



理工科计算机基础指北



应急管理大学（筹）（华北科技学院）

——风暴战队

目录

【前言】	3
1. 为什么会有这份《指北》?	3
2. 本指南是什么?	3
3. 邀请你一同完善	3
第一章 认识计算机	3
第一节: 外接设备与接口认识	3
第二节: 键盘篇: 你的双手, 就是最高效的快捷键	5
第三节: 认识计算机硬件: 你的算力引擎	7
第四节: Windows 11 初体验: 桌面、开始菜单、任务栏是什么?	10
第五节: Ubuntu 初体验: 进入开源世界的命令行圣殿	12
第六节: 常见的文件格式: 你的数字文件百科全书	14
第二章: 选择你的武器——代码编辑器与 IDE	21
第三章: 安全与礼仪——做一个靠谱的实验室公民	22
黄金法则: 先谷歌, 再问人。	23
1. 第一步: 自助 (Google/Firefox)	23
2. 第二步: 求助 (当你确实无法解决时)	23
附录 1 ASCII 码对照表	27
附录二 常用键盘快捷键表	28
附录三: 日常	30
一、效率神器类	30
二、媒体与截屏类	30
三、安全与系统类	31
四、信息获取与 AI 类	31
一些让你受益终身的电脑使用小提醒	31
后续补充内存	32
致谢	33
版权声明	35

【前言】

1. 为什么会有这份《指北》？

笔者为某大学理工科实验室的视觉组成员，在给新同学做培训时，我们发现大量同学由于初高中微机课的缺乏，计算机基础缺失，甚至不知道 ctrl 键的位置，严重影响后续的培训和学习进度。

2. 本指南是什么？

这是一份以 Windows 11 为主、Ubuntu 为辅的《计算机基础指北》。它完全面向日常学习和科研的实际操作，旨在帮你快速上手，弥补知识与技能断层。请注意，它的定位与大学的《计算机基础》课程不同

3. 邀请你一同完善

我个人行文风格较为随性，随心所欲，随性而干，若您在阅读中发现条理不清或任何错误，非常欢迎批评指正。让我们共同完善这份给后来者的参考资料。

第一章 认识计算机

前言：当同学们看到此书时，笔者默认大家具有打开电脑，阅读 docs 或者 pdf 的能力，顾不过多赘述，以下内容主要认识打开计算机后，可以看到的東西，例如键盘，桌面应用图标等等.....

第一节： 外接设备与接口认识

外接设备与接口认识：为你的电脑连接整个世界

作为视觉组的同学，你不可避免地要连接各种设备：传输数据的 U 盘、扩展屏幕的显示器、读取素材的 SD 卡、以及至关重要的网线。认识这些接口，是你高效使用电脑的第一步。

一、 数据传输接口：电脑的“进出口”

接口主要负责连接存储设备和其他外设，进行数据交换。

1. USB 接口家族：最常见的“万金油”

- 核心功能：连接键盘、鼠标、U 盘、移动硬盘、手机、摄像头等几乎一切外设。
- 如何识别与选择：
- USB-A 型：那个“怎么插都插不反，但总要试两次”的经典方口。最常见，但速度有差异。



- USB-C 型：小巧的椭圆形接口，正反插都可以。这是现在的绝对主流和未来。
- 一个重要特性：很多电脑的 USB-C 口也同时是充电口和视频输出口。



实验室建议：

- 传大文件（如数据集、模型）时，优先使用 USB-C 口或蓝色的 USB-A 口，速度会快很多。
- 如果新电脑只有 USB-C 口，而你的 U 盘是 USB-A 口，你需要一根 USB-C to A 的转接线或扩展坞。

2. 雷电接口

- 外形：通常也采用 USB-C 的物理形态，旁边会有一个闪电 ⚡ 标志。
- 特点：速度极快，功能极强。可以外接高性能显卡坞和高速硬盘阵列。你可以把它理解为“超级加强版”的 USB-C。

3. SD / TF 卡读卡槽

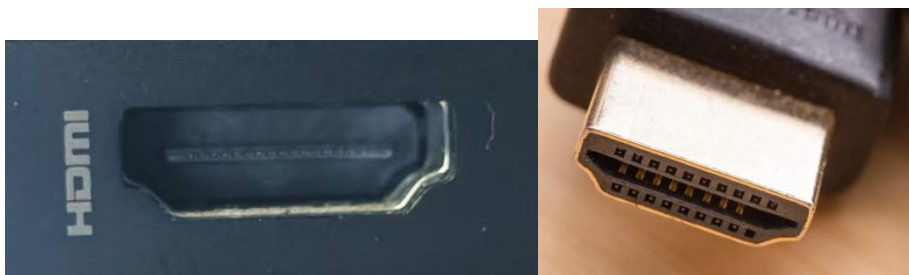
- 外形：一个扁平的缝隙，通常在笔记本电脑侧面。（部分电脑没有）
- 功能：直接从相机、无人机等设备的存储卡中读取素材。

二、 显示输出接口：为你开启更多“视野”

实验室工作，多一块屏幕，效率翻倍。以下是常见的视频接口，优先级从高到低：

1. HDMI

- 外形：类似 USB-A 但更宽，梯形设计，不会插反。
- 地位：最普及、最通用的接口。显示器、投影仪、电视基本都有。一根线同时传输视频和音频。



2. DisplayPort

- 外形：类似 HDMI，但一角是直角，另一角有切角。
- 地位：性能通常优于 HDMI，是电脑显示器领域的主流。同样传输视频和音频。



3. VGA

- 外形：蓝色的、带有两排针脚的梯形接口。
- 地位：老旧的模拟信号接口。画质差，不支持音频。实验室原则：除非万不得已，否则不用它。当你看到会议室或老投影仪只有这个口时，才会需要一根 VGA 转接线。



特殊说明：对于绝大部分游戏本和一小部分轻薄本能插电使用一定要插电使用！电池的作用只是为了让你面对意外断电应急使用

实验室实战场景：

你的笔记本电脑只有一个 HDMI 口，但想接两台显示器怎么办？

答案：使用扩展坞。通过一个 USB-C 或雷电接口扩展出多个 HDMI/DP 接口，同时还能接网线、U 盘等。

三、 有线网络接口：稳定可靠的“信息高速公路”

- RJ-45 网线接口：那个看起来像大型电话线的插口。
- 为什么重要：
 - 稳定性：比 Wi-Fi 更稳定，不掉线，延迟低。
 - 速度：在相同网络条件下，网线快于 WiFi

四、 音频接口

- 台式机通常有两个：
 - 麦克风图标：接耳机上的麦克风。
 - 耳机图标：接耳机或音箱。
- 笔记本已合二为一，只有一个耳机/麦克风二合一接口。

认识接口就像认识工具，能让你在需要时快速找到对的“钥匙”。从此，连接显示器、传输数据、接入网络都将不再是难题。大胆地去摸索和连接吧（在断电情况下更安全）！

第二节：键盘篇：你的双手，就是最高效的快捷键

作为视觉组的成员，你的大部分时间都将与键盘和鼠标为伴。很多人觉得键盘只是打字工具，但对于追求效率的开发者而言，键盘是你与计算机“直接对话”的通道，熟练使用键盘能让你脱离繁琐的点击，行云流水，快人一步。

本章节，我们将重新认识这个你最熟悉又可能最陌生的伙伴。

一、 键盘分区：你的指挥中心

放眼望去，键盘可以被划分为几个功能区，就像你实验室桌面的不同区域：

1. 主键盘区：

- 核心力量：字母、数字、符号键。这是你输入代码、文档和命令的主力。
- “幕后英雄”修饰键：它们是键盘操作的灵魂，通常不单独使用，而是与其他键组合，产生强大的效果。
 - Ctrl：控制键。位于键盘最左下或最右下。它是快捷键的绝对核心，比如 Ctrl + C（复制）、Ctrl + V（粘贴）。

- **Alt:** 替代键。在 Windows 中常用于激活菜单，也参与大量软件快捷键。
- **Shift:** 上档键。用于输入大写字母和键位上方的符号。
- **Win:** Windows 键。印有 Windows 徽标，快速打开开始菜单，是系统级快捷操作的入口。

- 注意：在 Ubuntu 系统中，这个键通常被称为 Super 键，功能类似。

2. 功能键区：

- 就是顶部的 F1 ~ F12。它们的功能因软件而异，是“情景快捷键”。
- 实验室高频示例：
 - **F2:** 在文件管理器中选择一个文件后按它，可以重命名。
 - **F11:** 在绝大多数浏览器和许多 IDE 中，可以快速进入/退出全屏模式，为你争取宝贵的屏幕空间。
 - **F5:** 刷新 或 运行/调试代码。

3. 导航键区：

- 包含方向键和 Insert, Delete, Home, End, Page Up, Page Down。
- 写代码时的神器：
 - **Home/End:** 让光标瞬间跳到当前行的行首/行尾。
 - **Ctrl + Home/Ctrl + End:** 让光标瞬间跳到整个文档的开头/结尾。
 - **Page Up/Page Down:** 翻页。

