Ubuntu Linux Cheat Sheet

Beginner Edition – Hackberry Pi CM5

1 – Getting Oriented on Your Hackberry Pi

Your Hackberry Pi CM5 runs a Cortex-A76 quad-core ARM SoC at 2.4 GHz with up to 16 GB RAM. Ubuntu runs natively on it as a first-class ARM64 platform. The BlackBerry keyboard works for quick commands, but for longer sessions plug in a USB keyboard via one of the two USB 3.0 ports.

First Things First

`sudo apt update && sudo apt upgrade -y`	Update everything. Do this first, always.
`uname -a`	Show kernel version and architecture (aarch64).
`cat /etc/os-release`	Confirm Ubuntu version and codename.
`free -h`	Check RAM – know what you're working with.
`df -h`	Disk space. Important if booting from SD.
`lsblk`	List block devices – see your SD, NVMe, USB drives.
`lsusb`	List USB devices.
`ip a`	Show network interfaces and IP addresses.
`vcgencmd measure_temp`	Check CPU temp – the aluminium case helps, but monitor it.

Tip: If you put an NVMe SSD in the M.2 slot and boot from it, everything will feel significantly faster than microSD.

2 – Navigating the Filesystem

`pwd`	Print working directory – where am I?
`ls -la`	List everything including hidden files, with permissions.
`cd /path/to/dir`	Change directory. **cd ~** goes home, **cd -** goes back.
`mkdir -p projects/lab1`	Create nested directories in one shot.
`cp -r src/ dst/`	Copy recursively.
`mv old new`	Move or rename.
`rm -rf dir/`	Delete recursively. **No undo.** Be careful.
`find / -name '*.conf' 2>/dev/null`	Search the whole system for .conf files.
`locate filename`	Fast search using a pre-built index (run **updatedb** first).
`tree -L 2`	Visual directory tree, 2 levels deep.

Tip: Tab-completion is your best friend. Type the first few letters and hit Tab.

3 – Reading & Editing Files

`cat file.txt`	Dump entire file to screen.
`less file.txt`	Page through a file. Press **q** to quit.
`head -n 20 file.txt`	First 20 lines.
`tail -n 20 file.txt`	Last 20 lines.

`tail -f /var/log/syslog`	Live-follow a log file – great for debugging.
`grep -ri 'error' /var/log/`	Recursively search for a string, case-insensitive.
`nano file.txt`	Beginner-friendly text editor. Ctrl+O saves, Ctrl+X exits.
`vim file.txt`	Powerful editor. Press **i** to type, **Esc** then **:wq** to save & quit.
`diff file1.txt file2.txt`	Show differences between two files.

Tip: Start with nano. Learn vim later when you're comfortable.

4 – Permissions & Users

Linux permissions matter. Every file has an owner, a group, and a mode (read/write/execute for owner, group, others). The format **rwxr-x---** means: owner can read/write/execute, group can read/execute, others get nothing.

`chmod 750 script.sh`	Owner: rwx, Group: r-x, Others: none.
`chmod u+x script.sh`	Make executable for owner only.
`chown user:group file`	Change ownership.
`whoami`	Who are you logged in as?
`id`	Your user ID, group memberships.
`sudo command`	Run a command as root.
`sudo -s`	Drop into a root shell. Use sparingly.
`passwd`	Change your password.
`adduser newuser`	Create a new user.
`usermod -aG sudo newuser`	Give a user sudo privileges.

5 – Processes & System Management

`htop`	Interactive process viewer. Install with **sudo apt install htop**.
`ps aux`	Snapshot of all running processes.
`ps aux | grep nginx`	Find a specific process.
`kill PID`	Gracefully stop a process.
`kill -9 PID`	Force-kill a stubborn process.
`systemctl status ssh`	Check if the SSH service is running.
`systemctl start ssh`	Start the SSH daemon.
`systemctl enable ssh`	Auto-start SSH on boot.
`systemctl list-units --type=service`	List all active services.
`journalctl -xe`	Read system logs when something goes wrong.
`dmesg | tail -30`	Kernel messages – useful for hardware/driver issues.
`uptime`	How long has the system been running?

Tip: If the Hackberry feels sluggish, run htop and look for runaway processes eating CPU or RAM.

6 – Networking

The CM5 has onboard WiFi, but the aluminium case can attenuate the signal. Consider the external FPC antenna mod documented in the Hackberry GitHub repo, or plug in a USB WiFi adapter.

`ip a`	Show all interfaces and IPs.
`ip link set wlan0 up`	Bring a wireless interface up.
`nmcli dev wifi list`	Scan for WiFi networks.
`nmcli dev wifi connect SSID password PASS`	Connect to a WiFi network.
`nmcli con show`	List saved connections.
`ping -c 4 1.1.1.1`	Test connectivity (4 pings).
`traceroute 8.8.8.8`	Trace the route packets take.
`ss -tlnp`	Show listening TCP ports and which process owns them.
`curl ifconfig.me`	What's my public IP?
`wget https://example.com/file.zip`	Download a file.
`hostname -I`	Quick way to see your local IPs.
`resolvectl status`	Check DNS resolver configuration (systemd-resolved).

7 – Package Management (APT & Snap)

APT – The Core Package Manager

`sudo apt update`	Refresh the package index.
`sudo apt upgrade -y`	Upgrade all installed packages.
`sudo apt full-upgrade -y`	Upgrade, including removing obsolete packages.
`sudo apt install package`	Install a package.
`sudo apt remove package`	Remove a package (keeps config).
`sudo apt purge package`	Remove package AND config files.
`sudo apt autoremove`	Clean up orphaned dependencies.
`apt search keyword`	Search for packages.
`apt show package`	Show info about a package.
`dpkg -l | grep keyword`	Check if something is installed.

