



# 信息工程系

# 教 案

课程名称: Linux 操作系统  
教 师: 陈凯斌  
总学时: 54  
理论学时: 0  
实训学时: 54  
上课班级: 电子信息工程技术 251 班  
电子信息工程技术 251 订单班  
授课学期: 2025-2026 第 2 学期

# 实验一：认识 Linux 操作系统(3 课时)

## 教学目标

### 1、知识目标:

理解 Linux 系统的定义、发展历史及核心人物；掌握 Linux 与 Windows 操作系统的核心区别；熟悉 Linux 系统的特点及应用场景；了解 Linux 系统的学习方法和学习重点。

### 2、能力目标:

能准确区分 Linux 不同发行版（Ubuntu、CentOS 等）的差异；能独立查阅 Linux 应用案例，梳理 Linux 系统的核心优势；能结合专业需求，明确自身 Linux 学习的重点方向。

### 3、素质目标:

培养“先查阅资料、再总结归纳”的自主学习习惯；形成“理论结合实操”的学习意识；树立“多思考、多积累”的学习态度。

### 4、课程思政目标:

引导学生理解 Linux 开源精神的内涵，尊重开源协议和知识产权；结合 Linux 在服务器、物联网等领域的应用，认识我国信息技术自主可控的重要性；培养学生的科技报国情怀。

## 重难点内容

重点: Linux 系统的特点及应用场景；Linux 与 Windows 操作系统的核心区别；Linux 系统的学习方法。

难点: 理解 Linux 系统的开源特性及优势；建立 Linux 系统的学习思维，明确学习重点和方法。

## 教学方法与过程

教学方法: 讲授法+演示法

教学过程:

1. 导入: 教师通过服务器、嵌入式设备、物联网终端等使用 Linux 系统的场景，提问“为什么这些设备大多选择 Linux 系统？”“Linux 和我们常用的 Windows 有什么不一样？”，引出本节课主题，激发学生兴趣。

2. 讲授: 结合课件讲授 Linux 系统的定义、发展历史及重要人物；详细对比 Linux 与 Windows 操作系统的核心区别（开源性、多用户多任务、安全性、资源占用等）；讲

解 Linux 系统的特点及典型应用场景（服务器、嵌入式、物联网等）；分享 Linux 系统的学习方法和学习重点。

3. 演示: 演示浏览 Linux 系统相关资料，展示不同 Linux 发行版（Ubuntu、CentOS 等）的界面差异；展示 Linux 系统在实际场景中的应用案例（如网站服务器、智能家居设备）。

4. 总结: 教师回顾本节课关键知识点，梳理 Linux 系统的核心优势和学习方向；布置课后任务。

# 实验二：Ubuntu Linux 安装(含虚拟机安装、linux 子系统)(6 课时)

## 教学目标

### 1、知识目标:

掌握 Linux 系统的三种安装途径（双系统、虚拟机、Linux 子系统）；了解安装前的准备工作（软件、硬件、镜像文件下载与校验）；熟悉 VMware 软件或 Virtual Box 的安装、激活与基础配置；掌握 Ubuntu 系统的详细安装步骤；了解 Windows 系统下 Linux 子系统的启用、安装与初始化方法。

### 2、能力目标:

能独立完成 VMware 软件或 Virtual Box 安装及 Ubuntu 系统虚拟机安装、初始化；能熟练操作虚拟机（启动、关闭、快照、硬件配置修改）；能启用 Windows 系统下的 Linux 子系统，完成安装与初始化；能实现虚拟机与物理机、Linux 子系统与 Windows 系统的文件互传；能排查安装过程中的常见故障。

### 3、素质目标:

培养“动手前先准备、操作中多观察”的严谨习惯；形成“故障排查先自查、再求助”的问题解决意识；树立“规范操作、注重细节”的工程素养。

### 4、课程思政目标:

引导学生使用官方渠道下载软件和镜像文件，树立安全意识和版权意识；通过小组互助安装，体验“集中力量办大事”的集体主义精神；在分区设置、配置优化中，培养学生的全局思维和责任意识。

## 重难点内容

重点: VMware 软件或 Virtual Box 的安装与基础配置；Ubuntu 系统虚拟机的详细安装步骤；Linux 子系统的启用与安装；虚拟机与物理机的文件共享方法。

难点: 三种安装方式的区别及场景选择；VMware 或者 Vbox 虚拟机的硬件配置优化；Ubuntu 系统安装过程中的分区设置；Linux 子系统的启用故障排查与文件共享配置。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师回顾上节课 Linux 系统的应用场景，提问“我们要学习 Linux 实操，首先需

要搭建什么环境？”“不同的安装方式有什么区别，我们该如何选择？”，引出本节课安装任务。

2. 讲授: 结合课件讲解 Linux 系统的三种安装途径及适用场景；讲解安装前的准备工作（硬件要求、镜像文件下载与校验、VMware 软件获取）；详细讲解 VMware 软件的安装、激活与基础配置（内存、CPU、磁盘设置）；讲解 Ubuntu 系统的安装步骤（分区设置、账户配置、语言与时区设置）；讲解 Linux 子系统的启用、安装与初始化方法；讲解文件共享的原理与操作步骤。

3. 演示: 教师现场演示 VMware 软件安装与激活；演示 Ubuntu 系统虚拟机安装的完整流程，重点演示分区设置和初始化配置；演示虚拟机硬件配置修改和快照操作；演示 Linux 子系统的启用与安装；演示虚拟机与物理机的文件互传操作；演示常见安装故障的排查方法。

4. 实践: 学生上机实践，完成以下任务：（1）安装 VMware 软件并激活；（2）安装 Ubuntu 系统虚拟机，完成初始化配置；（3）修改虚拟机硬件配置，创建快照；（4）启用 Windows 下的 Linux 子系统，完成安装；（5）实现虚拟机与物理机、子系统与 Windows 的文件互传；（6）尝试双系统安装（可选）。教师巡视指导，及时解答学生疑问。

5. 总结: 教师回顾本节课安装流程和关键注意事项，梳理常见故障及解决方法；布置课后任务：记录安装过程中的问题及解决方法，完成安装总结报告；尝试优化虚拟机配置，提升系统运行流畅度。

# 实验三：熟悉 Ubuntu Linux 桌面环境(1 课时)

## 教学目标

### 1、知识目标:

熟悉 Ubuntu 桌面环境的组成（顶部面板、左侧任务栏、桌面图标）；掌握桌面环境的基本操作（打开应用、切换窗口、关闭应用）；了解 Linux 系统文件管理器的基础使用；掌握网络连接配置方法。

### 2、能力目标:

能快速熟悉 Ubuntu 桌面各组件的功能及操作；能熟练打开、切换、关闭应用程序；能使用文件管理器浏览目录、打开文件；能成功配置网络连接，测试网络通畅性。

### 3、素质目标:

培养“快速适应新环境、主动探索功能”的自主学习能力；形成“注重操作熟练度、提高效率”的意识。

### 4、课程思政目标:

引导学生对比 Ubuntu 与 Windows 桌面环境的差异，培养包容不同系统的思维；在探索桌面功能的过程中，培养学生的探索精神和创新意识。

## 重难点内容

重点: Ubuntu 桌面环境的组成及基本操作；文件管理器的基础使用；网络连接配置。

难点: 快速熟悉桌面操作逻辑，区分与 Windows 桌面的差异；无明显技术难点，重点在于操作熟练度。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师展示 Ubuntu 桌面环境，提问“大家熟悉这个桌面吗？和我们常用的 Windows 桌面有什么不同？”，引导学生观察桌面组件，引出本节课内容。
2. 讲授: 结合桌面演示，讲解 Ubuntu 桌面环境的组成（顶部面板、左侧任务栏、桌面图标）及各组件功能；讲解桌面基本操作（打开应用、切换窗口、关闭应用、桌面图标管理）；讲解文件管理器的基础使用（浏览目录、打开文件、创建文件/目录）；讲解网络连接配置（有线、无线连接设置）。
3. 演示: 教师演示桌面各组件的操作，演示应用程序的打开、切换与关闭；演示文件管