4. 数字小键盘区：

- 用于快速输入数字，在处理数据时非常方便。注意其上的 Num Lock 键，灯亮时输入数字，灯灭时变为导航键。



二、实验室生存必备快捷键（肌肉记忆系列）

下面这些，请务必像记住你的学号一样记住它们。它们将贯穿你从写文档到调试代码的全过程。

通用核心操作（Win & Ubuntu 通用）：

- **Ctrl + C:** 复制
- **Ctrl + V:** 粘贴
- **Ctrl + X:** 剪切
- **Ctrl + Z:** 撤销 - 当你误操作时，这是你的“后悔药”。
- **Ctrl + Y 或 Ctrl + Shift + Z:** 恢复 - 把撤销的操作再恢复回来。
- **Ctrl + S:** 保存 - 求求你，写几行代码就下意识按一下！这是对抗程序崩溃、系统死机的最强护身符。

- **Ctrl + A:** 全选

文本/代码编辑专用：

- **Ctrl + F:** 查找 - 在代码里找一个变量/函数名？用它！
- **Ctrl + H:** 替换 - 想把所有 `color` 改成 `colour`？用它！
- **Shift + 方向键:** 以字符/行为单位选中文本，比用鼠标拖拽精准得多。
- **Ctrl + 方向键左/右:** 以词为单位移动光标。
- **Ctrl + Backspace / Ctrl + Delete:** 以词为单位删除。

系统与窗口管理：

- **Alt + Tab:** 在打开的应用程序间快速切换 - 程序员摸鱼（划掉）高效工作的必备技能。
- **Win + D:** 一键显示桌面，瞬间隐藏所有窗口，方便你找桌面文件。
- **Win + E:** 快速打开文件资源管理器。
- **Win + L:** 瞬间锁屏 - 离开座位时保护隐私和数据安全的好习惯。
- **Alt + F4:** 关闭当前活动窗口。

ctrl + 鼠标滚轮: 在浏览器、Word、代码编辑器里，能快速放大/缩小字体，看小字或调整界面显示比例超方便。

Ubuntu 特色补充：

- **Super 键:** 按一下打开活动概览，快速搜索和启动应用。
- **Ctrl + Alt + T:** 快速打开终端 - 这是你在 Ubuntu 下的主战场，请把这个快捷键刻在 DNA 里。

三、指法建议

我们不是专业打字员，不要求盲打。但请尽量让双手放在键盘上，尤其是左手，应该常驻在 **Ctrl, Alt, Shift** 这几个修饰键周围。这样当你需要按快捷键时，右手可以继续操作鼠标或按其他键，效率倍增。

第三节：认识计算机硬件：你的算力引擎

你可以把电脑主机想象成一个实验室团队，每个硬件部件都是团队中的一名成员，各司其职，协同完成复杂的计算任务。

一、CPU：团队的“总指挥”

它是什么？

中央处理器，是电脑的“大脑”。它负责执行程序指令、处理数据、协调所有其他硬件的工作。

核心比喻：实验室的“博后”或“导师”

核心数：就像团队里有几个博后。核心越多，能同时独立处理的任务就越多。**8 核 CPU** 就像有 **8 个博后** 可以同时算题。

线程数：类似于一个博后可以“一心二用”，同时指导两个实验思路（超线程技术）。线程数越多，并行处理能力越强。

频率：就像博后思考和解算问题的速度，单位是 **GHz**。频率越高，处理单个任务的速度越快。



实验室工作场景：

负责程序的逻辑控制、数据预处理、大部分科学计算（如果没用到 GPU 的话）。

当你打开任务管理器，看到所有 CPU 核心都接近 100% 时，说明你的计算任务已经把 CPU 跑满了，这就是性能瓶颈。

二、GPU：团队的“大规模并行计算兵团”

它是什么？

图形处理器，最初是为处理图像而设计。但现在，它是负责大规模并行计算的专家。

核心比喻：实验室的“大量研究生”

CUDA 核心/流处理器：这就是“研究生”的数量。一个 GPU 有数千上万个这样的核心，而 CPU 通常只有几个到几十个核心。

特点：每个核心能力不如 CPU（博后）强，但数量极多，且擅长同时执行大量简单的、重复性的任务。

实验室工作场景：

深度学习模型训练与推理：这是 GPU 最核心的用途。海量的矩阵运算可以完美地拆分成成千上万个任务，交给 GPU 的数千个核心同时计算，速度比 CPU 快几个数量级。

科学计算与仿真：越来越多的科学计算库支持 GPU 加速。

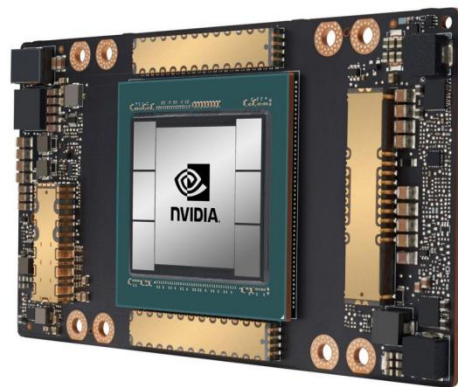
关键指标：显存

这是 GPU 的“专属工作内存”。你的模型、数据和中间计算结果都必须能放进显存里。如果显存不足，程序就会报错或无法运行。对于深度学习，显存大小往往比核心数量更重要。

CPU vs. GPU 核心区别总结：

CPU：少数几个“博后”，能力全面，负责复杂逻辑和任务调度。

GPU：成千上万个“研究生”，擅长简单重复的大规模并行计算。



三、内存：团队的“公共实验台”

它是什么？

随机存取存储器，是 CPU 的“临时工作区”。

核心比喻：实验室的“中央实验台”

所有正在运行的程序、打开的数据、需要 CPU 处理的文件，都必须先从硬盘加载到内存这个“实验台”上，CPU 才能快速地进行操作。

特点：速度快，但断电后数据会丢失。

实验室工作场景：

当你用 Python 加载一个几十 GB 的数据集时，这个数据集就占用了等量的内存。

如果你同时打开很多程序，或者处理超大文件，内存占用会很高。当内存被占满时，系统会变得极其卡顿，因为要用硬盘来模拟内存（虚拟内存），而硬盘速度慢得多。



建议：做机器学习/深度学习的电脑，16GB 内存是起步，32GB 或以上推荐配置。

四、 硬盘：团队的“资料档案馆”

它是什么？

用于长期存储操作系统、软件、文档、数据和所有文件。

核心比喻：实验室的“档案柜”或“仓库”

所有不正在被 CPU 处理的东西都存放在这里。断电后数据不会丢失。

两种主要类型：

机械硬盘：

像老式的光盘机，有机械臂和磁碟。

优点：便宜，容量大。适合做冷数据备份（不常访问的数据）。

缺点：速度慢，怕震动。

固态硬盘：

像超快的 U 盘，纯电子结构，没有机械部件。

优点：读写速度极快（是机械硬盘的几倍到几十倍），抗震。

缺点：价格较贵。

实验室工作场景：

系统、软件、当前项目必须装在 SSD 上！这能让你开机、打开软件、加载数据的速度快上几个量级。

机械硬盘可以用来存储大量的历史数据、备份、电影等不要求速度的文件。



五、 主板：团队的“办公楼”

它是什么？

一块巨大的电路板，上面有各种插槽和接口。

它提供了电源、网络 and 座位（插槽），让 CPU、GPU、内存、硬盘等所有部件能够安装在一起，并相互通信、协同工作。

关键作用：

决定了扩展性：主板上有几个 GPU 插槽、几个内存插槽、几个硬盘接口，决定了你未来能升级到什么程度。

芯片组：相当于大楼的“中枢神经系统”，决定了支持哪种 CPU、最大内存速度等。



六、 电源：团队的“能源中心”

它是什么？

为所有硬件提供稳定、纯净的电力。

核心比喻：实验室的“供电系统”

非常重要但常被忽视！一个劣质或功率不足的电源，会导致电脑不稳定、蓝屏、重启，甚至在电压波动时烧毁其他贵重部件（如 CPU、GPU）。

建议：千万不要在电源上省钱。为高性能电脑（尤其是有独立 GPU 的）选择一个功率足够、品牌可靠的电源。

实验室硬件选购与使用小贴士

如何查看自己电脑的配置？

Windows: 任务管理器 -> “性能” 选项卡。

Ubuntu: 使用 `lscpu`、`nvidia-smi` (查看 GPU)、`free -h` (查看内存)、`lsblk` (查看硬盘) 等命令。

如何判断性能瓶颈？

做任务时，打开任务管理器 (Windows) 或 `htop` (Ubuntu)：

CPU 100% -> 计算瓶颈。

内存 100% -> 内存瓶颈，需要加内存或优化代码。

GPU 使用率低 -> 可能不是 GPU 瓶颈，可能是数据从 CPU/硬盘到 GPU 的传输慢了，或者代码没写好。

硬盘灯常亮，系统卡顿 -> 硬盘读写瓶颈，考虑升级 SSD。

给新生的建议：

优先顺序：GPU (显存) > 内存 > SSD > CPU > 其他。

如果你的工作重度依赖深度学习，一块显存足够大的 NVIDIA GPU 是性价比最高的投资。

确保你的主板能支持你选择的 CPU 和内存，并且电源能扛得住整个系统的峰值功耗。

了解你的硬件，就是了解你手中工具的能力边界。这能让你在分配计算任务、申请服务器资源、甚至自己配置电脑时，做出最明智的选择。

第四节:Windows 11 初体验：桌面、开始菜单、任务栏是什么？

当你第一次成功登录实验室的 Windows 11 电脑，映入眼帘的就是这个界面。别慌，我们把它拆解成三个核心部分来理解：桌面、开始菜单和任务栏。你可以把它们想象成你未来工位的桌面、万能工具抽屉和手边最顺手的工具架。

