



Course Information Sheet

UCC111-2: Linux Installation and Package Management

This course is part of the Teaching Unit (TU): UCC111 - Linux Foundational (LPIC-101).

No	Title	Description
0	Code + Title	UCC111-2: Linux Installation and Package Management
1	Preamble	<p>Purpose: To provide the essential basics for installing, managing startup, and maintaining software on Linux systems, which are fundamental skills for working on servers and in cloud environments.</p> <p>Target audience: Students in the Professional Bachelor's Degree in Cloud Computing (Semester 1).</p> <p>Type of access: Paid (UoM Standard).</p> <p>Hours: 12</p> <p>Teaching mode: CS (Synchronous Chat), CA (Asynchronous Chat) (UdM Standard).</p>
2	Teaching team	Teaching supervisor: Teaching team:
3	Institution	This course is offered at the UNIVERSITY OF THE MOUNTAINS (UdM) , within the INSTITUTE OF SCIENCE AND TECHNOLOGY (ISST) .
4	Prerequisites	None
5	Summary	This course covers the key concepts necessary for setting up a functional Linux system. It addresses storage preparation (disk design), boot system installation, and, above all, effective management of software packages and libraries for the main Linux distribution families (Debian/RPM).

6	Glossary <p>Linux distribution</p> <p>A coherent set consisting of the Linux kernel, a package manager, and a set of tools and applications (e.g., Ubuntu, Debian, Fedora).</p> <p>Kernel</p> <p>The core of the operating system that manages hardware resources, processes, memory, and peripherals.</p> <p>CLI (Command Line Interface)</p> <p>Command line interface, used to administer the system without a graphical interface.</p> <p>ISO</p> <p>Installation image of an operating system containing the files necessary for installation.</p> <p>Bootloader</p> <p>Program loaded when the PC starts up, allowing the operating system to be launched.</p> <p>GRUB (GRand Unified Bootloader)</p> <p>Bootloader widely used on Linux systems, configurable and multi-OS.</p> <p>UEFI (Unified Extensible Firmware Interface)</p> <p>Modern replacement for BIOS, enabling advanced boot management and GPT disk support.</p> <p>BIOS (Basic Input/Output System)</p> <p>Legacy firmware that manages hardware initialization at startup.</p> <p>Dual-Boot</p> <p>Configuration allowing multiple operating systems to coexist</p>
----------	---

	<p>on the same machine.</p> <p>MBR (Master Boot Record)</p> <p>Old partitioning scheme limited to 2 TB and 4 primary partitions.</p> <p>GPT (GUID Partition Table)</p> <p>Modern partitioning scheme supporting large disks and an unlimited number of partitions.</p> <p>Partition</p> <p>Logical division of a disk used to store systems or data.</p> <p>Swap</p> <p>Disk space used as an extension of RAM.</p> <p>Mount point</p> <p>Directory in which a partition or external disk is integrated into the Linux tree structure.</p> <p>fstab</p> <p>File containing the configuration of file systems to be mounted automatically.</p> <p>Shared library (.so)</p> <p>File containing code that can be reused by several programs simultaneously, avoiding duplication.</p> <p>ldconfig</p> <p>Command that updates the shared library cache.</p> <p>ld.so.conf</p> <p>File listing the paths containing dynamic libraries.</p>
--	---

	<p>Dependency</p> <p>A package or file required for another program to function properly.</p>
	<p>Package</p> <p>File containing a program, its data, and metadata, managed via a manager.</p>
	<p>dpkg</p> <p>Low-level package manager on Debian/Ubuntu that allows you to install or remove .deb files.</p>
	<p>apt (Advanced Package Tool)</p> <p>High-level manager that automatically resolves dependencies and manages repositories.</p>
	<p>Repository</p> <p>Server containing software packages accessible via APT.</p>
	<p>sources.list</p> <p>File containing the list of repositories used by APT.</p>
	<p>RPM (Red Hat Package Manager)</p> <p>Package format and low-level manager used on RHEL, CentOS, Fedora.</p>
	<p>YUM (Yellowdog Updater Modified)</p> <p>Former high-level manager for RPM systems allowing automatic installation of dependencies.</p>
	<p>DNF (Dandified Yum)</p> <p>A modern replacement for YUM, faster and more reliable.</p>
	<p>Package group</p>