Snap – Ubuntu's Containerised Packages

`snap list`	Show installed snaps.
`snap find keyword`	Search the snap store.
`sudo snap install package`	Install a snap.
`sudo snap remove package`	Remove a snap.
`sudo snap refresh`	Update all snaps.

Tip: APT packages are generally leaner. Snaps auto-update and are sandboxed. Use whichever fits – no religion required.

8 – Software & Development Setup

Ubuntu is a great platform for learning to code and build projects. Here's how to get a basic dev environment set up on your Hackberry.

Essential Build Tools

`sudo apt install build-essential`	GCC, G++, make – the C/C++ toolchain.
`sudo apt install git`	Version control. Learn this early.
`sudo apt install curl wget`	HTTP tools for downloading things.
`sudo apt install python3 python3-pip python3-venv`	Python 3 with pip and virtual envs.
`sudo apt install nodejs npm`	Node.js and npm.

Git Basics

`git clone https://github.com/user/repo.git`	Download a repository.
`git status`	See what's changed.
`git add . && git commit -m 'msg'`	Stage everything and commit.
`git push`	Push commits to remote.
`git pull`	Pull latest changes.
`git log --oneline -10`	Last 10 commits, compact view.

Tip: Set up git config – git config --global user.name and user.email – before your first commit.

Python Virtual Environments

`python3 -m venv myenv`	Create a virtual environment.
`source myenv/bin/activate`	Activate it (prompt changes).
`pip install package`	Install packages inside the venv.
`deactivate`	Leave the virtual environment.

9 – Shell Productivity

`history`	Show command history.
`!!`	Re-run the last command. **sudo !!** is a lifesaver.
`Ctrl+R`	Reverse-search command history. Start typing to filter.
`command1 | command2`	Pipe: feed output of one command into another.
`command > file.txt`	Redirect output to a file (overwrite).
`command >> file.txt`	Append output to a file.
`command 2>&1`	Redirect stderr to stdout.
`alias ll='ls -la'`	Create a shortcut. Add to ~/.bashrc to persist.
`screen -S mysession`	Start a named terminal session (survives disconnect).
`screen -r mysession`	Reattach to it.
`tmux`	More modern alternative to screen.

Tip: Ubuntu uses bash by default. Add your favourite aliases to ~/.bashrc.

10 – Storage & Disks

`lsblk`	Show block devices and mount points.
`df -h`	Disk usage by filesystem.
`du -sh /path/`	Size of a directory.
`sudo fdisk -l`	List all partitions on all disks.
`sudo mount /dev/sda1 /mnt/usb`	Mount a USB drive.
`sudo umount /mnt/usb`	Unmount it.
`sudo blkid`	Show UUIDs and filesystem types.
`cat /etc/fstab`	See what mounts automatically at boot.

Tip: To auto-mount your NVMe SSD, add an entry to /etc/fstab using its UUID from blkid.

11 – Hackberry Pi CM5 – Hardware Tips

`vcgencmd measure_temp`	CPU temperature.
`vcgencmd get_throttled`	Check for thermal throttling (0x0 = all good).
`lsblk`	Confirm your boot device (SD vs NVMe).
`sudo fdisk -l /dev/nvme0n1`	Inspect NVMe SSD if installed.
`bluetoothctl`	Manage Bluetooth (for the onboard speakers).
`i2cdetect -y 1`	Scan I2C bus – see battery monitor, RTC, Stemma QT devices.
`sudo hwclock -r`	Read the RTC (CR927 battery backed).
`sudo hwclock -w`	Write system time to RTC.
`cat /sys/class/power_supply/*/voltage_now`	Read battery voltage (if exposed).

The keyboard is fully programmable via VIAL (vial.rocks). You can remap keys, create layers, and set up macros – worth doing to make the BlackBerry layout work for terminal use.

12 – Beginner Practice Projects

Project 1 – Set Up SSH & Go Headless

Enable SSH (sudo systemctl enable --now ssh), find your Hackberry's IP, and SSH into it from your laptop. Set up key-based auth with ssh-keygen and ssh-copy-id, then disable password login. This teaches SSH, keys, and service management.

Project 2 – Build a Personal Web Server

Install nginx (sudo apt install nginx), put a simple HTML page in /var/www/html, and access it from another device on your network. Then try adding a reverse proxy to a Python or Node app running on a different port.

Project 3 – Automate with Cron

Write a script that logs CPU temp and battery voltage to a file every 5 minutes. Schedule it with crontab -e. This teaches cron, scripting, and system monitoring.

Project 4 – Containerise Something with Docker

Install Docker (curl -fsSL https://get.docker.com | sh), pull a container like Pi-hole or Home Assistant, and run it. Learn docker ps, docker logs, docker compose. Containers are everywhere – start early.

Project 5 – Clone & Build an Open-Source Project

Pick a project on GitHub that interests you, clone it, read the README, install dependencies, and build it from source. Start with something small in Python or C. This teaches git, build systems, and reading other people's code.

13 – Resources

OverTheWire Bandit (overthewire.org/wargames/bandit) – The best free interactive Linux command line tutorial. Start here.

Linux Journey (linuxjourney.com) – Well-structured beginner curriculum covering fundamentals.

Ubuntu Server Guide (ubuntu.com/server/docs) – Official documentation for Ubuntu Server on ARM.

Hackberry GitHub (github.com/ZitaoTech/HackberryPiCM5) – Hardware docs, assembly guides, OS images.

explainshell.com – Paste any command and it breaks down every flag.

tldr pages (sudo apt install tldr) – Simplified man pages with practical examples. Try: tldr tar