一、 桌面：你的工作台面

- 它是什么？

桌面就是你开机后看到的主背景区域。它就像你实验室的物理桌面，你可以在上面放置最常用、最紧急的文件和工具。

- 你能用它做什么？

1. 放置快捷方式：将你最常用的软件（如 VS Code、PyCharm）、文件夹（如你的项目目录）或文件，创建快捷方式放在桌面上，方便快速打开。

2. 临时文件暂存：从 U 盘或下载文件夹里拖出来的文件，可以暂时放在桌面进行处理。

3. 个性化：右键点击桌面 -> “个性化”，可以更换你喜欢的壁纸，让你心情愉悦。

- 实验室潜规则：

- 不要堆满图标！就像杂乱的物理桌面会降低效率一样，一个混乱的虚拟桌面也会让你找不到东西。只保留最核心的几个快捷方式。

- 重要文件不要长期放在桌面！桌面文件通常默认在 C 盘，如果系统崩溃需要重装，C 盘数据有丢失风险。请及时将重要数据归档到 D 盘。

二、 开始菜单：你的应用程序库与控制中心

- 它是什么？

点击任务栏正中的 **Windows** 徽标按钮 就会弹出的那个面板。它是你电脑的总控制台和所有程序的起点。



- 它里面有什么？（两大区块）

1. “已固定”区域：

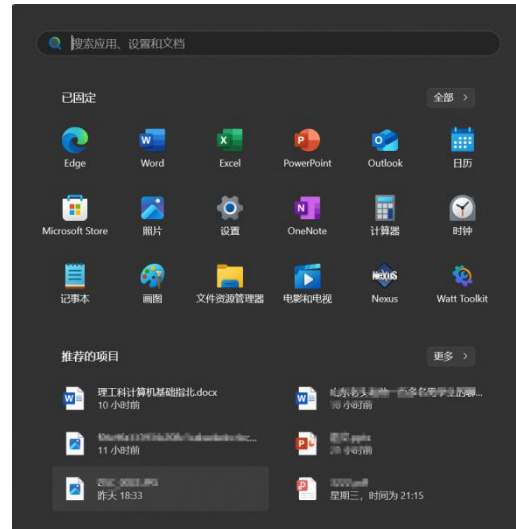
- 位置：打开后上半部分。
- 功能：这里是自定义的“神器收藏夹”

你可以把最常用的软件（如浏览器、VS Code、终端）钉在这里。

- 如何操作：找到任何一个软件，右键它 -> 选择“固定到开始屏幕”。

2. “所有应用”列表：

- 位置：点击“所有应用”按钮。
- 功能：这里是电脑上安装的所有程序的完整清单，按字母顺序排列。当你找不到一个软件时，就来这里搜索或翻找。



3. 推荐区域：

- 位置：下半部分。
- 功能：**Windows** 会在这里显示你最近打开的文件和安装的新应用。可以快速找回最近的工作。
- 实验室高效技巧：
 - 别急着点鼠标，直接敲键盘！想打开一个软件？直接按 **Win** 键，然后键入软件名（比如 **code** 或 **vs code** 找 VS Code，**chr** 找 Chrome 浏览器），然后按回车。这比用鼠标找快十倍！
 - 把你所有的开发环境（VS Code, PyCharm, QT 等）和常用工具固定到“已固定”区域，打造你的专属工作台。

三、 任务栏：你的多任务管理中心

- 它是什么？

就是屏幕底部那条通常包含开始菜单、搜索、程序图标和时间的长条。它是你的实时工作状态指示器。

- 它为什么重要？

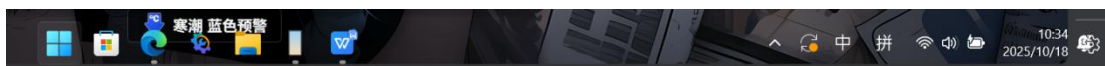
1. 程序坞：你可以把最最常用的程序图标（如文件资源管理器、浏览器）固定到任务栏上（右键程序 -> “固定到任务栏”），实现一键启动。

2. 运行中程序的“码头”：所有正在运行的程序，都会在任务栏上有一个图标。你可以直接点击切换不同程序，就像在实验室里转身拿不同工具一样方便。

3. 系统状态一目了然：

- 网络、音量、电池：在右下角，可以快速查看和调整。
- 通知中心：点击右下角的日期时间区域，可以弹出通知中心和日历快捷视图。
- 快速设置：点击网络、音量等图标，可以快速弹出面板，调整 Wi-Fi、蓝牙、

亮度等。



- 实验室高效技巧：
 - 任务视图与虚拟桌面：点击任务栏上的“任务视图”图标（或直接按 **Win + Tab**），你可以：
 - 看到所有打开的窗口。
 - 在底部点击“新建桌面”，创建多个虚拟桌面。比如：一个桌面专门写代码，一个桌面专门查资料和看文档，一个桌面用来沟通（微信、邮件）。用 **Ctrl + Win + 左/右方向键** 可以快速切换，实现工作区的完美隔离。
 - 固定你的核心武器：将 **VS Code**、终端（**Windows Terminal** 或 **Command Prompt**）和文件资源管理器固定到任务栏，这是你作为开发者的“三件套”。

第五节：Ubuntu 初体验：进入开源世界的命令行圣殿

当你选择进入 **Ubuntu** 系统后，首先迎接你的通常是一个简洁、现代的桌面环境。别被它的平静外表迷惑，其下蕴藏着强大的力量。对于视觉组的同学来说，**Ubuntu** 是你未来与服务器、高性能计算和纯命令行环境打交道的基石。

我们以最主流的 **Ubuntu Desktop**（带图形界面）为例。

一、桌面与环境：似曾相识，又与众不同

Ubuntu 的图形界面主要由 顶栏、程序坞 和 活动概览 组成。

1. 顶栏

- 位置：屏幕顶部。
- 功能：类似于 **Windows** 的任务栏+菜单栏。左边是活动按钮，中间是当前程序的菜单（如 文件、编辑），右边是系统状态区（网络、声音、电源、用户等）。

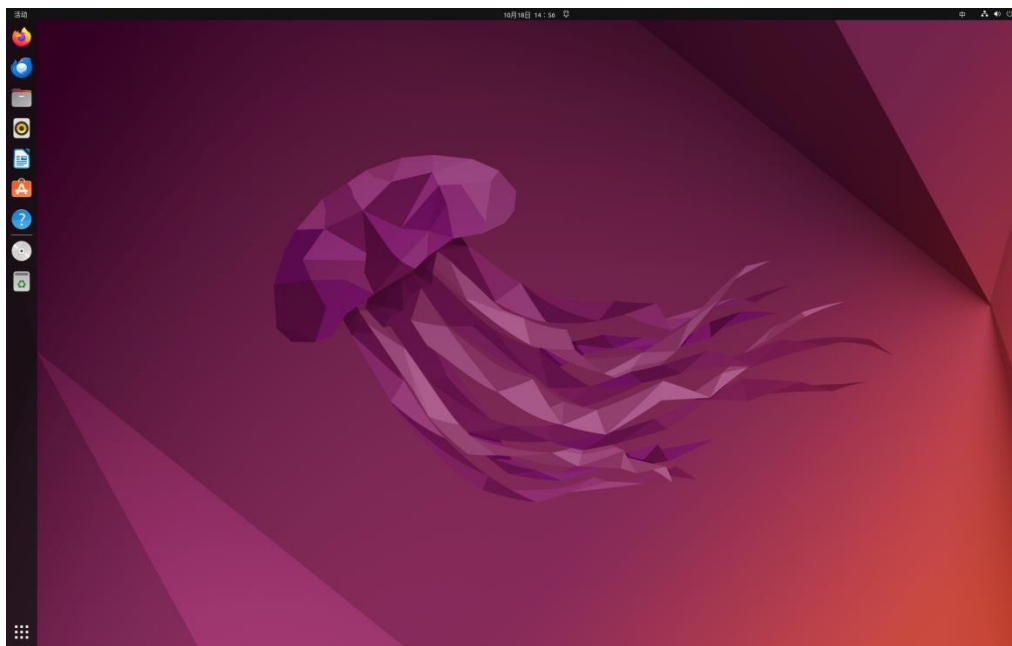
2. 程序坞

- 位置：屏幕左侧（默认）。
- 功能：这相当于你的“快速启动栏”。上面固定着你最常用的程序，正在运行的程序下方会有一个小指示条。
- 你能做什么：
 - 点击图标打开应用。
 - 将常用程序固定上去：打开一个程序后，右键其在程序坞上的图标，选择“添加到收藏夹”。
- 实验室核心三件套：请首先找到并固定“文件”、“终端”和“文本编辑器”。

