a machine that will not start: where it stops tells you what is wrong.

IN THE CHAT

6 MIN

Where did it stop?

Three machines will not start. Using the boot sequence, name the stage that failed — and what you would check.

A Press power. No lights, no fan, no beep. Nothing at all.

B It powers on, fans spin, then: ‘No bootable device found.’

C The maker's logo appears, then it hangs on a spinning circle forever.

Type your answers as A-, B-, C- with the stage number. Then defend one of them out loud.

THE FAMILIES YOU WILL MEET

Four operating system families

WINDOWS

Windows 10/11 on desktops. Windows Server in the machine room. Licensed, GUI-first, dominant in offices.

LINUX

Free and open. Ubuntu, CentOS, Debian, Red Hat. Runs most servers and almost all of the internet.

macOS / UNIX

Apple's desktop OS, built on UNIX. Same family as Linux underneath — the commands feel familiar.

MOBILE

Android and iOS. Full operating systems, and increasingly your responsibility as an administrator.

You will run both Windows Server and Ubuntu Server this semester. Not one or the other — both.

EXAMINABLE · AND ASKED AT INTERVIEW

Windows Server vs Linux server

	WINDOWS SERVER	LINUX SERVER
Cost	Licensed per server and per user	Free; you may pay for support
How you drive it	Graphical tools first, PowerShell too	Command line first, GUI optional
User directory	Active Directory	LDAP, or joined to AD
Installing software	Installers, or winget	Package managers: apt, yum, dnf
Updates	Windows Update; often needs a reboot	apt upgrade; reboot less often
Typical use	File, print, email, AD, business apps	Web, database, DNS, containers, cloud

Neither is better. They are chosen for different jobs — and most real organisations run both.

TWO MAPS YOU MUST BE ABLE TO READ

Where things live: Windows and Linux

WINDOWS

C:\Windows	the operating system
C:\Program Files	installed software
C:\Users	people's files
C:\Users\...\AppData	settings
Registry	configuration database

LINUX

/	the top of everything
/etc	configuration files
/home	people's files
/var/log	logs — you will live here
/usr/bin	programs

Linux has no C drive. Everything hangs off one root, /, including other disks. That surprises everybody once.

WHAT IS RUNNING RIGHT NOW

Processes and services

A process	Any program currently running. Your browser is a process. It starts, uses memory, and stops.
A service	A program that runs in the background with no window, started by the system, usually at boot.
Why it matters	Servers are almost entirely services. ‘The portal is down’ usually means one service has stopped.

WINDOWS

```
Task Manager  ·  services.msc
Get-Service  ·  Get-Process
```

LINUX

```
ps aux  ·  top  ·  htop
systemctl status  ·  systemctl list-units
```

DO IT NOW

10 MIN

Meet the things running on your machine

Windows: open Task Manager, then the Services tab. Linux: run systemctl list-units --type=service

1. Count how many services are running right now. Most people guess ten. Look at the real number.
2. Pick any three. Write down the exact name of each one.
3. Search what each one does. Use the name plus the word 'service'.
4. For each: is this something the OS needs, something an app installed, or something you cannot identify?

That fourth question is the security habit. Anything running that you cannot explain deserves your attention.

THE GLUE BETWEEN HARDWARE AND SOFTWARE

Drivers, firmware and updates

A driver	Software that teaches the OS how to talk to one piece of hardware. No driver, no printing, no Wi-Fi, no sound.
Firmware	Software stored on the hardware itself — in the BIOS chip, the disk, the router. It runs before any OS exists.
Why update them	Fixes faults, closes security holes, adds support for newer devices.
Why be careful	A failed firmware update can permanently kill the device. Never update firmware on a machine running on battery or unstable mains.

Rule for this country: never flash firmware without a UPS. A power cut mid-update destroys the board.

HOW YOU WILL PRACTISE EVERYTHING FROM HERE ON

One machine pretending to be many

Virtualisation lets one physical computer run several complete operating systems at the same time, each believing it owns the hardware.

Type 1 hypervisor	Runs directly on the hardware, no OS underneath. VMware ESXi, Hyper-V, Proxmox. This is what data centres use.
Type 2 hypervisor	Runs as a program inside your normal OS. VirtualBox, VMware Workstation. This is what you will use.
Containers	Lighter again. Share the host's kernel and package only the application. Docker. Topic 6 territory.
Why it matters here	You can break, rebuild and snapshot a whole server on your own laptop — free, and without risking anything real.

Install VirtualBox this week. Every practical task from Topic 4 onwards assumes you have it.

EVERYTHING YOU HAND IN FOR TOPIC 2

Your four tasks

#	What you do	What you hand in	Due
1	Spec sheet. Document two machines fully, using commands — not guesswork.	A table per machine: model, CPU, RAM, disk type and size, OS and version, plus the command used for each value.	Before session 2
2	Name the parts. Photograph a real computer, inside or its ports.	A labelled photo identifying at least six components or ports, with one line on what each does.	Before session 3
3	Compare the two servers. Windows Server against Ubuntu Server.	A six-row comparison table in your own words, and one paragraph on which you would choose for your kingdom, and why.	End of Topic 2
4	Build and boot. Install Ubuntu Server in VirtualBox.	Screenshots of the six boot stages you can observe, plus your notes on every problem you hit and how you solved it.	End of Topic 2

Task 4 is the foundation of every practical from Topic 4 onwards. Do not leave it to the last night.

BEFORE YOU GO

Name one component and one job of the OS.

In the chat. Two words each. Nobody may repeat what is already there.

NEXT — Topic 3: Networked Communities. The machines are ready. Now the people arrive.