理器的使用，浏览系统核心目录；演示网络连接配置，测试网络通畅性。

4. 实践: 学生上机实践，完成以下任务：（1）熟悉桌面各组件功能，练习应用程序的打开、切换与关闭；（2）配置网络连接，浏览网页测试网络；（3）使用文件管理器浏览系统核心目录，熟悉目录布局；（4）创建新的文件和目录，练习文件管理器的基本操作。教师巡视指导，纠正不规范操作。

5. 总结: 教师回顾本节课桌面操作和网络配置的关键步骤，强调与 Windows 桌面的差异；布置课后任务：熟练掌握桌面基本操作和文件管理器使用，为后续命令行操作奠定基础。

## 实验四：文字模式入门(2 课时)

### 教学目标

#### 1. 知识目标:

掌握 Linux 文字界面的启动与切换方法；了解终端的打开、关闭与外观配置；掌握命令行界面的基本操作（输入命令、命令补全、查看历史命令）；熟悉帮助文档的使用方法（man 命令、--help 参数）。

#### 2. 能力目标:

能熟练实现图形界面与文字界面的切换；能灵活打开、关闭终端，配置终端外观；能正确输入命令，使用命令补全和历史命令功能；能使用帮助文档查询常用命令的用法。

#### 3. 素质目标:

培养“命令输入严谨、注重细节”的习惯；形成“遇到不懂查帮助、自主解决问题”的能力；树立“高效操作、规范使用”的意识。

#### 4. 课程思政目标:

引导学生认识命令行界面在 Linux 运维中的重要性，培养学生脚踏实地、精益求精的工匠精神；在练习命令输入的过程中，培养学生的耐心和专注力。

### 重难点内容

重点: 图形界面与文字界面的切换；命令行界面的基本操作；帮助文档的使用方法。

难点: 文字界面与图形界面的快速切换操作；帮助文档的内容解读与关键信息查找；命令输入规范及常见错误排查。

### 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师演示命令行界面操作, 提问“为什么很多 Linux 管理员更喜欢用命令行操作?”“命令行相比图形界面有什么优势?”, 引出本节课文字模式的学习。
2. 讲授: 结合课件讲解 Linux 文字界面的启动与切换方法 (快捷键、命令); 讲解终端的打开、关闭方法及外观配置 (字体、颜色、大小); 讲解命令行界面的基本操作 (命令输入格式、命令补全、历史命令调用); 详细讲解帮助文档的使用方法 (man 命令、--help 参数的用法)。
3. 演示: 教师演示图形界面与文字界面的切换; 演示终端的打开、关闭与外观配置; 演示命令输入、命令补全和历史命令调用; 演示使用 man 命令和--help 参数查询 ls、pwd 等常用命令的用法, 解读帮助文档内容。
4. 实践: 学生上机实践, 完成以下任务: (1) 练习图形界面与文字界面的切换; (2) 打开终端, 配置终端外观; (3) 练习命令输入、命令补全和历史命令调用; (4) 使用帮助文档查询 ls、pwd、cd 命令的用法, 记录关键参数。
5. 总结: 教师回顾本节课命令行操作的关键知识点, 梳理常见错误及解决方法; 布置课后任务: 熟练掌握帮助文档的使用方法, 练习常用命令的输入, 为后续目录、文件管理奠定基础。

# 实验五：目录管理(3 课时)

## 教学目标

### 1、知识目标:

掌握 Linux 目录树结构及核心目录 (/root、/home、/etc 等) 的功能; 理解绝对路径与相对路径的概念及区别; 掌握目录相关操作命令的用法 (ls、cd、mkdir、rmdir、pwd); 了解特殊目录 (., ..、~、/) 的含义及使用方法。

### 2、能力目标:

能熟练使用 ls、cd、pwd 命令查看、切换目录; 能使用 mkdir、rmdir 命令创建、删除目录; 能灵活运用绝对路径和相对路径访问不同层级的目录; 能批量创建与删除目录。

### 3、素质目标:

培养“规范管理目录、条理清晰”的习惯; 形成“路径使用准确、命令操作熟练”的严谨性; 树立“高效操作、减少误操作”的意识。

### 4、课程思政目标:

引导学生认识目录树结构的逻辑性, 培养学生的逻辑思维和全局观念; 在批量操作中, 培养学生的高效意识和责任意识, 避免误操作导致的损失。

## 重难点内容

重点: Linux 目录树结构及核心目录功能; 目录相关命令的用法; 绝对路径与相对路径的运用。

难点: Linux 目录树结构的理解与核心目录功能记忆; 绝对路径与相对路径的灵活运用; 批量创建与删除目录的命令使用技巧。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师展示 Linux 目录结构示意图, 提问“Linux 系统的目录是如何组织的?”“我们如何快速找到指定的目录?”, 引出本节课目录管理的内容。

2. 讲授: 结合示意图讲解 Linux 目录树结构, 详细介绍核心目录 (/root、/home、/etc、/bin 等) 的功能; 讲解绝对路径与相对路径的概念及区别, 举例说明两种路径的使用场景; 逐一讲解目录相关命令的用法 (ls 命令的常用参数、cd 命令的切换技巧、

mkdir 命令的批量创建、rmdir 命令的使用限制、pwd 命令的作用)；讲解特殊目录 (., .., ~, /) 的含义及使用方法。

3. 演示: 教师演示 ls 命令的不同参数使用 (ls -l、ls -a 等)；演示 cd 命令切换不同层级目录，灵活运用绝对路径和相对路径；演示 mkdir 命令创建单个、多个目录，mkdir -p 命令创建多级目录；演示 rmdir 命令删除空目录；演示 pwd 命令查看当前目录路径。

4. 实践: 学生上机实践，完成以下任务：(1) 使用 ls 命令查看不同目录的内容，尝试不同参数；(2) 使用 cd 命令切换到不同核心目录，分别用绝对路径和相对路径实现；(3) 使用 mkdir 命令创建单个、多个目录，创建多级目录；(4) 使用 rmdir 命令删除空目录；(5) 练习特殊目录的使用。教师巡视指导，及时纠正学生的误操作（如误删系统目录）。

5. 纠错: 针对学生实践中出现的常见错误（如路径输入错误、mkdir 创建多级目录忘记加参数、rmdir 删除非空目录失败等），集中讲解错误原因及解决方法，强化学生记忆。

6. 总结: 教师回顾本节课目录结构和命令用法，梳理绝对路径与相对路径的使用技巧；布置课后任务：熟练掌握所有目录操作命令，练习批量创建、删除目录，巩固目录树结构记忆。

# 实验六：文本文件(3 课时)

## 教学目标

### 1、知识目标:

掌握简单文本编辑器 nano 的基础使用方法；熟练掌握文本文件查看命令（cat、less、more、head、tail）的用法及区别；了解文件类型的查看方法（file 命令）；掌握文本内容查找命令 grep 的基础用法，能使用 grep 查找指定字符串。