3. 活动概览

- 如何打开：点击顶栏最左边的“活动”按钮，或直接按键盘上的 **Super** 键。
- 它是什么：这是你多任务管理的核心！按下后，你会看到：
 - 所有打开的窗口平铺在屏幕上。
 - 左侧是固定的程序坞。
 - 顶部是一个全局搜索框。
- 实验室高效技巧：

· 超级搜索：直接按 **Super** 键，然后输入你想找的任何东西——应用名、文件名、甚至是一个计算式，它都能快速给出结果。这是你启动应用最快的方式，比鼠标找快得多！



二、 文件管理器：你的数据导航员

1. 如何打开：点击程序坞上的“文件”图标，或按 **Super** 键搜索“Files”。
2. 它是什么：这就是 **Ubuntu** 的“我的电脑”或“文件资源管理器”。
3. 重要概念：
 - 家目录：打开文件管理器，你首先进入的就是你的“家目录”。这相当于 **Windows** 的 `C:\Users\[你的用户名]` 文件夹。你的所有个人文档、下载、图片、视频默认都存放在这里的对应文件夹下。
 - 根目录：在左侧栏，有一个“其他位置”，点击进去可以看到“计算机”。这里才是真正的系统根目录，相当于 **Windows** 的 `C:\`。
 - 实验室潜规则：
 - 路径表示不同：在 **Ubuntu** 中，路径用正斜杠 `/` 而不是反斜杠 `\`。你的家目录路径是 `/home/你的用户名/`。
 - 没有 C 盘 D 盘：所有硬盘、U 盘、网络驱动器都会“挂载”到根目录下的某个文件夹（如 `/media`），而不是以盘符存在。

万物皆文件

三、 终端：你与机器对话的“魔法窗口”

这是你未来在实验室最亲密的伙伴，没有之一。

1. 为什么必须学它？
 - 服务器上没有图形界面，全靠它操作。
 - 绝大多数开发工具、环境配置、软件安装，在终端里一条命令就能搞定，比图形

界面更高效、更强大。

2. 如何打开：

- 快捷键：Ctrl + Alt + T —— 请把这个快捷键刻在 DNA 里！
- 鼠标方式：按 Super 键，搜索

“terminal”或“终端”。

3. 生存级初体验：

打开后，你会看到一个等待输入的命令提示符，通常以 用户名@电脑名:~\$ 结尾。那个闪烁的光标在等待你的指令。

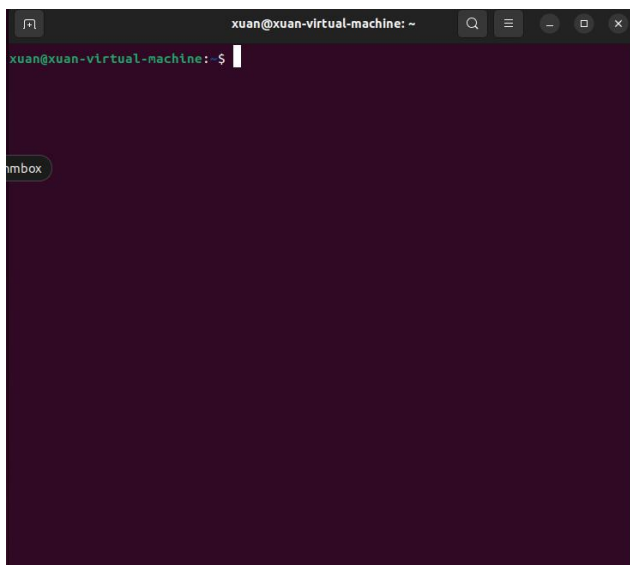
- pwd：输入后按回车。它会告诉你“你现在在哪个文件夹？”（Print Working Directory）。

- ls：输入后按回车。它会列出当前文件夹里所有可见的文件和子文件夹（List）。

- cd 文件夹名：用于切换目录（Change Directory）。比如 cd Downloads 进入下载文件夹。

- cd ..：返回上一级目录。.. 代表上级目录。

- cd ~ 或 cd：瞬间回到你的家目录。



四、 软件安装：全新的“应用商店”体验

在 Ubuntu 中，安装软件的主要方式不是去浏览器下载 .exe 文件，而是通过软件中心或终端命令。

1. 图形化方式：Ubuntu Software

- 就像手机的应用商店，搜索、点击安装即可。简单，但有时版本更新较慢。

2. 命令行方式（实验室主流！）：apt 命令

- 这是 Ubuntu 自带的超级软件管理器。
- 更新软件源列表：sudo apt update（sudo 是获取管理员权限，需要你输入密码）。
- 安装一个软件：sudo apt install 软件包名
- 实验室例子：安装我们必备的代码编辑器 VS Code。

```
```bash
sudo apt update
sudo apt install code
```
```

- 优势：一条命令，自动解决所有依赖，干净利落。

结语：

初次接触 Ubuntu，不必追求掌握所有细节。请先熟悉图形界面的操作，并勇敢地打开终端，把上面几条生存命令当成咒语一样敲几遍。感受这种直接、高效的交互方式。

很快你就会发现，这个“黑色窗口”是你最得力的助手。

第六节：常见的文件格式：你的数字文件百科全书

文件扩展名（后缀）就像文件的“姓氏”，告诉了你它的性质和用途。

一、 文档与文本类

| 格式
后缀 | 全称 / 格式说明 | 核心特点 | 适用场景举例 | 常用打开程序 |
|----------|-----------------|--|----------------------|---|
| .pdf | 便携式文档格式 | 1. 排版固定, 跨设备 / 平台显示效果一致; 2. 无法直接编辑, 适合保护内容; 3. 通用性极强, 几乎所有设备可打开。 | 论文终稿提交、产品手册分发、合同签署 | Adobe Acrobat、浏览器 (Chrome/Edge)、WPS |
| .docx | Word 文档格式 | 1. 可自由编辑, 支持字体、图片、表格等丰富格式; 2. 格式兼容性强, 主流办公软件均支持。 | 论文草稿撰写、工作报告、会议纪要、信件 | Microsoft Word、WPS Office、LibreOffice |
| .pptx | PowerPoint 演示格式 | 1. 以“幻灯片”为单位, 支持动画、过渡效果; 2. 适合可视化呈现信息, 突出重点。 | 学术汇报、会议演讲、产品介绍、培训课件 | Microsoft PowerPoint、WPS Office、LibreOffice |
| .xlsx | Excel 表格格式 | 1. 支持数据录入、公式计算、图表生成; 2. 可高效处理大量结构化数据, 自动分析结果。 | 数据统计、财务报表、实验数据记录、考勤表 | Microsoft Excel、WPS Office、LibreOffice |

| | | | | |
|------|---------------|---|------------------------|--------------------------------------|
| .txt | 纯文本格式 | 1. 仅包含文字, 无任何格式 (如字体、颜色); 2. 体积极小, 打开速度快, 通用性 100%。 | 存储代码片段、简单日志、纯文字笔记 | 记事本 (Windows)、VS Code、TextEdit (Mac) |
| .md | Markdown 标记语言 | 1. 用简单符号 (如 #、-) 定义格式, 易上手; 2. 可快速转换为 PDF、HTML 等格式, 适合轻量文档。 | 项目 README 文件、技术笔记、博客草稿 | VS Code (需插件)、Typora、Notion |

实验室提示:
与他人分享最终版文档时, 优先使用 PDF, 以确保对方看到的和你一模一样。
写项目说明文档 (README) 时, 强烈推荐使用 Markdown (.md)。

二、 数据与存储类

一、演示与表格类格式

| 格式后缀 | 全称 / 说明 | 核心特点 | 适用场景 | 常用打开程序 |
|-------|-----------------|-------------------------------|----------------|--------------------------------------|
| .pptx | PowerPoint 演示文稿 | 以幻灯片为单位, 支持动画 / 过渡效果, 适合可视化呈现 | 学术汇报、会议演讲、培训课件 | Microsoft PowerPoint、WPS、LibreOffice |
| .xlsx | Excel 工作簿 | 支持数据录入、公式计算、图表生成, 高效处理结构化数据 | 统计报表、实验数据、考勤表 | Microsoft Excel、WPS、LibreOffice |

二、数据交换与存储类格式

| 格式后缀 | 全称 / 说明 | 核心特点 | 适用场景 | 常用打开程序 |
|-----------|------------------|--|---------------------|---------------------------|
| .csv | 逗号分隔值文件 | 纯文本数据格式，用逗号分隔记录，跨程序 (Python/Excel/MATLAB) 兼容 | 数据交换、表格数据导出 | Excel、记事本、Python (pandas) |
| .json | JavaScript 对象表示法 | 轻量级数据格式，键值对结构，易读易解析，适合 Web 传输和配置文件 | API 数据、配置文件、轻量数据库 | 浏览器、VS Code、Python |
| .xml | 可扩展标记语言 | 标签定义数据结构，格式较冗长，多用于旧式软件配置或数据存储 | 传统系统配置、数据存档 | 浏览器、记事本、VS Code |
| .h5/.hdf5 | 层次数据格式 | 存储大量复杂数据，支持高效压缩，像文件系统一样管理数据集 | 科学计算、深度学习模型 / 数据集 | Python (h5py 库)、专业查看工具 |
| .npy/.npz | NumPy 数组文件 | Python NumPy 专用二进制格式，读写速度远快于文本格式 | NumPy 数组存储、科学计算中间数据 | Python (NumPy 库) |