	<p>A set of related software programs that can be installed together (e.g., web server, graphical environment).</p> <p>Log</p> <p>File containing system events and actions, useful for diagnosing errors.</p> <p>Broken package</p> <p>A package that is partially installed or contains unmet dependencies.</p> <p>Package manager lock</p> <p>Situation where another process (e.g., automatic update) prevents the installation of new packages.</p> <p>APT/YUM cache</p> <p>Folder containing metadata and previously downloaded packages.</p> <p>Checksum (SHA256)</p> <p>Fingerprint used to verify the integrity of a downloaded ISO image.</p> <p>GPG (GNU Privacy Guard)</p> <p>Encryption tool used to verify the authenticity of repositories and packages.</p> <p>lsblk / fdisk / parted</p> <p>Commands for inspecting and managing disks and partitions.</p> <p>mount / umount</p> <p>Commands used to mount or unmount a file system.</p>
--	---

		systemctl Command for managing services, useful for checking whether an installed service is working correctly.
7	Progress sheet	CHAPTER 1 — Introduction to Linux installation 1.1. Understanding Linux distributions <ul style="list-style-type: none"> • Definition of a distribution • Differences between distributions (Debian, Ubuntu, RHEL, CentOS, Fedora, etc.) • Package management models according to families
8	Learning objectives	By the end of this course, students should be able to: <ul style="list-style-type: none"> 1. Master the installation of a Linux system <ul style="list-style-type: none"> • Prepare installation media (ISO, bootable USB drive). • Choose and configure an appropriate partitioning scheme (MBR/GPT). • Understand and apply the file system hierarchy 2. Configure and manage system startup <ul style="list-style-type: none"> • Understand the role of the bootloader (GRUB). <ul style="list-style-type: none"> • Install, repair, and customize GRUB. • Manage multi-boot environments. 3. Manage system libraries and dependencies. <ul style="list-style-type: none"> • Identify shared libraries (.so). • Manipulate the library cache via <code>ldconfig</code>. • Resolve issues related to missing dependencies. 4. Install, update, and remove packages. <ul style="list-style-type: none"> • Use low-level tools (dpkg, rpm). • Master high-level managers (apt, yum/dnf). • Configure software repositories and manage GPG keys. 5. Diagnose and maintain a functional Linux system. <ul style="list-style-type: none"> • Detect and repair broken packages. • Read the logs related to package installation.

		<ul style="list-style-type: none"> • Update the system safely.
9	Learning materials	<p>Types of resources available: audio, video, source files (Word, PDF, PPT, links), Google Meet.</p>
10	Assessment	<p>Teaching strategies adopted: The specific assessment method is: Continuous assessment, Practical work, Final exam.</p> <p>Standard school protocol:</p> <ul style="list-style-type: none"> - Continuous assessment (CC) accounts for 30% of the overall mark, broken down as follows: (Participation in tutorials 20%, Completion of activities 20%, Attendance/presence in class 20%, Completion of practical work 40%) - The written exam accounts for 70%.
11	Course outline (detailed content)	<p>CHAPTER 1 — Introduction to Linux installation</p> <p>1.1. Understanding Linux distributions</p> <ul style="list-style-type: none"> • Definition of a distribution • Differences between distributions (Debian, Ubuntu, RHEL, CentOS, Fedora, etc.) • Package management models according to families <p>1.2. Life cycle of a Linux system</p> <ul style="list-style-type: none"> • Installation • Configuration • Maintenance • Updates and upgrades <p>1.3. Preparing for installation</p> <ul style="list-style-type: none"> • Choosing a distribution based on context • ISO download and integrity check (SHA256, GPG) • Creating bootable media (Rufus, Balena, dd) <p>CHAPTER 2 — Disk design and organization</p> <p>2.1. Partitioning types</p> <ul style="list-style-type: none"> • MBR vs. GPT • Comparison and limitations • Primary, extended, and logical partition tables <p>2.2. Partition types</p>

- System partition
- Swap partition
- /home partition
- /boot partition
- EFI partition (ESP)

2.3. Linux file systems

- Ext2, Ext3, Ext4
- XFS, Btrfs
- FAT32, NTFS (compatibility)

2.4. Mount points

- Role of the mount point
- File system hierarchy (FHS)
- Automatic mounting: /etc/fstab
- Mount options (rw, ro, noexec, etc.)

