1. 操作系统是什么

1.1 导入

日常使用计算机时，你是否思考过它的工作原理？就拿屏幕显示信息来说，这得显卡和屏幕默契配合才行。要是想看 VCD 影片，所需硬件就更繁杂了，要有存影音数据的光盘、能读盘的光驱、处理数据的 CPU、负责影像显示的显卡、传输声音的声卡，以及输出影像的屏幕与发声的喇叭。显然，执行各项任务靠的都是硬件。

不过，硬件可不会 “自主思考” 怎么播放 VCD，背后必然有个掌控全局的 “总指挥”，这就是操作系统。操作系统牢牢把控整部计算机的硬件，指挥 CPU 精准运算，精准识别并读取硬盘数据，还能与各类适配卡 “无缝对接”，有它坐镇，硬件们才能有条不紊协同工作，缺了操作系统，计算机就是一堆毫无用处的金属。

仅有操作系统管理硬件还不够，要是用户无法和操作系统交流，那操作系统也就形同虚设。好比看 VCD，即便系统能控制硬件播放，用户却没办法下达播放指令，那也只能干瞪眼。

所以，一个完备的操作系统包含两大关键部分：一是 “核心及其提供的接口工具”，二是 “利用核心接口工具开发出来的软件” 。以常见的 Windows 系统为例，打开 “资源管理器” 能看到硬盘数据，靠的是核心功能；而指定显示特定目录的数据，则要仰仗 “资源管理器” 这个工具。

核心也有力不能及的时候。比如给计算机装新显卡时，可能就会碰到 Windows 提示 “找不到合适的驱动程序来显示”，这意味着即便有新显卡和播放程序，核心不认它，硬件就没法工作。

简单讲，能让计算机硬件稳定运行的就是操作系统，如今说的操作系统通常包含核心与应用软件，因为基础核心缺个方便用户沟通的界面。核心英文叫 “Kernel”，处于操作系统底层，掌管硬件资源运作状态。每个操作系统都有专属核心，新硬件出现，要是 “Kernel” 不支持，那就没法用。

那 Kernel 负责哪些关键事务呢？

- **系统呼叫接口**：给程序开发者搭起便利之桥，借助它，开发者能轻松和 kernel 沟通，把硬件资源利用得更高效。
- **行程管理**：大家对 “多任务环境” 不陌生吧，同一时间计算机可能有一堆任务等 CPU 处理，Kernel 就得巧妙安排，实现 CPU 资源高效分配。
- **内存管理**：负责管控整个系统内存，内存不够时，还得提供虚拟内存功能。
- **档案系统管理**：涵盖数据输入输出、不同档案格式支持等工作，要是核心不认识某种档案系统，对应的档案格式就没法用，比如 Windows 98 不认识 NTFS 格式硬盘。
- **装置的驱动**：硬件管理是 Kernel 核心职责，驱动程序适配就归它。好在如今有 “可加载模块” 功能，驱动编成模块，不用每次都重新编译核心，后续介绍核心编译会再提。

总之，硬件资源都由 kernel 统一管理，人们完成任务，除用核心自带功能，像文件总管，还能靠其他应用软件。比如看 VCD，除 Windows 自带播放程序，自己装的也行，这类播放程序就是应用软件，能指挥核心播放影片。可见，核心把控硬件，是操作系统底层基石，搭配丰富工具和应用软件，操作系统才更完善。

1.2 Linux 简介

了解完操作系统，再看 Linux。它本质上是一个操作系统，自带核心与核心提供的工具，共同构建起控制硬件、管理资源的底层架构。这架构传承自 Unix 的优良基因，既稳定又强大，还能在常见的个人计算机（X86 系统）上运行，引得众多软件开发者将成果迁移过来，应用软件越来越丰富，与核心、工具整合后，Linux 发展成功能完备强大的操作系统。

