

For a competent Linux user, the main challenge with Windows is not technical capability — it is learning Microsoft's conventions, administrative model, and UI logic. The quickest route is to treat Windows as “a different operating culture,” not as a simplified Unix.

A practical approach:

1. Install Windows in a Safe, Disposable Way First

Before committing to bare metal:

- Install it in a VM using:
 - [VirtualBox](#)
 - or [VMware Workstation Player](#)

This lets him:

- break things safely
- snapshot and restore
- explore admin tools without risk
- understand the filesystem and permissions model

A Linux user learns Windows much faster when experimentation is consequence-free.

2. Learn the Windows Mental Model

A Linux expert usually expects:

- text-first administration
- composability
- transparency
- config files
- package managers
- predictable paths

Windows is different:

- GUI-first historically
- registry-centric
- services + MMC tools
- many settings duplicated in old/new interfaces
- strong backward compatibility
- lots of hidden state

Tell him not to fight this initially.

The fastest path is:

“Learn how Windows expects to be administered before trying to Unix-ify it.”

3. Core Things He Should Learn First

Filesystem Layout

Equivalent mental map:

Linux	Windows
/	C:\
/home/user	C:\Users\name
/etc	Registry + scattered config
/var/log	Event Viewer
/bin	System32 + PATH
sudo	UAC elevation

Important point:

System32 on 64-bit Windows is actually mostly 64-bit binaries. Microsoft naming history is odd.

Learn These Administrative Tools

The essentials:

Purpose	Tool
Task/processes	Task Manager
Services	services.msc
Event logs	Event Viewer
Disks	Disk Management

Drivers/devices	Device Manager
Startup programs	Task Manager → Startup
Networking	ncpa.cpl + PowerShell
System config	Settings + Control Panel

The “.msc” consoles are particularly important because they are the closest thing to classic sysadmin tooling.

4. Use PowerShell Early

This is critical.

A Linux user who ignores PowerShell usually ends up hating Windows.

PowerShell is Microsoft’s real administrative interface now:

- object-based shell
- strong automation
- remote administration
- scripting
- package management

He should learn:

- `Get-Help`
- `Get-Command`
- `Get-Service`
- `Get-Process`
- `Get-ChildItem`
- `Get-NetAdapter`
- `Get-WinEvent`

Good starting reference:

- [Microsoft PowerShell Documentation](#)
-

5. Install a Proper Package Manager

Without this, Linux users often become frustrated immediately.

Recommend:

- [WinGet](#)

Examples:

```
winget install Firefox  
winget install VLC  
winget upgrade --all
```

This makes Windows feel much less primitive.

Some Linux users also like:

- [Chocolatey](#)

But WinGet is now the native Microsoft approach.

6. Explain UAC Properly

Linux users often misunderstand Windows privilege escalation.

Key idea:

- being in Administrators does not mean processes run elevated
- elevation is per-process
- UAC is effectively a privilege boundary

This confuses many experienced Unix users initially.

7. Teach the Important Keyboard Shortcuts

These massively improve usability:

Shortcut	Function
Win+X	Power admin menu
Win+R	Run dialog
Win+E	File Explorer
Win+Terminal	Terminal

Ctrl+Shift+Esc	Task Manager
Alt+Tab	Task switching
Win+V	Clipboard history

Especially:

Win+X is the hidden “power user” menu.

8. Tell Him to Ignore the GUI Noise Initially

Windows 11 in particular contains:

- consumer clutter
- ads/recommendations
- Copilot integration
- duplicated settings
- legacy menus

A technically competent user should focus first on:

- filesystem
- networking
- services
- PowerShell
- drivers
- update model

The cosmetic layer is secondary.

9. Learn Windows Troubleshooting Philosophy

Linux:

- logs
- stderr
- config files

Windows:

- Event Viewer
- Reliability Monitor
- Device Manager
- SFC/DISM

- registry/state corruption
- driver conflicts

Useful commands:

sfc /scannow

DISM /Online /Cleanup-Image /RestoreHealth

10. Use Windows Terminal Immediately

The default legacy console experience is poor.

Install/use:

- [Windows Terminal](#)

Then add:

- PowerShell
- Command Prompt
- WSL
- SSH sessions

This gives a far more coherent environment.

11. WSL Helps, But Use It Carefully

Windows Subsystem for Linux is excellent, but:

He should avoid retreating into Linux immediately.

Otherwise he may never properly learn:

- Windows networking
- filesystem semantics
- permissions
- tooling
- update mechanisms

Use WSL as a supplement, not an escape hatch.

12. Best Learning Method

The fastest route is practical tasks.

Tell him to do things like:

- install printers
- configure shares
- map network drives
- manage services
- install drivers
- create scheduled tasks
- remote into another PC
- configure backups
- recover from a failed update

Windows becomes understandable through operational tasks, not theory.

13. Very Important Mindset Point

A lot of experienced Linux users waste time trying to make Windows behave like Linux.

That is usually counterproductive.

The productive mindset is:

“Understand native Windows administration first. Then adapt workflows.”

Once he understands:

- PowerShell
- MMC tools
- services
- event logging
- Windows networking
- update management

...Windows stops seeming “weird” and instead becomes internally consistent, just with very different design priorities from Unix.