三、压缩与打包类格式

| 格式后缀 | 全称 / 说明 | 核心特点 | 适用场景 | 常用打开程序 |
|----------|----------|--------------------------------|-----------------|-------------------------|
| .zip/.7z | 压缩文件 | 最通用的压缩格式，打包多文件 / 文件夹，节省空间，便于传输 | 文件备份、邮件附件、资源打包 | 7-Zip、Bandizip、系统自带压缩工具 |
| .rar | RAR 压缩文件 | 压缩率较高，但通用性不及 ZIP，需第三方软件解压 | 大文件压缩、需要高压缩率的场景 | 7-Zip、WinRAR |

实验室提示：

处理小型、通用的表格数据，用 CSV。

处理大型矩阵、数组或模型权重，用 NPY/NPZ 或 HDF5。

永远不要用 Word 或 Excel 来保存用于程序计算的原始数据，请使用 CSV 等纯文本格式。

三、 代码与程序类

| 格式后缀 | 全称 / 说明 | 核心特点 | 适用场景 | 常用打开程序 |
|-----------|--------------|-------------------------------------|----------------|---------------------------|
| .py | Python 源代码文件 | 包含 Python 程序的源代码，是 Python 开发的基础文件格式 | 脚本编写、程序开发、算法实现 | VS Code、PyCharm、IDLE |
| .cpp / .c | C++/C 源代码文件 | 分别包含 C++ 或 C 语言的源 | 系统开发、高性能程序、嵌入式 | VS Code、Qt Creator、Visual |

| | | | | |
|-----------------|------------------------|--|--|---|
| | | 代码, 用于编译
型程序开发 | 开发 | Studio |
| .ipynb | Jupyter Notebook
文件 | 交互式 Python
计算笔记本, 整
合代码、输出结
果、图文说明,
支持分步执行 | 数据分析、机器
学习实验、教学
演示 | Jupyter
Notebook、
JupyterLab、VS
Code |
| .exe | 可执行文件 | Windows 系统
下的应用程序,
经编译后可直
接运行, 无需依
赖编程环境 | 软件分发、应用
程序运行 | 双击直接运行 |
| .sh | Shell 脚本文件 | Linux/Unix 系
统 (含
Ubuntu) 的脚
本文件, 包含一
系列终端命令,
可批量执行 | 系统自动化操
作、批量任务处
理 | 终端 (Linux/Unix) |
| .yaml
/ .yml | YAML 配置文件 | 人性化的数据
序列化格式, 语
法简洁, 适合存
储配置信息 | Docker 配置、
Anaconda 环
境配置、程序参
数设置 | VS Code、记事本 |

四、 图像与多媒体类

| 格式后缀 | 全称 / 说明 | 核心特点 | 适用场景 | 常用打开程序 |
|--------------|---------------|---|---------------------------|---------------------------------|
| .jpg / .jpeg | 联合图像专家
组格式 | 采用有损压
缩, 文件体积
小, 画质损失
可控, 兼容所
有设备 | 彩色照片、自
然风光图、日
常图像存储 | 系统自带图片查看
器、Photoshop、浏
览器 |

| | | | | |
|------|-------------------|--|-------------------|-----------------------------------|
| .png | 便携式网络图形格式 | 无损压缩, 支持透明背景, 边缘清晰度高, 无画质损失 | 屏幕截图、图表、Logo、图标设计 | 系统自带图片查看器、Photoshop、VS Code (预览) |
| .gif | 图形交换格式 | 支持静态 / 动态图像, 最多仅 256 种颜色, 动画效果简单 | 表情包、简单动画、低色彩需求场景 | 浏览器、系统自带图片查看器、微信 / QQ (预览) |
| .bmp | 位图格式 | 未经压缩, 文件体积极大, 画质无损失, 兼容性差 | 早期图像存储、特定工业软件场景 | 系统自带图片查看器、Photoshop |
| .mp4 | MPEG-4 Part 14 格式 | 主流视频容器格式, 压缩率高, 画质清晰, 支持音频 + 视频同步, 跨平台兼容性强 | 视频录制、影视播放、网络视频传输 | 播放器 (PotPlayer、VLC)、浏览器、系统自带视频播放器 |
| .avi | 音频视频交错格式 | 较老旧的视频格式, 无统一压缩标准, 文件体积通常比 MP4 大, 兼容性逐步下降 | 早期视频文件、特定设备导出场景 | 播放器 (PotPlayer、VLC)、系统自带视频播放器 |

实验室提示:

保存照片用 JPG。

保存算法生成的图表、示意图、带透明背景的图片, 用 PNG。

总结: 记住这几个“之最”

最通用的文档: .pdf

最通用的数据：.csv
最通用的压缩包：.zip
最通用的图片：.jpg (照片) 和 .png (图表)
最通用的代码：.py / .ipynb (Python 环境)

第二章：选择你的武器——代码编辑器与 IDE

Qt Creator：Qt C++ 图形界面开发的“官方重器”

为什么实验室会用到它？

当你的项目需要开发带图形界面的桌面应用程序，或者进行嵌入式设备（如机器人、车载中控）的界面开发时，Qt Creator 就是实验室的不二之选。它是 Qt 框架的“官方 IDE”，为 Qt 开发提供了无与伦比的支持。

它的核心优势是什么？

1. Qt 原生支持，无缝集成
 - 它天生就为 Qt 而生。无论是代码补全、语法高亮，还是对 Qt 特有的信号与槽（Signals & Slots）、元对象系统等特性的理解，都做到了极致。
 - 内置 Qt Designer：这是一个可视化拖拽界面设计器。你可以直接拖放按钮、文本框等控件来设计 UI，然后自动生成对应的 .ui 文件，极大提高了 GUI 开发的效率。
2. 强大的跨平台开发能力
 - 使用 Qt 和 Qt Creator 编写的代码，可以轻松编译并运行在 Windows、Linux、macOS 乃至嵌入式系统上，真正做到“一次编写，到处编译”。
3. 集成调试与可视化工具
 - 除了常规的调试功能，它还集成了 Qt 特有的调试工具，如 Qt Assistant（集成文档）、Linguist（国际化翻译工具）等，形成了一套完整的开发套件。

基本设置：从安装到第一个界面

1. 安装（关键步骤）
 - 前往官网：<https://www.qt.io/download-open-source>
 - 下载 Qt Online Installer。运行后，在组件选择步骤：
 - 勾选 Qt Creator。
 - 在 Qt 版本下，选择一个最新的稳定版（如 Qt 6.5.0）。
 - 最重要的一步：展开该版本，务必勾选你所在操作系统的编译器。对于：
 - Windows：勾选 MSVC 2019 64-bit 和/或 MinGW 64-bit。
 - Ubuntu：勾选 GCC 相关的套件。
 - 勾选 CMake 和 Ninja（现代 Qt 项目常用的构建工具）。
2. 配置中文（可选）
 - 打开 Qt Creator。
 - 顶部菜单：Edit -> Preferences (macOS: Qt Creator -> Settings)。
 - 左侧 Environment -> General -> 在 Language 下拉菜单中选择 简体中文，重启生效。

3. 创建你的第一个 Qt 项目

因项目而异，故不过多阐述。

还有 VScode pycharm 等等，但与此书（计算机基础）不相符，故不过多阐述。

第三章：安全与礼仪——做一个靠谱的实验室公民

技术和知识可以慢慢学，但良好的安全习惯和团队礼仪，是你从第一天起就必须具备的素养。本章内容，请务必牢记于心。

一、安全第一：保护你的数字生命线

在数字世界，麻痹大意就是最大的风险。

1. 定期备份！备份！备份！

- 核心原则：任何只存在于一个地方的数据，都等于不存在。
- 为什么？ 硬盘损坏、电脑丢失、误删除、病毒勒索.....任何一种情况都能让你数月的研究成果瞬间清零。
- 如何做？（3-2-1 备份原则）
 - 3 份副本：总共有 3 份数据副本。
 - 2 种介质：保存在 2 种不同的存储介质上（如电脑硬盘 + 移动硬盘/U 盘 + 实验室 NAS）。
 - 1 份异地：至少有 1 份备份放在不同的物理地点（如实验室一份，宿舍一份）。
- 实验室实践：定期将重要数据同步到实验室的 NAS 或服务器上。

2. 设置强密码并保密

- 强密码标准：长度大于 10 位，包含大小写字母、数字、特殊符号。避免使用生日、姓名等易猜解信息。
- 不要重复使用密码：实验室服务器、电脑、个人邮箱等账号请使用不同密码。
- 绝不分享密码：即使是师兄师姐索要，也绝不提供。正确的做法是请管理员为你单独创建账号。