1.3 Linux 稳定的原因

Linux 的稳定性和它的发展历程息息相关，在它诞生前，已有成熟稳定的 Unix 系统，二者渊源深厚，下面一起梳理它们的发展脉络。

1.3.1 Unix 的发展历程

- **早期困境（1969 年以前）**：早期计算机堪称 “奢侈品”，只有军事、高科技、学术等特定领域才用得起，不仅贵，用起来还麻烦，运算慢、操作接口不友好。编写程序更是折磨，要在读卡纸上打孔，再插入读卡器录入主机运算，程序稍有差池，重新打孔就得费好大劲，而且主机少、使用者多，光排队就得等好久。后来操作系统有改进，支持键盘输入输出，但仍有不足。像麻省理工学院的 “兼容分时系统 (CTSS)”，虽说能让主机连终端机供用户用主机资源运算，可终端机只有输入输出功能，没法运算和安装软件，一台主机最多连 30 个左右终端机。为强化主机系统，让更多用户受益，1965 年前后，贝尔实验室、麻省理工学院及奇异公司联合发起 Multics 计划，目标是让主机连 300 个以上终端机，可惜到 1969 年，因进度滞后、资金短缺，计划宣告失败。
- **Unix 雏形出现（1969 年）**：Multics 计划泡汤后，贝尔研究室的 Ken Thompson 却从中得到灵感，打算给自己搞个小操作系统。刚好有台闲置的 DEC 的 PDP - 7 主机，他就以此为基础写核心程序。1969 年 8 月，妻儿去美西探亲，他多出一个月时间，经过四周奋斗，用组译语言写出核心程序、一些工具程序与小型档案系统，这就是 Unix 雏形，当时还被同事打趣叫 Unics。它有两个重要理念：一是把所有程序或系统装置当作档案；二是编写程序要目标明确、高效完成任务，这对后来 Linux 发展影响深远。
- **Unix 正式诞生（1973 年）**：Thompson 写的操作系统在贝尔实验室内部流传，历经多次改版。1973 年那次改版意义重大，起初 Unix 用组译语言写，因系统移植与效能需求，先用 B 语言改写，效能不理想，后来 Dennis Ritchie 把 B 语言改成 C 语言，并用 C 语言重写操作系统，发行了正式版本。当时贝尔实验室属于 AT&T ，AT&T 忙于其他商业事务，对 Unix 态度开放，参与开发的又是自家技术过硬的工程师，这使得 Unix 不太容易被大众接受。不过用 C 语言写有好处，它和硬件关联性弱，Unix 更容易移植到不同机器。
- **BSD 分支诞生（1977 年）**：虽说贝尔实验室归 AT&T 管，但 AT&T 对 Unix 开放，加上 Unix 用 C 语言写有移植性，只要拿到源码，按不同主机特性改改，就能移植。加州柏克莱大学的 Bill Joy 拿到 Unix 核心源码后，改成适合自己机器的版本，加了很多工具软件与编译程序，命名为 Berkeley Software Distribution (BSD) ，这成了 Unix 重要分支，Bill Joy 还创立 Sun 公司，用 BSD 发展商业 Unix 版本，后来的 FreeBSD 就是 BSD 改版来的。
- **版权风云（1979 年）**：Unix 因可移植性强、效能高，又没版权纠纷，吸引众多商业公司开发，像 AT&T 自家的 System V、IBM 的 AIX 、HP 和 DEC 等公司，都给自家主机配自家 Unix 。但早期硬件公司没 “协议” 意识，各公司硬件差异大，对应的 Unix 系统只能支持自家硬件，没法在其他硬件运行，也没人给个人计算机设计 Unix ，因为早期 x86 个人计算机 CPU 没多任务能力。所以早期 Unix 基本和服务器、大型工作站绑定。1979 年，AT&T 出于商业考虑收回 Unix 版权，在第七版 Unix 中规定不给学生提供源码，瞬间引发商业紧张氛围和纠纷。
- **Minix 问世（1984 年）**：1979 年版权声明可把教 Unix 的教授愁坏了，没源码咋教学生？安德鲁・谭宁邦（Andrew Tanenbaum）教授另辟蹊径，鉴于 Unix 第七版能在 Intel x86 架构移植，他决定改写移植到 x86 上，于是有了 Minix 这个类似 Unix 的核心程序。写的时候为避版权纠纷，他不看 Unix 核心源码，还强调要和 Unix 兼容。从 1984 年开始写，1986 年写完，次年出版相关书籍，还和新闻群组结合推广。不过 Minix 不是免费的，得买磁盘或磁带，好在便宜，还附源码，方便学习核心程序设计概念，对 Linux 早期开发很关键。只是开发者只有谭宁邦教授一人，他觉得 Minix 用于教育，后续发展受限，难满足用户新需求。
- **GNU 与 FSF 计划启动（1984 年）**：1984 年，Richard Mathew Stallman（史托曼）发起的 GNU 计划，对如今自由软件发展意义非凡。史托曼从小聪明，1971 年进知名人工智能实验室，那时黑客爱分享软件，没专利困扰，这影响他很深。后来因管理问题，实验室不少优秀黑客去商业公司，他不服气，留在实验室继续开发。再后来，实验室换硬件，他原来用的 Lisp 操作系统是麻省理工学院专利软件，不能共享，他就改用 Unix 。但之前写的软件在 Unix 上跑不了，他就把软件移植到