### 2、能力目标:

能使用 nano 编辑器独立完成文本文件的创建、编辑、保存与退出；能根据文件大小和查看需求，灵活选择合适的文本查看命令；能使用 file 命令查看不同类型文件的属性；能使用 grep 命令查找文本文件中的指定内容，排查简单匹配错误。

### 3、素质目标:

培养“文本编辑严谨、命令使用规范”的习惯；形成“根据场景选择工具、高效完成操作”的意识；树立“遇到错误自主排查、反复验证”的实操理念。

### 4、课程思政目标:

引导学生在文本编辑和内容查找中注重细节，培养严谨细致的工匠精神；在自主排查 grep 匹配错误、文本查看异常等问题时，培养坚韧不拔、自主解决问题的能力；引导学生规范保存文本文件，树立文件管理的责任意识。

## 重难点内容

重点: nano 编辑器的基础操作；文本查看命令（cat、less、more、head、tail）的用法及场景选择；grep 命令的基础匹配用法；file 命令的使用。

难点: 不同文本查看命令的场景区分与灵活运用（如大文件与小文件的查看技巧）；grep 命令的基本匹配规则及常见错误排查；nano 编辑器的快捷键记忆与操作流畅性。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师回顾上节课目录管理的内容，提问“我们在目录中会接触到大量的文本文件，比如配置文件、日志文件，那么我们如何创建、编辑和查看这些文本文件？”“不同大小的文本文件，用什么方法查看更高效？”，引出本节课文本文件操作的核心内容。

2. 讲授: 结合课件讲解简单文本编辑器 nano 的核心功能，讲解 nano 的打开、编辑、

保存、退出操作及常用快捷键；逐一讲解文本查看命令的用法——**cat** 命令查看完整文本、**less** 和 **more** 命令分页查看（重点区分两者差异）、**head** 查看前几行、**tail** 查看后几行，举例说明不同命令的适用场景；讲解 **file** 命令的用法，用于查看文件类型；讲解 **grep** 命令的基础用法，包括查找指定字符串、忽略大小写匹配等简单参数，说明 **grep** 命令在文本筛选中的作用。

3. 演示: 教师演示 **nano** 编辑器的使用，创建文本文件、输入内容、保存并退出，演示快捷键的操作；演示不同文本查看命令的使用，对比 **cat**、**less**、**more** 查看大文件和小文件的差异，演示 **head**、**tail** 查看指定行数的内容；演示 **file** 命令查看不同类型文件（文本文件、可执行文件）的结果；演示 **grep** 命令查找文本中的指定字符串，排查匹配错误的常见原因。

4. 实践: 学生上机实践，完成以下任务：（1）使用 **nano** 创建、编辑文本文件，练习保存、退出及快捷键操作；（2）使用 **cat**、**less**、**more**、**head**、**tail** 命令查看不同大小的文本文件，记录各命令的使用感受；（3）使用 **file** 命令查看多个不同类型文件的属性，整理结果；（4）使用 **grep** 命令查找文本文件中的指定字符串，尝试忽略大小写匹配。教师巡视指导，及时纠正学生的误操作（如 **nano** 忘记保存、**grep** 拼写错误）。

5. 纠错: 针对学生实践中出现的常见错误（如 **nano** 保存时快捷键使用错误、**less** 命令退出方法不当、**grep** 匹配不到内容的原因排查、**file** 命令参数错误），集中讲解错误原因及解决方法，让学生现场纠正，强化记忆。

6. 总结: 教师回顾本节课文本编辑和查看的关键知识点，梳理各命令的适用场景，强调 **grep** 命令的基础匹配技巧；布置课后任务：熟练掌握 **nano** 编辑器和各文本查看命令，使用 **grep** 命令查找系统中的简单文本内容，记录操作过程。

# 实验七：文件管理（含权限管理）（3 课时）

## 教学目标

### 1. 知识目标:

掌握文件相关操作命令（touch、cp、mv、rm）的用法；理解 Linux 文件权限的结构（所有者、所属组、其他人）；掌握权限的两种表示方法（文字表示法、数字表示法）；熟练掌握权限相关命令（chmod、chown、chgrp）的用法；了解权限修改后的效果验证方法。

### 2. 能力目标:

能使用 touch、cp、mv、rm 命令熟练完成文件的创建、复制、移动、删除操作；能准确查看文件权限，熟练使用 chmod 命令修改文件权限（文字方式、数字方式）；能使用 chown、chgrp 命令修改文件的所有者和所属组；能验证权限修改后的访问效果，排查权限相关的简单故障。

### 3. 素质目标:

培养“文件操作规范、权限设置严谨”的习惯；形成“修改权限前先确认、删除文件前先检查”的安全意识；树立“权责对应、规范管理”的工程素养。

### 4. 课程思政目标:

引导学生认识文件权限在信息安全中的重要性，培养信息安全意识和责任担当；通过权限分配（所有者、所属组、其他人），引导学生理解“权责分明、各司其职”的管理理念；在删除文件、修改权限的操作中，培养严谨细致、认真负责的工作态度，避免误操作造成损失。

## 重难点内容

重点: 文件操作命令（touch、cp、mv、rm）的用法；文件权限的结构及两种表示方法；chmod 命令修改权限的用法；chown、chgrp 命令的基础使用。

难点: 文件权限结构的理解与权限表示方法的转换（文字表示法→数字表示法）；chmod 命令的数字方式与文字方式的灵活运用；权限修改后不同用户访问效果的验证；rm 命令删除文件的安全注意事项。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师展示一个带有权限标识的文件，提问“大家看到文件后面的 rwx 符号是什

么意思？”“为什么有的用户能打开文件，有的用户不能？”“我们创建的文件，如何控制其他人的访问权限？”，引出本节课文件管理与权限管理的核心内容。

2. 讲授: 结合课件讲解文件操作命令——touch 命令创建空文件、cp 命令复制文件（含复制目录的参数）、mv 命令移动/重命名文件、rm 命令删除文件（强调-f、-r 参数的使用风险）；详细讲解 Linux 文件权限的结构（所有者、所属组、其他人），讲解权限的文字表示法（r、w、x）和数字表示法（4、2、1），举例说明两种表示法的转换方法；逐一讲解权限相关命令——chmod 命令修改权限（文字方式、数字方式）、chown 命令修改所有者、chgrp 命令修改所属组；讲解权限修改后的效果验证方法，对比不同用户的访问权限差异。

3. 演示: 教师演示文件操作命令的使用，创建空文件、复制文件到不同目录、移动/重命名文件、删除文件（强调谨慎使用 rm -rf）；演示查看文件权限（ls -l 命令），解读权限标识；演示权限表示法的转换，使用 chmod 命令以文字方式和数字方式修改文件权限；演示 chown、chgrp 命令修改文件的所有者和所属组；演示不同用户访问修改权限后的文件，验证权限效果。

4. 实践: 学生上机实践，完成以下任务：（1）使用 touch、cp、mv、rm 命令完成文件的创建、复制、移动、删除操作；（2）使用 ls -l 命令查看文件权限，解读权限标识；（3）将指定文件的权限从文字表示法转换为数字表示法，使用 chmod 命令修改权限；（4）使用 chown、chgrp 命令修改文件的所有者和所属组；（5）切换不同用户，验证权限修改后的访问效果。教师巡视指导，解答学生疑问。