3. 离开锁屏：最简单的安全习惯

- 快捷键：Win + L (Windows) / Super + L (Ubuntu)
- 为什么？ 防止他人无意或恶意查看你的实验数据、代码，甚至利用你登录的账号进行操作。只要离开座位，无论时间长短，请务必锁屏。

二、实验室礼仪：尊重他人，就是尊重自己

实验室的公共资源（电脑、服务器）是大家的，你的行为会影响整个团队。

1. 用完公共电脑，请恢复环境

- 场景：你使用了实验室的公共测试电脑或他人的电脑。
- 操作：
 - 关闭你打开的所有软件和文件。
 - 如果修改了配置（如环境变量、路径），请尽量恢复原状。

- 登出你的个人账户（如果登录了）。
 - 理念：为下一位使用者提供一个干净、原始的环境，就像你希望用到的一样。
2. 不要在服务器上跑死循环或耗尽资源
 - 场景：你在实验室的共享服务器上运行你的代码。
 - 禁忌：
 - 死循环：一个写错的 `while True` 循环可能会占满整个 CPU 核心，让其他同学的任务无法执行。
 - 耗尽内存：加载远超服务器内存容量的数据，会导致系统因内存不足而崩溃，害死所有人的任务。
 - 最佳实践：
 - 在本地小规模测试通过后，再提交到服务器。
 - 使用 `htop` (Ubuntu) 或任务管理器 (Windows) 监控你的程序资源占用。
 - 如果使用任务调度系统（如 Slurm），请按规定申请合理的计算资源。
 3. 提交代码前，先测试！
 - 场景：你要将代码提交到团队的共享仓库（如 GitLab）。
 - 禁忌：直接提交一个自己都没运行过、不知道是否报错的代码。
 - 最佳实践：
 - 至少确保它能跑通，不报致命错误。
 - 如果可能，确保它不会破坏已有的功能。
 - 写清楚本次提交的注释，让他人一目了然。

三、如何提问：聪明地获取帮助

遇到问题张口就问，是效率最低且最招人反感的方式。学会“聪明地提问”，能让你更快地解决问题，并赢得他人的尊重。

黄金法则：先谷歌，再问人。

1. 第一步：自助（Google/Firefox）

- 将报错信息直接复制到搜索引擎中搜索。
- 在 Stack Overflow、GitHub Issues、CSDN 等技术社区查找类似问题。
- 大概率（超过 90%）你遇到的问题，世界上已经有人遇到并解决了。

2. 第二步：求助（当你确实无法解决时）

在向师兄师姐或老师提问时，请务必一次性提供以下所有信息，这能节省双方大量时间：

- 环境
 - 操作系统（Windows 11 / Ubuntu 22.04?）
 - 软件/语言版本（Python 3.8 / PyTorch 1.12?）
 - 如果是硬件问题，说明设备型号。
- 你做了什么
 - 清晰、简洁地描述你为了达成什么目标，执行了哪些步骤。
 - 例如：“我为了在项目 A 中安装 pandas 库，在终端里输入了 `pip install pandas`。”
- 报错信息
 - 完整地、原封不动地复制粘贴终端的报错信息。不要自己概括！
 - 提供截图有时更直观。
- 你已尝试过的方法

- 告诉他你已经搜索过了，并简要说明你尝试过的解决方案和结果。
- 例如：“我搜了这个报错，尝试了用 `pip3` 代替 `pip`，也换了清华的镜像源，但都报同样的错误。”

一个糟糕的提问 vs 一个优秀的提问

- 糟糕的：“师兄，我代码报错了，怎么办？”（让人无从下手）
- 优秀的：
 - “师兄你好，我的代码在导入 `numpy` 时报错了，能帮我看下吗？”
 - 环境：Ubuntu 22.04, Python 3.10。
 - 操作：我在终端里直接输入 `python` 然后 `import numpy`。
 - 报错：`ImportError: Something wrong with numpy...`（附上完整报错截图）。
 - 已尝试：我谷歌了这个问题，看到有人说用 `conda install numpy` 可以解决，我试了之后还是同样的错误。”

结语：

技术和知识决定了你能力的下限，而安全意识和团队礼仪则决定了你职业发展的上限。从今天起，做一个让同伴放心、让师兄师姐省心的靠谱实验室成员。

常见问题及解决方案

重启！！！！！重启可以解决百分之八十的电脑问题！！！！！！

本章节旨在解决你在使用电脑时最可能遇到的“小麻烦”。它们不深奥，但若不会处理，会严重影响工作效率和心情。请放心，解决它们通常不需要高深技术。

一、基础使用问题

问题 1：电脑开机后，屏幕不亮（黑屏）

- 解决思路：顺着信号路径，从简单到复杂逐一排查。
 1. 检查电源：确认主机和显示器的电源线都插紧了，并且插座有电（可以插手机充电器测试）。
 2. 检查显示器：确认显示器电源开关已打开（指示灯亮着）。尝试按一下显示器的菜单键，看是否能调出菜单，以判断是显示器问题还是信号问题。
 3. 检查连接线：检查连接主机和显示器的视频线（如 `HDMI` 线）两端是否都插紧了。
 4. 终极妙招：重启一下。长按电源键强制关机，再重新启动。

问题 2：U 盘或移动硬盘插上电脑后，没有反应

- 解决思路：判断是设备问题还是电脑问题。
 1. 换个 `USB` 口：拔下来，换一个电脑上的 `USB` 接口再试。特别是不要插在机箱前面板，试试后面的接口。
 2. 检查设备本身：将 U 盘插到另一台电脑上试试。如果另一台电脑也能识别，问题在你的电脑；如果都不能识别，可能是 U 盘本身坏了。

3. 在系统中查看：打开“文件资源管理器”（Win+E），看看左侧栏有没有多出一个盘符，它可能没有弹窗提示，但已经识别了。

问题 3：电脑突然没有声音了

- 解决思路：从软件设置到硬件连接。

1. 检查音量：首先点击任务栏右下角的“小喇叭”，确认音量不是静音，并且音量调得足够大。

2. 检查播放设备：右键点击“小喇叭” -> “打开声音设置” -> 确保“输出设备”选择的是你正在使用的音箱或耳机（比如不是选成了显示器音频）。

3. 检查物理连接：确认音箱/耳机的线缆正确插入了电脑的音频口（通常是绿色接口）。确认音箱电源已打开。

二、文件与操作问题

问题 4：不小心把文件删除了

- 解决方案：

1. 第一时间：不要往硬盘里存入新文件！

2. 回收站：立即打开桌面上的“回收站”，找到被删的文件，右键点击 -> “还原”。文件就会回到原来的位置。

3. 如果回收站没有：如果你用了 **Shift+Delete** 彻底删除，回收站就没有了。这时需要求助实验室管理员，尝试用专业数据恢复软件，但成功率不保证。所以，重要文件务必备份！

问题 5：忘记文件保存在哪里了

- 解决思路：让系统帮你找。

1. 使用搜索：

- Windows：打开文件资源管理器（Win+E），点击右上角的搜索框，输入文件名或文件类型（如 *.pdf）。

- Ubuntu：按 Super 键，直接在顶部搜索框输入文件名。

2. 查看“最近”记录：

- 很多软件（如 Word、VS Code）的“文件”菜单里有“最近使用的文档”列表。

- 文件资源管理器的左侧导航栏也可能有“快速访问”或“最近”的记录。

问题 6：软件“卡住”或“未响应”了

- 解决方案：

1. 耐心等待：有时只是暂时卡顿，等待几十秒。

2. 强制关闭：如果一直无响应，按 **Ctrl + Alt + Delete**，选择“任务管理器”，在“进程”中找到那个卡住的软件，右键点击 -> “结束任务”。这是最干净、最安全的强制关闭方式。