Unix，还写成跨平台形式。为开发软件，他开始写著名的 GNU C (gcc) 编译器。起初不顺利，他先把 Emacs 编辑器改成能在 Unix 上用的，公开源码，很多人买，他赚了点钱，接着全力写其他软件，还成立自由软件基金会 (FSF)，找更多人一起写。到 1990 年左右，GCC 写完了，还搞出了 GNU C 库、BASH shell 等。1985 年，为防自由软件被商业化利用，他和律师搞出通用公共许可证 (GPL)，叫 copyleft，与专利软件的 copyright 对着来。有了 GNU 这些重要软件，像 Emacs、GCC、GNU C 库、Bash shell，很多开发者能借此开发程序，壮大了自由软件团体。不过，GNU 最初想建自由的 Unix 操作系统，光有这些软件不够，还缺自由的 Unix 核心，直到 Linux 出现才补上。

- **图形接口计划（1988 年）**：随着用户对图形使用者接口 (GUI) 需求渐涨，1984 年 MIT 等厂商发表了 X Window System，1988 年成立非营利的 XFree86 组织，名字是 X Window System + Free + x86 的整合。1994 年 Linux 1.0 版发布时，把 XFree86 的 GUI 界面整合进去了。注意，是 X Window，不是 X Windows 。
- **Linux 初露锋芒（1991 年）**：1991 年，芬兰赫尔辛基大学的 Linus Torvalds 在 BBS 上发消息，说用 bash、gcc 等工具写了个小核心程序，能在 Intel 386 机器上跑，一下子吸引好多人，Linux 就此开启不凡之旅。

1.3.2GNU 计划解读

GNU 计划秉持开放源代码理念，史托曼觉得，写程序最大快乐是分享优质软件，不同用户软硬件环境不同，只有公开源码，大家才能修改适配自己电脑，这就是 Open Source 。他还坚信，公开源码后，软件好就会有很多人用，大家能帮忙查错，软件会越来越强。为防自由软件被搞成专利软件，他给 GNU 与 FSF 开发的软件都挂 GPL 版权声明。

GPL 强调的 “自由” 软件，不是免费，而是给使用者这些自由：自由执行、复制、再发行、学习、修改、强化软件。软件挂 GPL 声明就成自由软件，有这些特性：

- **获取与使用**：能按需执行软件，获取源码。
- **复制权限**：可自由复制。
- **修改权利**：能改源码满足工作需求。
- **再发行自由**：修改后的程序能再发行，不冲突。
- **回馈义务**：建议把修改代码回馈社群。