5. 总结: 教师回顾本节课文件操作和权限管理的关键知识点，梳理权限表示法的转换技巧和 chmod 命令的使用要点，强调 rm 命令的使用安全；布置课后任务。

# 实验八：用户账户管理(3 课时)

## 教学目标

### 1、知识目标:

理解 Linux 多用户特性及用户分类（root 用户、普通用户）；掌握用户相关查看命令（whoami、who）的用法；熟练掌握用户创建（adduser）、删除（deluser）、密码修改（passwd）的命令用法；掌握用户切换命令（su、sudo）的用法及区别；了解 /etc/passwd 文件的基本结构。

### 2、能力目标:

能使用 whoami、who 命令查看当前用户及登录用户信息；能独立创建普通用户，设置用户密码，配置用户基本信息；能删除普通用户，选择保留或删除其家目录；能熟练使用 su、sudo 命令切换用户、提升权限；能解读/etc/passwd 文件中的用户信息。

### 3、素质目标:

培养“用户管理规范、权限使用谨慎”的习惯；形成“root 权限不滥用、操作过程留痕迹”的安全意识；树立“分工协作、权责明确”的管理理念。

### 4、课程思政目标:

引导学生认识多用户管理在系统安全中的重要性，培养安全管理意识和规则意识；通过 sudo 命令的权限控制，引导学生理解“权力制衡、规范用权”的理念；通过小组协作管理用户，培养集体协作精神和责任担当。

## 重难点内容

重点: 用户创建（adduser）、删除（deluser）、密码修改（passwd）的命令用法；用户切换命令（su、sudo）的用法及区别；用户查看命令的使用。

难点: root 用户与普通用户的权限区别及安全注意事项；sudo 命令的配置与权限提升逻辑；用户删除时家目录的处理方式选择；su 与 sudo 命令的核心区别。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师提问“我们使用 Linux 系统时，为什么有的操作需要输入密码？”“root 用户和我们当前使用的普通用户有什么区别？”“如果多人共用一台 Linux 主机，如何为每个人分配独立的使用权限？”，引出本节课用户账户管理的内容。

2. 讲授: 结合课件讲解 Linux 多用户特性, 介绍用户分类 (root 用户、普通用户) 及权限差异; 讲解用户相关查看命令——`whoami` 查看当前用户、`who` 查看登录用户信息; 详细讲解用户创建命令 `adduser` 的用法, 包括设置用户名、密码、家目录等参数; 讲解用户删除命令 `deluser` 的用法, 说明保留/删除家目录的区别; 讲解用户密码修改命令 `passwd` 的用法, 包括修改自身密码和 `root` 修改其他用户密码; 讲解用户切换命令 `su` (切换用户) 和 `sudo` (提升权限) 的用法及核心区别, 强调 `sudo` 命令的安全优势; 简单介绍 `/etc/passwd` 文件的结构, 解读用户信息字段。

3. 演示: 教师演示 `whoami`、`who` 命令的使用, 查看用户信息; 演示 `adduser` 命令创建普通用户, 设置密码和家目录; 演示 `deluser` 命令删除普通用户, 分别演示保留和删除家目录的效果; 演示 `passwd` 命令修改密码, 包括普通用户修改自身密码和 `root` 修改其他用户密码; 演示 `su` 命令切换到普通用户和 `root` 用户, 演示 `sudo` 命令提升权限执行 `root` 操作; 演示查看 `/etc/passwd` 文件, 解读其中的用户信息。

4. 实践: 学生上机实践, 完成以下任务: (1) 使用 `whoami`、`who` 命令查看用户信息; (2) 使用 `adduser` 命令创建 2 个普通用户, 设置不同的密码; (3) 使用 `passwd` 命令修改其中一个用户的密码; (4) 使用 `su` 命令切换到创建的普通用户, 执行简单操作; (5) 使用 `sudo` 命令提升权限, 执行 `root` 操作; (6) 使用 `deluser` 命令删除一个普通用户, 保留其家目录, 删除另一个用户并删除其家目录; (7) 查看 `/etc/passwd` 文件, 解读用户信息。教师巡视指导, 强调 `root` 权限的使用安全, 及时解答学生疑问。

5. 总结: 教师回顾本节课用户账户管理的关键知识点, 梳理 `adduser`、`deluser`、`passwd`、`su`、`sudo` 命令的用法, 强调 `root` 权限的使用注意事项和 `sudo` 命令的优势; 布置课后任务: 熟练掌握所有用户管理命令, 尝试配置 `sudo` 命令的简单权限, 记录操作过程和遇到的问题。

# 实验九：vi 文本编辑器(3 课时)

## 教学目标

### 1、知识目标:

了解 vi 编辑器的定义、特点及适用场景；掌握 vi 编辑器的两种核心模式（输入模式、编辑模式）及切换方法；熟练掌握 vi 编辑器的基础操作（打开、保存、退出文件）；掌握编辑模式下的光标移动、字符删除、复制、粘贴命令；了解 vi 编辑器的常见误操作及处理方法。

### 2、能力目标:

能使用 vi 编辑器打开、创建、编辑文本文件；能熟练实现输入模式与编辑模式的切换；能在编辑模式下灵活移动光标、删除字符、复制粘贴内容；能熟练保存、退出 vi 编辑器；能处理 vi 编辑器的常见误操作（如忘记切换模式、非正常关闭文件）。

### 3、素质目标:

培养“模式切换严谨、操作规范熟练”的习惯；形成“遇到误操作不慌乱、自主排查解决”的应变能力；树立“耐心细致、精益求精”的操作态度。

### 4、课程思政目标:

引导学生认识 vi 编辑器在 Linux 运维中的重要性，培养脚踏实地、精益求精的工匠精神；在练习 vi 操作的过程中，培养学生的耐心和专注力；通过处理误操作，培养学生冷静应对问题、主动解决问题的能力。

## 重难点内容

重点: vi 编辑器的两种模式及切换方法；vi 编辑器的基础操作（打开、保存、退出）；编辑模式下的光标移动、字符删除、复制、粘贴命令。

难点: vi 两种模式的切换逻辑及操作规范（避免误操作）；编辑模式下各类快捷键命令的记忆与灵活使用；忘记模式切换时的应急处理方法；vi 编辑器的常见误操作排查。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师演示 vi 编辑器编辑文本文件的操作，提问“大家有没有发现，vi 编辑器和我们之前用的 nano 编辑器不一样，它不能直接输入内容，这是为什么？”“vi 编辑器作为 Linux 下最常用的编辑器，有什么优势？”，引出本节课 vi 编辑器的学习。