三、网络问题

问题 7：能上微信，但浏览器打不开网页

- 解决思路：这通常是浏览器或 DNS 的问题。

1. 换个浏览器试试：用系统自带的 Edge 浏览器试试，如果能打开，就是你常用浏览器（如 Chrome）的配置问题。

2. 最简单有效的方法：重启你的电脑和路由器。90%的网络问题可以通过重启解决。

问题 8：Wi-Fi 信号满格，但网速很慢或断断续续

- 解决思路：

1. 远离干扰源：离路由器近一些，避开微波炉、无线座机等设备。

2. 减少占用：检查是否有程序在后台大量下载（如系统更新、游戏更新）。

3. 换种方式：如果实验室条件允许，插上网线，这是最稳定、最快的选择。

核心心法：解决问题的通用思路

遇到任何电脑问题，不要慌张，请按以下顺序尝试：

1. 重启大法：关闭程序，甚至重启电脑。这是解决电子设备问题的“万灵药”。

2. 检查连接：无论是电源线、视频线、网线还是 USB 线，都确认一下是否插稳、插对。

3. 换个试试：换个 USB 口、换个浏览器、换个软件版本，用于判断问题是局部的还是普遍的。

4. 求助搜索：将你遇到的问题用简单的语言描述，例如“电脑插 U 盘没反应”，输入搜索引擎，你会发现你绝不是一个人。

5. 清晰求助：如果自己解决不了，向他人求助时，请说清楚：“我想做什么 -> 我遇到了什么现象 -> 我已经尝试过哪些方法”。

记住，熟练解决这些小问题，是成为电脑高手的必经之路。每解决一个问题，你的经验值就+1！

附录 1 ASCII 码对照表

ASCII可显示字符（共95个）

| 二进制 | 十进制 | 十六进制 | 图形 | 二进制 | 十进制 | 十六进制 | 图形 | 二进制 | 十进制 | 十六进制 | 图形 |
|-----------|-----|------|---------|-----------|-----|------|----|-----------|-----|------|----|
| 0010 0000 | 32 | 20 | (space) | 0100 0000 | 64 | 40 | @ | 0110 0000 | 96 | 60 | ` |
| 0010 0001 | 33 | 21 | ! | 0100 0001 | 65 | 41 | A | 0110 0001 | 97 | 61 | a |
| 0010 0010 | 34 | 22 | " | 0100 0010 | 66 | 42 | B | 0110 0010 | 98 | 62 | b |
| 0010 0011 | 35 | 23 | # | 0100 0011 | 67 | 43 | C | 0110 0011 | 99 | 63 | c |
| 0010 0100 | 36 | 24 | \$ | 0100 0100 | 68 | 44 | D | 0110 0100 | 100 | 64 | d |
| 0010 0101 | 37 | 25 | % | 0100 0101 | 69 | 45 | E | 0110 0101 | 101 | 65 | e |
| 0010 0110 | 38 | 26 | & | 0100 0110 | 70 | 46 | F | 0110 0110 | 102 | 66 | f |
| 0010 0111 | 39 | 27 | ' | 0100 0111 | 71 | 47 | G | 0110 0111 | 103 | 67 | g |
| 0010 1000 | 40 | 28 | (| 0100 1000 | 72 | 48 | H | 0110 1000 | 104 | 68 | h |
| 0010 1001 | 41 | 29 |) | 0100 1001 | 73 | 49 | I | 0110 1001 | 105 | 69 | i |
| 0010 1010 | 42 | 2A | * | 0100 1010 | 74 | 4A | J | 0110 1010 | 106 | 6A | j |
| 0010 1011 | 43 | 2B | + | 0100 1011 | 75 | 4B | K | 0110 1011 | 107 | 6B | k |
| 0010 1100 | 44 | 2C | , | 0100 1100 | 76 | 4C | L | 0110 1100 | 108 | 6C | l |
| 0010 1101 | 45 | 2D | - | 0100 1101 | 77 | 4D | M | 0110 1101 | 109 | 6D | m |
| 0010 1110 | 46 | 2E | . | 0100 1110 | 78 | 4E | N | 0110 1110 | 110 | 6E | n |
| 0010 1111 | 47 | 2F | / | 0100 1111 | 79 | 4F | O | 0110 1111 | 111 | 6F | o |
| 0011 0000 | 48 | 30 | 0 | 0101 0000 | 80 | 50 | P | 0111 0000 | 112 | 70 | p |
| 0011 0001 | 49 | 31 | 1 | 0101 0001 | 81 | 51 | Q | 0111 0001 | 113 | 71 | q |
| 0011 0010 | 50 | 32 | 2 | 0101 0010 | 82 | 52 | R | 0111 0010 | 114 | 72 | r |
| 0011 0011 | 51 | 33 | 3 | 0101 0011 | 83 | 53 | S | 0111 0011 | 115 | 73 | s |
| 0011 0100 | 52 | 34 | 4 | 0101 0100 | 84 | 54 | T | 0111 0100 | 116 | 74 | t |
| 0011 0101 | 53 | 35 | 5 | 0101 0101 | 85 | 55 | U | 0111 0101 | 117 | 75 | u |
| 0011 0110 | 54 | 36 | 6 | 0101 0110 | 86 | 56 | V | 0111 0110 | 118 | 76 | v |
| 0011 0111 | 55 | 37 | 7 | 0101 0111 | 87 | 57 | W | 0111 0111 | 119 | 77 | w |
| 0011 1000 | 56 | 38 | 8 | 0101 1000 | 88 | 58 | X | 0111 1000 | 120 | 78 | x |
| 0011 1001 | 57 | 39 | 9 | 0101 1001 | 89 | 59 | Y | 0111 1001 | 121 | 79 | y |
| 0011 1010 | 58 | 3A | : | 0101 1010 | 90 | 5A | Z | 0111 1010 | 122 | 7A | z |
| 0011 1011 | 59 | 3B | ; | 0101 1011 | 91 | 5B | [| 0111 1011 | 123 | 7B | { |
| 0011 1100 | 60 | 3C | < | 0101 1100 | 92 | 5C | \ | 0111 1100 | 124 | 7C | |
| 0011 1101 | 61 | 3D | = | 0101 1101 | 93 | 5D |] | 0111 1101 | 125 | 7D | } |
| 0011 1110 | 62 | 3E | > | 0101 1110 | 94 | 5E | ^ | 0111 1110 | 126 | 7E | ~ |
| 0011 1111 | 63 | 3F | ? | 0101 1111 | 95 | 5F | _ | | | | |

电脑常用快捷键大全

| | | |
|----------------|--------------------|----------------|
| 截屏:Win+shift+s | 程序切换: Alt+Tab | 放大: Ctrl+鼠标轮 |
| 录屏:Win+G | 程序关闭: Alt+F4 | 字体加粗: Ctrl+B |
| 锁屏:Win+L | 文件命名:fn+F2 | 字体斜体:Ctrl+I |
| 切屏:Win+P | 文件删除:Ctrl+D | 字下划线:Ctrl+U |
| 设置:Win+I | 撤销删除:Ctrl+Z | 文字全选:Ctrl+A |
| 此电脑:Win+E | 打@符号: Shift+2 | 字居中:Ctrl+E |
| 回桌面:Win+D | 网页刷新:fn+F5 | 左对齐:Ctrl+L |
| 剪贴板:Win+V | 网页关闭:Ctrl+W | 右对齐:Ctrl+R |
| 放大:Win++ | 关闭找回:Ctrl+shift+t | 搜索:Ctrl+F |
| 缩小:Win+- | 网页切换: Ctrl+Tab | 复制:Ctrl+C |
| 表情包:Win+。 | 任务管理: Ctrl+alt+Eso | 剪切:Ctrl+X |
| 左分屏:Win+← | 微信自带截图: Alt+a | 粘贴:Ctrl+V |
| 右分屏:Win+→ | 微信隐藏: Ctrl+alt+w | 撤销:Ctrl+Z |
| 最小化:Win+↓ | 聊天换行: shift+Enter | 替换:Ctrl+H |
| 通知管理:Win+A | 锁定大写: CapsLock | 新建:Ctrl+N |
| 快速搜索:Win+Q | 屏蔽数字: NumLock | 保存:Ctrl+S |
| 快速指令:Win+R | 全角切换: shift+空格 | 关闭:Ctrl+W |
| 输入法:Win+空格 | shift+点任务栏窗口 | 输入法:Ctrl+shift |

电脑常用快捷键大全

Win+A 打开操作中心

Win+D 快速显示桌面

Win+E 打开资源管理器

Win+H 语音听写

Win+I 打开设置

Win+K 投屏到外部设备

Win+L 一键锁屏

Win+M 窗口最小化

Win+N 打开通知面板

Win+S 一键搜索

Win+R 运行

Win+Home 最小化非活动窗口

Win+空格 切换输入法

Win+TAB 切换虚拟桌面

Win+Ctrl+D 新建虚拟桌面

Win+Prtscn 一键截屏

Ctrl+W 关闭应用程序

Ctrl+空格 中英文切换

Ctrl+Esc 显示【开始】菜单

Ctrl+Shift+N 新建文件夹

Ctrl+Alt+End 快速关机

Ctrl+Alt+Home 强制重启

Alt+空格+C 关闭窗口

Alt+空格+N 最小化当前窗口

Alt+空格+R 恢复最小化窗口

Alt+空格+X 最大化当前窗口

Alt+空格+S 改变窗口大小

Alt+Tab 两个程序切换

Alt+Enter 查看文件属性

Shift+Del 永久删除所选项

附录三：日常.....