CHAPTER 3 — Installation and management of bootloaders

3.1. How a bootloader works

- Definition and role
- BIOS/UEFI interaction
- Linux boot chain

3.2. GRUB2 — Installation and configuration

- GRUB structure: grub.cfg, scripts
- Menu customization
- Modifying boot options
- Recovery modes

3.3. Bootloader troubleshooting

- Reinstalling GRUB
- Handling common errors (error 15, no such device, etc.)
- Backup and restore

3.4. Special case: Multi-boot

- Dual-boot with Windows
- Automatic detection via os-prober
- Boot order

CHAPTER 4 — Shared Library Management

	<p>4.1. Understanding shared libraries</p> <ul style="list-style-type: none"> • .so files • Dynamic links • Application dependencies <p>4.2. Library identification and configuration</p> <ul style="list-style-type: none"> • The <code>ldd</code> command • Configuration file <code>/etc/ld.so.conf</code> • Standard library directories <p>4.3. Library cache management</p> <ul style="list-style-type: none"> • ldconfig function • Creating symbolic links • Cache update <p>CHAPTER 5 — Package management in Debian and derivative distributions</p> <p>5.1. Basic package management with dpkg</p> <ul style="list-style-type: none"> • Local installation of a .deb • Removal and purging • Inspection with <code>dpkg -l, -s, -c</code> <p>5.2. Advanced package management</p> <ul style="list-style-type: none"> • Manipulation of configuration files • Status and partial installation of packages • Troubleshooting incomplete installations <p>5.3. Managing repositories with APT</p> <ul style="list-style-type: none"> • <code>/etc/apt/sources.list</code> • Adding repositories and GPG keys • Updating lists: <code>apt update</code> <p>5.4. Common operations with APT</p> <ul style="list-style-type: none"> • Installation, upgrade, removal • Search and diagnostics • Major upgrade (dist-upgrade, full-upgrade) <p>5.5. Meta package management</p> <ul style="list-style-type: none"> • Tasksel • Functional groups
--	--

CHAPTER 6 — Package management in RedHat, Fedora, and derivatives

6.1. Basic management with RPM

- Installing an .rpm
- Signature verification
- Querying RPM packages

6.2. Using YUM/DNF

- Manager architecture: repositories, cache, metadata
- Essential commands: installation, removal, update

6.3. Advanced management

- Package groups
- Cache cleanup
- Conflict and broken dependency management

6.4. Repository configuration

- Creating a local repository
- Adding an external repository
 - Temporary activation/deactivation

CHAPTER 7 — System maintenance and diagnostics

7.1. Updating the system

- Security
- Critical patches
- Automatic Updates (cron, systemd timers)

7.2. Diagnosing package-related problems

- Missing dependencies
- Broken packages
- Packet manager locking

7.3. Logging and logs

- dpkg.log
- yum.log / dnf.log
- journalctl for issues related to installed services

CHAPTER 8 — Practical exercises

		8.1. Lab 1: Complete installation of Linux on a VM 8.2. Lab 2: Manual MBR and GPT partitioning 8.3. TP 3: Installation and configuration of GRUB 8.4. TP 4: Installing packages on Debian and RedHat 8.5. TP 5: Creating a local repository (Debian or RPM)
12	Activity	Learning activity (tutorials), Assessment activity (tutorials + corrected assignments), Self-assessment activity (self-assessment test or multiple-choice questions), Summative activity (problem situation).
13	Bibliographies and list of links	<i>Provide a list of resources for further study of the course content.</i> (UdM standard)