2. 讲授: 结合课件讲解 vi 编辑器的定义、特点及适用场景 (命令行下编辑、轻量高效); 重点讲解 vi 编辑器的两种核心模式——输入模式 (可输入内容) 和编辑模式 (可执行操作命令), 详细说明两种模式的切换方法 (编辑模式→输入模式: i、a、o 等键; 输入模式→编辑模式: Esc 键); 讲解 vi 编辑器的基础操作——打开文件 (vi 文件名)、保存文件 (:w)、退出编辑器 (:q)、保存并退出 (:wq)、强制退出 (:q!); 逐一讲解编辑模式下的操作命令: 光标移动 (上下左右键、h/j/k/l 键、行首行尾快捷键)、字符删除 (x、dd 等键)、复制 (yy 键)、粘贴 (p 键); 讲解 vi 编辑器的常见误操作 (如忘记切换模式、输入乱码、非正常关闭文件) 及处理方法。
3. 演示: 教师演示 vi 编辑器的打开、创建文件; 演示输入模式与编辑模式的切换, 强调 Esc 键的作用; 演示编辑模式下的光标移动, 对比不同光标移动快捷键的差异; 演示字符删除、复制、粘贴操作; 演示 vi 编辑器的保存、退出、强制退出操作; 演示常见误操作的处理方法 (如输入乱码后切换模式、非正常关闭文件后删除.swp 文件)。
4. 实践: 学生上机实践, 完成以下任务: (1) 使用 vi 编辑器创建、打开文本文件; (2) 熟练实现输入模式与编辑模式的切换; (3) 练习编辑模式下的光标移动, 尝试不同的光标移动快捷键; (4) 练习字符删除、复制、粘贴操作, 编辑文本内容; (5) 练习 vi 编辑器的保存、退出、强制退出操作; (6) 模拟常见误操作, 练习处理方法。教师巡视指导, 及时纠正学生的模式切换错误, 解答学生疑问。
5. 总结: 教师回顾本节课 vi 编辑器的关键知识点, 梳理两种模式的切换方法和核心操作命令, 强调常见误操作的处理技巧; 布置课后任务: 熟练掌握 vi 编辑器的所有基础操作, 使用 vi 编辑一个简单的文本文件, 记录操作过程中的快捷键使用情况。

# 实验十：vim 文本编辑器(3 课时)

## 教学目标

### 1. 知识目标:

了解 vim 编辑器与 vi 编辑器的区别及优势；掌握 vim 编辑器的语法高亮功能及触发条件；熟练掌握 vim 的块选择功能（Visual Block）及操作方法；了解 vim 的多文件编辑、多窗口功能及用法；掌握 vim 的基础配置（显示行号、缩进）及文件恢复方法。

### 2. 能力目标:

能使用 vim 编辑器打开、编辑文本文件，观察语法高亮效果；能使用 vim 的块选择功能批量编辑多行文本；能使用 vim 实现多文件编辑、多窗口操作；能修改 vim 的基础配置，优化编辑体验；能处理 vim 编辑器的文件恢复问题。

### 3. 素质目标:

培养“高效编辑、灵活运用工具”的习惯；形成“主动探索工具高级功能、优化操作流程”的意识；树立“精益求精、追求高效”的操作理念。

### 4. 课程思政目标:

引导学生认识 vim 编辑器的功能优势，培养创新探索精神和高效意识；在练习块选择、多窗口操作的过程中，培养学生的逻辑思维和全局观念；通过优化 vim 配置，培养学生追求卓越、精益求精的工匠精神。

## 重难点内容

重点: vim 与 vi 编辑器的区别及优势；vim 的语法高亮功能；vim 的块选择功能及操作方法；vim 的基础配置。

难点: vim 块选择功能的操作技巧（批量编辑多行内容）；多文件编辑时的文件切换与管理；多窗口的创建与关闭、窗口大小调整；vim 基础配置的修改方法及效果验证。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师回顾上节课 vi 编辑器的操作，提问“vi 编辑器虽然高效，但编辑代码时没有颜色区分，看起来很不方便，有没有什么方法可以优化？”“如果我们需要同时编辑多个文本文件，vi 编辑器能实现吗？”，引出本节课 vim 编辑器的学习，说明 vim 是 vi 的增强版，功能更强大。

2. 讲授: 结合课件讲解 vim 编辑器与 vi 编辑器的区别及优势 (语法高亮、块选择、多文件编辑等); 讲解 vim 的语法高亮功能及触发条件 (编辑代码文件时自动触发); 详细讲解 vim 的块选择功能 (Visual Block) 的操作方法 (Ctrl+v 进入块选择模式、选择内容、执行编辑操作); 讲解 vim 的多文件编辑功能 (vim 文件名 1 文件名 2), 包括文件切换 (:n、:N)、保存单个文件、保存所有文件; 讲解 vim 的多窗口功能, 包括创建窗口 (:sp、:vsp)、切换窗口 (Ctrl+w)、关闭窗口 (:q); 讲解 vim 的基础配置 (显示行号: :set nu、缩进设置: :set tabstop=4), 说明临时配置与永久配置的区别; 讲解 vim 的文件恢复方法, 处理非正常关闭文件后的.swp 文件。

3. 演示: 教师演示 vim 编辑器打开 C 语言、Shell 脚本文件, 观察语法高亮效果; 演示 vim 的块选择功能, 批量选中多行文本、批量删除、批量添加内容; 演示 vim 的多文件编辑, 打开多个文件并实现文件切换、保存; 演示 vim 的多窗口创建、切换、关闭及窗口大小调整; 演示 vim 的基础配置, 设置显示行号和缩进距离; 演示 vim 的文件恢复方法, 删除.swp 文件并恢复文本内容。

4. 实践: 学生上机实践, 完成以下任务: (1) 使用 vim 编辑 C 语言、Shell 脚本文件, 观察语法高亮效果; (2) 练习 vim 的块选择功能, 批量编辑多行文本 (批量添加注释、批量删除内容); (3) 使用 vim 打开多个文件, 练习文件切换、保存操作; (4) 练习 vim 的多窗口创建、切换、关闭及窗口大小调整; (5) 修改 vim 的基础配置, 设置显示行号和缩进距离; (6) 模拟非正常关闭文件, 练习文件恢复方法。教师巡视指导, 解答学生在块选择、多窗口操作中遇到的疑问。

5. 总结: 教师回顾本节课 vim 编辑器的关键知识点, 梳理 vim 与 vi 的区别、核心高级功能及基础配置方法; 布置课后任务: 熟练掌握 vim 的块选择、多文件编辑、多窗口操作, 修改 vim 的永久配置, 优化自己的编辑环境, 记录实用的 vim 操作技巧。

# 实验十一：GCC 编译器使用(3 课时)

## 教学目标

### 1. 知识目标:

了解 GCC 编译器的定义、功能及适用场景 (C/C++ 程序编译) ; 掌握 Ubuntu 系统下 GCC 编译器的安装方法 (apt 命令) ; 熟练掌握 GCC 编译 C 语言源文件的基本命令及参数; 了解可执行程序的运行方法; 掌握 GCC 编译常见错误及排查方法; 了解 Linux 下 Python 程序的运行方法。

### 2. 能力目标:

能在 Ubuntu 系统下安装 GCC 编译器 (若未安装) ; 能使用 vim 创建 C 语言源文件; 能使用 GCC 命令编译 C 语言源文件, 生成可执行程序并运行; 能解读 GCC 编译常见错误 (语法错误、链接错误) , 并排查解决; 能在 Linux 下创建并运行简单的 Python 程序。

### 3. 素质目标:

培养“代码编写规范、编译操作严谨”的习惯; 形成“遇到编译错误自主排查、反复调试”的问题解决能力; 树立“注重细节、精益求精”的实操理念, 养成编译前检查代码、运行前验证程序的良好习惯。

### 4. 课程思政目标:

引导学生认识 GCC 编译器在开源生态中的重要作用, 尊重开源技术和开发者成果, 树立知识产权保护意识; 在代码调试、错误排查的过程中, 培养坚韧不拔、持之以恒的钻研精神; 结合 C 语言、Python 程序在 Linux 下的运行, 引导学生认识信息技术自主可控的重要性, 激发科技报国的责任感。

## 重难点内容

重点: Ubuntu 系统下 GCC 编译器的安装方法; GCC 编译 C 语言源文件的基本命令及参数使用; 可执行程序的运行方法; GCC 编译常见错误 (语法错误、链接错误) 的排查方法; Linux 下 Python 程序的运行方法。