优秀学习资源和求助网站推荐



CSDN : 专业开发者社区，可以在里面查到大佬们分享的资源或者学习资料 (www.csdn.net)



Bilibili : 视频分享网站，可以找到许多优秀博主的资料分享

(<https://www.bilibili.com/>)

廖雪峰的官方网站：廖雪峰是一个全栈工程师，廖雪峰的官方网站为您提供原创精品中文教程。 (<https://liaoxuefeng.com/>)

菜鸟教程：菜鸟教程是一个提供各种编程语言和技术的在线教程的网站，涵盖了 HTML、CSS、JavaScript、Python、Java、C、SQL 等多个领域。 (<https://www.runoob.com/>)

推荐自学内容：

markdown 语法，终端的使用，git 的使用，科学上网

优秀软件推荐：

软件推荐：武装你的科研电脑

本清单强烈反对安装“xx 卫士”、“xx 管家”等软件。对于科研电脑而言，稳定、安静、高效是第一要务。！！！！以下推荐软件均为免费软件！！！！

一、效率神器类

Everything

是什么：一款基于名称的本地文件搜索工具。

为什么是神器：Windows 自带的搜索慢如蜗牛，而 Everything 输入即所得，秒出结果。当你忘记论文、代码或数据文件放在哪里时，它是你的救命稻草。

用法：安装后，按 Ctrl+Space（可自定义）唤出搜索框，输入文件名，结果实时呈现。

Geek Uninstaller

是什么：一个极致轻量级的软件卸载工具。

为什么好用：Windows 自带的卸载经常有残留。Geek 能彻底清除软件的所有文件和注册表项，保持系统清洁。绿色版，无需安装，打开即用。

7-Zip

是什么：一款完全免费、开源、无广告的压缩/解压缩软件。

为什么选它：兼容所有常见格式 (ZIP, RAR, TAR.GZ 等)，性能优秀。是替代收费的 WinRAR 和广告众多的国产压缩软件的最佳选择。

二、媒体与截屏类

系统自带截屏

Win + Shift + S：最强截图快捷键。可以矩形、窗口、全屏幕截图，并直接复制到剪贴板，方便粘贴到微信、QQ 或论文里。

PrtSc 键：全屏截图，可粘贴到画图等工具中。

OBS Studio

是什么：一款专业且免费的录屏与直播软件。

实验室用途：

录制操作教程：给你的代码运行过程、软件操作录屏，方便向他人演示或自己回顾。

线上会议分享：线上组会时，可以共享你的屏幕和操作。

虽然专业，但基础录屏功能上手很快。

三、安全与系统类

火绒安全软件

为什么推荐：如果你实在需要一款安全软件，只推荐火绒。它安静、轻量、无弹窗广告，主打“防病毒+防火墙+弹窗拦截”，完美符合实验室的纯净需求。

核心原则：**坚决杜绝 360**、金山毒霸、电脑管家等。这些软件本身就会拖慢系统、产生大量弹窗，是“麻烦制造者”。

最佳选择：对于能规范上网的同学，Windows 系统自带的 Windows Defender 已经完全足够，无需安装任何第三方杀软。

四、信息获取与 AI 类

Edge 或 Google Chrome 浏览器

为什么：现代、快速、兼容性最好的浏览器，是科研工作的标准配置。

重要提醒：安装后第一件事，是将默认搜索引擎设置为 Google。在浏览器设置中搜索“搜索引擎”，将其改为 Google。杜绝使用百度，在查找技术问题和中文学术资料时，百度结果的质和量都远逊于 Google。

Z-Library

是什么：全球最大的数字图书馆之一（请注意其访问地址可能经常变动）。

实验室用途：寻找教科书、学术专著、小说的电子版。是科研学习的强大助力。请善用此资源。

活用 AI 工具

ChatGPT / DeepSeek / Claude 等：

实验室正确用法：

解释复杂概念：让 AI 用简单的语言向你解释一个陌生的学术概念。

debug 代码：将报错信息丢给 AI，让它帮你分析可能的原因。

生成代码框架：让 AI 为你生成某个功能的代码框架或示例。

润色英文邮件/论文：辅助进行英文写作。

重要提醒：AI 是强大的助手，但不是权威。它可能会“一本正经地胡说八道”（幻觉）。所有它提供的信息，尤其是代码和学术内容，你必须批判性地验证和测试，绝不能直接照搬。

一些让你受益终身的电脑使用小提醒

这些小贴士无关高深技术，但每一条都凝聚了前人的“血泪教训”。遵循它们，你的数字生活会更顺畅。

关于系统和软件

程序尽量安装在 D 盘：安装软件时，不要无脑“下一步”。在选择安装路径的步骤，将默认的 C:\Program Files\... 改为 D:\Program Files\...。这能有效为 C 盘（系统盘）减负，防止系统因空间不足而变慢，也方便后续管理。

C 盘只装系统和必备软件：把你的 C 盘想象成电脑的“大脑”，只存放操作系统和最核心的软件（如 Office、VS Code）。游戏、大型软件（如 MATLAB）、娱乐软件等，请一律安装到 D 盘或其他盘。

定期重启一下电脑：电脑长时间不关机，内存中会堆积很多垃圾，导致系统变慢、出现

一些玄学问题。每周重启一次，能让它“焕然一新”。

减少开机启动项，过多的开机启动项会导致电脑开机缓慢.....

如何管理开机自动启动的程序

Ctrl + Shift + Esc 打开任务管理器 -> “启动”选项卡。禁用那些不需要一开机就运行的程序，能显著提升开机速度。

卸载软件利用系统自带或者专业软件卸载，不要简单的删除快捷方式！！！！！！！！

关于文件和数据

文件别堆在桌面：桌面文件默认占用 C 盘空间。养成好习惯，在 D 盘建立清晰的文件分类夹（如“学习”、“项目”、“个人”），并将文件妥善存放。

重要文件“备份三二一”：重要数据（论文、代码、实验数据）一定要遵循 3-2-1 备份原则：3 个副本，2 种不同介质（如电脑+移动硬盘），1 份异地备份（如实验室 NAS+宿舍电脑）。

给文件起个好名字：使用 20231026_项目名_描述_v2.pdf 这样的格式命名文件。它会自动按时间顺序排列，一目了然。

关于安全和习惯

离开座位，随手锁屏：使用 Win + L (Windows) 或 Super + L (Ubuntu) 快捷键。这是保护你隐私和数据安全最简单的习惯。

安装软件时，擦亮眼睛：从官网下载软件，安装时注意取消勾选那些捆绑的“全家桶”软件，它们往往是广告和病毒的来源。

系统别乱优化：除非你很清楚在做什么，否则不要使用来历不明的“系统优化/加速/清理”软件。它们往往是制造问题的元凶。Windows 系统自带的磁盘清理工具已经足够。

关于效率和体验

学会使用快捷键：哪怕只记住 Ctrl+C/V（复制/粘贴）、Ctrl+S（保存）、Win+E（打开文件管理器）、Win+D（显示桌面）这几个，你的效率都会提升一大截。

网线比 Wi-Fi 更可靠：在实验室固定工位上，如果条件允许，尽量使用网线上网。它比 Wi-Fi 更稳定、延迟更低，传输大文件时优势明显。

外接显示器是效率神器：如果你的工位支持，强烈建议申请一台外接显示器。双屏可以一个写代码，一个查资料/看效果，效率直接翻倍。

最后，也是最重要的一条：

保持好奇，大胆尝试：电脑没那么容易坏。在合理的范围内（比如不是去修改系统核心文件），大胆地去探索、去尝试。你遇到的绝大多数问题，都可以通过重启或搜索找到解决方案。真正的知识，源于解决一个个具体的问题。

希望这些小提醒能像一位耐心的师兄师姐一样，在细节处为你指点迷津，助你养成良好的电脑使用习惯！

后续补充内存

MacOS 的使用 各个技术组独特的计算机基础内容 ai 的使用.....

致谢

本指南的诞生，源于实验室视觉组同学们最真实的需求。它并非由一人完成，而是凝聚了无数个在实验室里共同调试代码、解决报错、分享技巧的瞬间。

感谢每一位在培训中提出“小白”问题的同学，你们的好奇心是完善这份指南最原始的动力。

感谢所有无私分享经验的师兄师姐，你们踩过的“坑”和总结的“捷径”，是本书最宝贵的财富。

特别感谢实验室提供的开放与协作的氛围，让我们能够自由地探索、试错并共同成长。

最后，感谢正在阅读的你。你的时间和关注，是对我们努力最大的肯定。愿你将从这里学到的知识、养成的好习惯，用于你未来的学习和研究中，并乐于将它们传递给下一届同学。

本指南将作为一个“开源项目”，持续迭代，不断完善。欢迎你的批评、指正与贡献。

—— 你们的实验室战友 谨致

本文档由风暴战队视觉组集体创作与维护

如果你发现错误或想补充内容，请通过以下联系方式联系沟通，或者在 GitHub 上提 Pull Request。

邮箱:xuan4938@outlook.com

qq:2557898868

微信:I_will_get_su7_ultra

版权声明

文档性质：本文档《理工科计算机基础指北》是一个开源、非商业性的教育项目，旨在分享知识，方便学习者。

内容版权：

本文档中使用的部分文字由人工智能生成，其版权状态遵循模型服务商的默认条款。

部分图片来源于公开网络，我们尽力标注了来源。这些图片的版权属于其原始创作者，我们在此按“合理使用”原则引用，仅用于教育和说明目的。

非商业性：我们严厉禁止任何个人或组织将本文档及其中包含的图片用于任何形式的商业盈利活动。

侵权处理：如果您是本文档中某张图片的版权所有者，并认为我们的使用侵犯了您的权益，请通过邮箱 [xuan4938@outlook.com] 与我们联系。请提供具体的图片信息和您的权属证明，我们将在核实后立即进行删除或替换处理。

我们衷心感谢所有图片的作者和开源社区的贡献者。