但也有限制：

- **授权变更限制**：不能把 GPL 授权软件改后取消 GPL 授权。
- **单纯贩卖禁止**：不能光卖自由软件，要搭配售后服务、手册，收工本费。

很多人奇怪为啥 Linux 开发商能卖 Linux，因为他们大多靠卖 “售后服务” 盈利，软件在官网能下，买光盘会有手册、咨询、售后、升级等服务。对开发者来说，GPL 授权软件安全性好、效能高、除错快，自己贡献代码能留名；对普通大众，用起来可能难，得会基本编译器操作。

1.3.3 Torvalds 的 Linux 发展脉络

- **早期接触与 Minix 关联**：Linus Torvalds（托瓦兹）的外祖父是赫尔辛基大学统计学家，小时候就带他接触微计算机，学了汇编语言。1988 年托瓦兹进赫尔辛基大学计算机科学系，学业和兴趣让他接触到 Unix，但学校只有一台 Unix 系统，16 个终端机，用起来太费劲，他就想自己搞个 Unix 。后来知道 Minix 系统，和 Unix 兼容还能在 Intel 386 跑，他就买了 386 电脑装 Minix，还从 Minix 源码学到好多核心程序设计概念。
- **386 硬件测试与探索**：托瓦兹之前用工作站型计算机，CPU 能多任务处理，对早期 x86 架构计算机多任务能力不满意。386 计算机推出后有改善，从性价比看合适，他就贷款买了一台。托瓦兹擅长汇编语言，深知与硬件关系密切，为发挥 386 效能，花时间测试 386 多任务功能。他写两个小程序，一个持续输出 A，一个输出 B，同时运行，看到屏幕顺利出现 ABABAB..... ，知道测试成

功。要实现多任务，硬件和操作系统都得支持，不支持多任务的系统，后面程序得等前面的执行完，多任务系统里每个程序有最大 CPU 使用时间，超了就得等下次调度。

- **Linux 首次发布**：Minix 发展受限，侧重教育，对一些用户功能需求响应慢，托瓦兹决定写个更适合自己的。他用 GNU 的 bash、gcc 等自由软件，参考 Minix 设计理念（没抄源码），写出能在 386 机器上流畅运行的 Linux 内核雏形，并将其公布在网络上。这一小小的内核，立刻在技术爱好者群体里引发强烈反响，大家被它蕴含的潜力所吸引，纷纷加入后续的开发与完善工作当中。
- 起初，这个内核功能还十分基础，但得益于开源模式，全球各地的开发者依据自身专长与实际需求，不断为它增添新特性、修复漏洞。代码在众人接力之下日益丰富，稳定性与兼容性也逐步提升。比如说，有的开发者专注优化内核的内存管理模块，让系统能够更高效利用有限的内存资源，应对多任务场景；有的则投身于驱动程序开发，促使 Linux 能够适配更多新型硬件设备。
- 在社区协作的热烈氛围里，Linux 的生态开始蓬勃生长。除了内核持续迭代，围绕它的各类应用软件、工具软件也如雨后春笋般冒了出来。像文本编辑器、网络浏览器、多媒体播放器等日常软件纷纷有了 Linux 版本，极大扩充了 Linux 系统的实用性，让它不再仅仅是极客手中的玩物，开始走入更广泛的大众视野与应用场景，无论是个人桌面使用，还是小型办公环境，都逐渐有了 Linux 的身影。
- 众多 Linux 发行版也陆续登上舞台，基于 Linux 内核，不同团队或组织整合进各具特色的软件包集合、桌面环境与系统配置方案，从而分化出诸如 Ubuntu 这般对新手极为友好、上手轻松的发行版，它有着简洁美观的桌面，丰富的软件仓库，还自带详细的新手引导；还有 CentOS 这类专为服务器场景优化的版本，高度重视稳定性与安全性，为企业级数据中心、网站托管等业务筑牢根基；以及 Debian，以其严谨的软件包管理体系、对开源精神的纯粹贯彻，收获大量追求原汁原味开源体验用户的拥趸 。这些发行版各展所长，把 Linux 生态的版图越铺越广，进一步巩固了 Linux 在操作系统领域的重要地位，让它从一颗破土而出的幼苗，茁壮成长为枝繁叶茂的参天大树，为全球数字世界源源不断输送着创新活力与可靠方案。