难点: GCC 编译参数的灵活运用 (如指定输出文件名、多文件编译) ; 编译错误信息的解读与排查 (尤其是链接错误) ; Python 程序在 Linux 下的运行环境配置及常见问题处理; 代码调试的思路与技巧。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师回顾上节课 vim 编辑器的操作, 提问“我们使用 vim 编写了 C 语言代码, 这些代码如何才能运行起来?”“Linux 系统下有哪些工具可以编译 C/C++ 程序?”, 引出本节课 GCC 编译器的学习, 说明 GCC 在 Linux 程序开发中的核心作用。
2. 讲授: 结合课件讲解 GCC 编译器的定义、功能及适用场景, 介绍 GCC 在开源生态中的地位; 详细讲解 Ubuntu 系统下 GCC 编译器的安装方法 (sudo apt update、sudo apt install gcc), 说明安装后的验证方法 (gcc --version); 讲解 GCC 编译 C 语言源文件的基本命令 (gcc 源文件名 -o 可执行文件名), 介绍常用参数 (-Wall 显示警告信息、-g 用于调试); 讲解可执行程序的运行方法 (./可执行文件名); 分类讲解 GCC 编译常见错误 (语法错误、链接错误) 的表现形式及排查思路, 举例说明错误解读方法; 简单讲解 Linux 下 Python 程序的运行方法 (python3 文件名.py), 说明 Python 环境的基本配置。
3. 演示: 教师演示 GCC 编译器的安装与验证; 演示使用 vim 创建简单的 C 语言源文件 (如 Hello World 程序); 演示 GCC 编译命令的使用, 包括指定输出文件名、显示警告信息; 演示可执行程序的运行, 观察运行结果; 演示故意编写带有语法错误、链接错误的代码, 解读编译错误信息, 演示排查与修改过程; 演示 Linux 下 Python 程序的创建与运行, 排查简单的运行错误。
4. 实践: 学生上机实践, 完成以下任务: (1) 检查 Ubuntu 系统中 GCC 是否安装, 未安装则完成安装与验证; (2) 使用 vim 创建 2 个 C 语言源文件 (简单计算程序、输入输出程序); (3) 使用 GCC 命令编译源文件, 指定输出文件名, 查看编译过程中的警告信息; (4) 运行可执行程序, 验证运行结果正确性; (5) 故意修改代码制造语法错误、链接错误, 练习错误排查与修改; (6) 创建简单的 Python 程序, 在 Linux 下运行并验证结果。教师巡视指导, 及时解答学生在编译、调试中遇到的疑问, 纠正不规范操作。
5. 纠错: 针对学生实践中出现的常见错误 (如 GCC 安装失败、编译命令拼写错误、代码语法错误解读不清、Python 程序运行环境配置问题等), 集中讲解错误原因及解决方法, 让学生现场修改验证, 强化记忆。
6. 总结: 教师回顾本节课 GCC 编译器的安装、编译、调试方法, 梳理常见错误及排查技巧, 对比 Linux 下 C 语言与 Python 程序的运行差异; 布置课后任务: 编写一个简单的 C 语言多文件程序, 使用 GCC 编译并运行, 记录编译命令和运行结果; 编写一个 Python 程序, 实现简单的数据处理功能, 在 Linux 下运行, 记录操作过程。

# 实验十二：文件打包与解压缩(3 课时)

## 教学目标

### 1. 知识目标:

理解文件打包与解压缩的概念及核心区别；掌握 Linux 系统下常用打包解压缩命令（tar、zip、unzip）的用法及参数含义；了解不同压缩格式（.tar、.tar.gz、.tar.bz2、.zip）的特点及适用场景；掌握打包解压缩的高级操作（加密、批量处理、指定解压路径）；了解打包解压缩过程中的常见错误及排查思路。

### 2. 能力目标:

能使用 tar 命令完成文件/目录的打包、压缩与解压缩操作，灵活运用不同参数组合；能使用 zip、unzip 命令完成文件的压缩、解压缩及加密操作；能根据需求选择合适的压缩格式，实现批量打包解压缩；能排查打包解压缩过程中的常见错误（如格式不兼容、权限不足、文件损坏）；能实现打包解压缩后的文件校验与备份结合。

### 3. 素质目标:

培养“数据备份规范、压缩操作高效”的习惯；形成“根据文件大小和需求选择合适工具”的意识；树立“数据安全优先、操作前先确认”的实操理念，养成打包后验证、解压缩前检查的良好习惯。

### 4. 课程思政目标:

引导学生认识打包解压缩在数据备份、传输中的重要作用，培养数据安全意识和责任担当；在批量操作、参数组合使用中，培养学生的逻辑思维和高效工作能力；通过加密压缩操作，引导学生理解信息安全的细节要求，树立隐私保护和数据保密的意识。

## 重难点内容

重点: tar 命令的用法及参数组合（打包、压缩、解压缩、指定路径）；zip、unzip 命令的基础使用及加密操作；不同压缩格式的特点及选择方法；打包解压缩的基本流程与规范。

难点: tar 命令多参数组合的灵活运用（如-c、-v、-f、-z、-j 的组合）；不同压缩格式之间的转换与兼容问题；加密压缩的操作技巧及密码管理；打包解压缩过程中常见错误的排查与解决。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师提问“我们在传输多个文件或备份大量数据时, 逐个操作效率很低, 如何才能快速实现多个文件的统一管理和传输?”“不同压缩格式的文件, 解压方法有什么区别?”, 结合数据备份、文件传输的实际场景, 引出本节课文件打包与解压缩的核心内容, 说明其在 Linux 运维中的实用价值。

2. 讲授: 结合课件讲解打包与解压缩的概念, 明确两者的核心区别(打包是合并文件, 不压缩; 解压缩是还原压缩文件, 释放空间); 详细讲解 tar 命令的用法, 包括打包 (tar -cvf)、压缩 (tar -zcvf、tar -jcvf)、解压缩 (tar -xvf、tar -zxvf) 的参数含义及操作格式, 举例说明不同参数组合的适用场景; 讲解 zip、unzip 命令的基础用法, 包括压缩文件 (zip)、解压缩文件 (unzip)、加密压缩 (zip -e)、指定解压路径 (unzip -d) 的操作; 对比不同压缩格式 (.tar、.tar.gz、.tar.bz2、.zip) 的压缩率、速度及适用场景, 引导学生根据需求选择; 讲解打包解压缩过程中的常见错误(如权限不足、文件损坏、格式错误)及排查思路。

3. 演示: 教师演示 tar 命令的打包、压缩与解压缩操作, 对比不同参数组合的效果, 强调 -v 参数的查看作用; 演示 zip 命令压缩单个、多个文件, unzip 命令解压缩文件及指定解压路径, 演示加密压缩与解密操作; 演示不同压缩格式的转换方法, 对比压缩率差异; 演示常见错误(如 tar 命令漏写 -f 参数、unzip 解压缩格式不兼容、权限不足)的排查与解决过程; 演示打包与备份结合的操作, 将打包文件复制到备份目录。

4. 实践: 学生上机实践, 完成以下任务: (1) 使用 tar 命令打包指定目录, 生成 .tar 文件, 验证打包结果; (2) 使用 tar 命令对打包文件进行 gzip、bzip2 压缩, 生成 .tar.gz、.tar.bz2 文件, 对比压缩大小; (3) 使用 tar 命令解压缩不同格式的压缩文件, 指定解压路径; (4) 使用 zip 命令压缩多个文件, 实现加密压缩, 使用 unzip 命令解密并解压缩; (5) 模拟常见错误, 练习排查与解决方法; (6) 根据需求选择合适的压缩格式, 完成指定文件的打包压缩与解压缩, 记录操作过程。教师巡视指导, 及时纠正学生的参数使用错误, 解答疑问, 强调操作规范。

5. 总结: 教师回顾本节课打包解压缩的核心命令 (tar、zip、unzip) 及参数用法, 梳理不同压缩格式的选择技巧, 总结常见错误及排查方法; 布置课后任务: 熟练掌握所有打包解压缩命令, 完成指定目录的备份与压缩操作, 尝试不同压缩格式的对比测试, 记录压缩率与速度; 编写简单的操作笔记, 整理常用命令组合。

# 实验十三：Linux 进程管理(3 课时)

## 教学目标

### 1、知识目标:

理解 Linux 进程的定义、生命周期及基本状态（运行、就绪、睡眠、终止）；掌握进程查看命令（ps、top、pstree）的用法及结果解读；熟练掌握进程控制命令（kill、killall、pkill）的用法及参数含义；了解进程优先级的概念及调整方法（nice、renice 命令）；了解后台进程的管理方法（&、jobs、fg、bg 命令）。

### 2、能力目标:

能使用 ps、top、pstree 命令查看系统进程信息，解读进程状态、PID、CPU/内存占用等关键参数；能使用 kill、killall、pkill 命令终止指定进程、异常进程，避免误杀系统进程；能使用 nice、renice 命令调整进程优先级，优化系统资源分配；能管理后台进程，实现进程的后台运行、前台切换与终止；能排查进程运行中的常见问题（如进程卡死、资源占用过高）。

### 3、素质目标:

培养“系统监控严谨、进程操作规范”的习惯；形成“实时关注系统状态、合理分配资源”的意识；树立“谨慎操作、避免误操作”的实操理念，养成终止进程前先确认的良好习惯。

### 4、课程思政目标:

引导学生认识进程管理在 Linux 系统运维中的核心作用，培养系统思维和全局观念；在排查异常进程、优化资源分配的过程中，培养学生的责任担当和解决问题的能力；通过进程优先级调整，引导学生理解“合理分配资源、提高效率”的管理理念，培养精益求精的工匠精神。

## 重难点内容

重点: 进程查看命令（ps、top）的用法及结果解读；进程控制命令（kill）的用法及信号含义；进程状态的识别；后台进程的基本管理方法。

难点: top 命令输出结果的详细解读（CPU、内存占用分析、进程状态判断）；kill 命令信号的灵活运用（不同信号的作用差异）；进程优先级的理解与调整技巧；异常进程的判断与排查（如僵尸进程、卡死进程）；后台进程与前台进程的切换管理。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师演示系统卡顿场景, 提问“当 Linux 系统运行缓慢、卡顿甚至无响应时, 我们该如何查找原因?”“系统中正在运行的程序, 我们如何查看和管理?”, 引出本节课 Linux 进程管理的核心内容, 说明进程管理对系统稳定运行的重要性。
2. 讲授: 结合课件讲解进程的定义、生命周期 (创建、运行、就绪、睡眠、终止) 及基本状态, 举例说明不同状态的特点; 详细讲解进程查看命令: `ps` 命令 (常用参数组合 `ps -aux`、`ps -ef`) 的结果解读, 包括 PID、USER、%CPU、%MEM、COMMAND 等字段的含义; `top` 命令的用法, 实时监控系统进程, 解读 CPU、内存占用、进程排序等关键信息; `pstree` 命令的用法, 查看进程树结构, 理解进程间的父子关系; 讲解进程控制命令: `kill` 命令的基本用法 (`kill PID`)、信号含义 (如-9 强制终止、-15 正常终止), `killall`、`pkill` 命令的区别与用法 (按进程名终止进程); 讲解进程优先级的概念, `nice` 命令 (创建进程时设置优先级)、`renice` 命令 (调整已运行进程优先级) 的用法; 讲解后台进程管理: `&` 符号 (后台运行进程)、`jobs` 命令 (查看后台进程)、`fg` 命令 (后台进程切换到前台)、`bg` 命令 (前台进程切换到后台) 的操作。
3. 演示: 教师演示 `ps -aux`、`ps -ef` 命令的使用, 解读输出结果, 筛选指定进程; 演示 `top` 命令的使用, 调整排序方式 (按 CPU、内存占用), 解读进程状态, 终止异常进程; 演示 `pstree` 命令查看进程树, 识别父子进程; 演示 `kill` 命令的不同信号使用, 对比正常终止与强制终止的差异, 演示 `killall`、`pkill` 命令按进程名终止进程; 演示 `nice`、`renice` 命令调整进程优先级, 查看优先级变化; 演示后台进程的管理, 实现进程后台运行、前台切换、终止操作; 演示异常进程 (如卡死进程) 的排查与终止过程。
4. 实践: 学生上机实践, 完成以下任务: (1) 使用 `ps -aux`、`ps -ef` 命令查看系统进程, 筛选指定进程, 记录 PID、CPU/内存占用情况; (2) 使用 `top` 命令实时监控系统进程, 调整排序方式, 解读进程状态, 终止指定异常进程; (3) 使用 `pstree` 命令查看进程树, 分析父子进程关系; (4) 使用 `kill` 命令的不同信号终止进程, 对比效果; 使用 `killall`、`pkill` 命令按进程名终止进程; (5) 使用 `nice` 命令创建后台进程并设置优先级, 使用 `renice` 命令调整其优先级; (6) 练习后台进程的管理, 实现后台运行、前台切换、终止操作; (7) 模拟进程卡死场景, 排查并终止异常进程。教师巡视指导, 强调 `kill` 命令的使用安全, 禁止误杀系统核心进程, 及时解答学生疑问。
5. 总结: 教师回顾本节课进程管理的核心知识点, 梳理进程查看、控制、优先级调整、后台管理的关键命令及用法, 强调 `kill` 命令的使用安全和异常进程的排查技巧; 布置课后任务: 熟练掌握所有进程管理命令, 编写进程监控操作笔记, 记录 `top` 命令的详细解读; 模拟系统卡顿场景, 完成异常进程排查与优化, 记录操作过程; 练习后台进程的复杂管理, 掌握 `jobs`、`fg`、`bg` 命令的灵活运用。

# 实验十四：Shell 脚本基础(3 课时)

## 教学目标

### 1、知识目标:

理解 Shell 脚本的定义、功能及适用场景；掌握 Shell 脚本的基本结构（脚本声明、注释、命令序列）；熟练掌握 Shell 脚本的创建、编辑、运行方法；了解 Shell 脚本中的变量（定义、赋值、使用）及简单运算；掌握 Shell 脚本的条件判断基础（if 语句）及简单循环（for 循环）用法；了解 Shell 脚本的常见错误及排查方法。

### 2、能力目标:

能使用 vim 编辑器创建、编辑 Shell 脚本，添加脚本声明和注释；能熟练运行 Shell 脚本（sh 命令、chmod +x 权限运行）；能定义和使用 Shell 变量，完成简单的数值运算；能编写包含条件判断和简单循环的 Shell 脚本，实现批量操作；能排查 Shell 脚本运行中的常见错误（语法错误、权限错误、命令错误）。

### 3、素质目标:

培养“脚本编写规范、注释清晰”的习惯；形成“批量操作高效、逻辑严谨”的意识；树立“脚本运行前先测试、错误排查逐行验证”的实操理念，养成良好的脚本编写与调试习惯。

### 4、课程思政目标:

引导学生认识 Shell 脚本在 Linux 自动化运维中的重要作用，培养自动化思维和高效工作理念；在脚本编写、调试过程中，培养学生的逻辑思维和坚韧不拔的钻研精神；通过批量操作脚本的编写，引导学生理解“高效办公、精益求精”的工作态度，激发创新意识。

## 重难点内容

重点: Shell 脚本的基本结构；Shell 脚本的创建、编辑与运行方法；Shell 变量的定义、赋值与使用；简单条件判断和 for 循环的用法。

难点: Shell 脚本的运行权限配置；变量的赋值规则及特殊变量的使用；条件判断的语法格式（括号、逻辑运算符）；循环语句的逻辑设计及脚本调试技巧。

## 教学方法与过程

教学方法: 讲授法+演示法+实践法

教学过程:

1. 导入: 教师回顾上节课进程管理的命令，提问“如果我们需要重复执行一系列 Linux

命令，逐个输入效率很低，有没有什么方法可以将这些命令整合起来，一键执行？”“Linux 系统中，如何实现命令的自动化批量执行？”，引出本节课 Shell 脚本基础的学习，说明 Shell 脚本在自动化运维中的实用价值。

2. 讲授: 结合课件讲解 Shell 脚本的定义、功能及适用场景（自动化批量操作、命令整合、简单任务调度）；讲解 Shell 脚本的基本结构，包括脚本声明（#!/bin/bash）、注释（#开头）、命令序列的作用及编写规范；详细讲解 Shell 脚本的创建与编辑方法（vim 编辑器），运行方法（sh 脚本名、./脚本名，强调./运行需添加执行权限）；讲解 Shell 变量的定义、赋值与使用方法，说明变量的命名规则，举例说明数值运算的实现（expr 命令、\${()}表达式）；讲解简单的条件判断（if 语句）语法格式，包括条件表达式的写法、if-else 的使用场景；讲解简单 for 循环的用法，实现批量处理任务；讲解 Shell 脚本常见错误（语法错误、权限不足、命令拼写错误）的排查思路（sh -x 调试命令）。

3. 演示: 教师演示使用 vim 创建 Shell 脚本，添加脚本声明、注释和简单命令序列（如目录创建、文件查看）；演示 Shell 脚本的两种运行方法，对比权限配置的差异（chmod +x 脚本名）；演示变量的定义、赋值与使用，完成简单的数值运算；演示包含 if 条件判断的脚本编写与运行（如判断文件是否存在）；演示 for 循环脚本的编写与运行（如批量创建文件）；演示使用 sh -x 命令调试脚本，排查语法错误和逻辑错误。

4. 实践: 学生上机实践，完成以下任务：（1）使用 vim 创建 Shell 脚本，添加脚本声明和注释，编写简单命令序列，练习两种运行方法；（2）定义多个 Shell 变量，实现变量赋值与使用，完成简单的数值运算；（3）编写包含 if 条件判断的脚本，判断指定文件/目录是否存在，输出对应提示信息；（4）编写 for 循环脚本，实现批量创建文件或批量查看进程的功能；（5）模拟脚本运行中的常见错误，使用 sh -x 命令调试排查；（6）编写一个整合多个命令的脚本，实现简单的系统信息查看功能。教师巡视指导，及时纠正学生的脚本语法错误，解答变量使用、循环逻辑中的疑问，强调脚本编写规范。

5. 总结: 教师回顾本节课 Shell 脚本的核心知识点，梳理脚本结构、运行方法、变量使用及条件判断、循环语句的用法，强调脚本调试的重要性；布置课后任务：熟练掌握 Shell 脚本的编写与运行，编写 3 个不同功能的脚本（变量运算、条件判断、for 循环）；尝试编写一个自动化备份脚本，整合打包解压缩命令，记录脚本编写与调试过程。

# 实验十五：综合考查(3 课时)

## 教学目标

### 1、知识目标:

全面回顾本学期 Linux 实训的核心知识点，包括 Linux 系统基础、目录文件管理、文本编辑、用户权限管理、进程管理、GCC 编译、打包解压缩及 Shell 脚本基础；巩固各模块的核心命令及操作规范，明确各知识点之间的关联的应用场景。

### 2、能力目标:

能综合运用所学 Linux 命令和操作，独立完成综合性实训任务；能快速排查操作过程中的常见错误，灵活运用调试技巧；能整合多个模块的知识，编写简单的自动化脚本，实现批量操作；能规范记录操作过程，整理实训报告，提升综合实操能力和问题解决能力。

### 3、素质目标:

培养“综合运用知识、逻辑严谨操作”的习惯；形成“自主排查错误、高效完成任务”的意识；树立“总结归纳、查漏补缺”的学习理念，养成良好的实训报告编写习惯。

### 4、课程思政目标:

引导学生全面总结本学期的学习成果，培养总结归纳、查漏补缺的学习态度；在综合任务完成过程中，培养学生的责任担当和抗压能力；通过知识的综合运用，激发学生对 Linux 技术的学习兴趣，树立科技报国、精益求精的理想信念。

## 重难点内容

重点: 各模块核心命令的综合运用；常见错误的快速排查与解决；简单自动化脚本的编写与运行；综合任务的规划与高效完成。

难点: 多个模块知识的整合运用（如 Shell 脚本整合打包解压缩、进程管理命令）；复杂错误的排查思路与调试技巧；综合任务的规划与时间管理。

## 教学方法与过程

教学方法: 实践法+指导法+总结法

教学过程:

1. 导入: 教师回顾本学期 Linux 实训的全部内容，梳理各实验模块的核心知识点和关联关系，说明本次综合考查的目的（全面检验学习成果、巩固知识、提升综合运用能力），明确考查任务、要求和时间安排，激发学生的积极性和主动性。

2. 任务布置: 结合本学期所学内容, 布置综合性实训任务 (单人独立完成), 任务包含以下模块: (1) 系统基础操作: 查看系统信息、网络配置, 验证系统运行状态;

(2) 目录文件管理: 创建多级目录、批量创建文件, 修改文件权限和所有者, 查找指定文本内容; (3) 文本编辑: 使用 vim 编辑文本文件和 Shell 脚本, 添加注释, 修改内容; (4) GCC 编译: 创建 C 语言源文件, 使用 GCC 编译并运行, 排查编译错误; (5) 打包解压缩: 将指定目录打包压缩, 加密压缩后解压缩到指定路径; (6) 进程管理: 查看系统进程, 终止指定进程, 管理后台进程; (7) Shell 脚本: 编写一个简单的自动化脚本, 整合至少 3 个模块的命令, 实现批量操作; (8) 实训报告: 记录所有操作步骤、命令及运行结果, 整理常见错误及解决方法。

3. 实践考查: 学生单人独立上机实践, 在规定时间内完成所有综合任务, 教师巡视指导, 不直接告知答案, 仅针对学生的疑问进行引导性提示, 重点关注学生的命令使用规范性、错误排查能力和知识综合运用能力, 记录学生的完成情况。

4. 总结与点评: 教师对本次综合考查进行总结, 点评学生的整体完成情况, 肯定优点, 指出普遍存在的不足和改进方向; 全面回顾本学期 Linux 实训的核心知识点, 梳理各模块的关联关系, 强调实操过程中的规范和技巧; 引导学生总结自己的学习收获和不足, 明确后续的学习方向。