2. 认识 Linux

2.1 引入

2.1.1 为什么要学习 Linux

在当今的数字世界中，学习 Linux 有着诸多重要意义：

- **广泛应用领域**：Linux 在服务器领域可是当仁不让的 “霸主”，全球知名搜索引擎、社交媒体平台等大型网站与企业级应用，背后的服务器大多采用 Linux 系统。这得益于它超强的稳定性，能长时间不间断运行；出色的安全性，抵御外界攻击的能力一流；还有高度的可扩展性，满足业务增长需求。
- **开源优势**：Linux 是开源操作系统，源代码完全公开。这就像打开了知识宝库的大门，任何人都能自由进出查看、按需求修改，还能分享给他人。如此开放性，吸引了全球海量开发者与用户汇聚成庞大社区，大家相互交流切磋。学习 Linux，就等同于拿到一把深入了解操作系统内部机制的钥匙，对提升个人技术水平、锤炼解决问题能力助力极大。
- **跨平台与兼容性**：Linux 简直是 “万能适配王”，不管是个人电脑、超级计算机，还是嵌入式设备、服务器，各种硬件平台它都能轻松驾驭。并且众多开发工具和软件都对 Linux 系统青睐有加，掌握 Linux，就等于在软件开发、系统管理等领域解锁了更多可能，拥有更多选择与便利。

2.1.2 Linux 与其他操作系统的比较

- 与 Windows 对比：
    - **用户界面**：Windows 有着对普通用户极为友好的图形用户界面（GUI），新手也能快速上手；Linux 的 GUI 则因发行版而异，部分发行版界面美观又易用，不过也有些更侧重命令行操作。

    - **软件生态**：Windows 坐拥庞大商业软件生态，热门办公软件、游戏大多首选在此发布；Linux 软件主要依靠软件包管理器安装，开源软件资源丰富，但商业软件支持稍显薄弱。好在技术发展日新月异，越来越多软件开始拥抱 Linux 平台。
    - **安全性**：Linux 凭借开源特性，漏洞一旦出现，很容易被发现并修复，系统架构与权限管理机制也更为安全；Windows 用户基数庞大，常沦为黑客攻击的首要目标，即便微软不断强化安保措施，在安全性上 Linux 仍略胜一筹。
- 与 macOS 对比：
    - **硬件兼容性**：macOS 基本 “绑定” 苹果自家硬件，硬件兼容性较窄；Linux 不挑硬件，能在各式各样的硬件平台畅行无阻，用户可按需搭配硬件。
    - **开发环境**：二者都受开发者喜爱，Linux 在服务器端开发、嵌入式开发领域应用更广，还配备了更丰富的开发工具与灵活的环境配置选项；macOS 则在移动应用开发，尤其是 iOS 开发方面独具优势。
    - **成本**：使用 macOS 往往要购置苹果硬件，成本高昂；Linux 免费就能安装使用，无论是个人用户还是企业，都能在成本上松一口气。

3. 重点回顾

1. 计算机主要以二进制为运算基础，常用磁盘容量单位是字节（bytes），其换算关系为 1 字节（Byte） = 8 比特（bits），其他单位皆以 1024 为倍数进阶，例如 1 吉字节（GByte） = 1024 兆字节（MBytes） 。
2. 操作系统的核心职责在于管理并驱动硬件，这要求它能够进行内存管理、设备管理、进程管理以及处理系统调用等操作。从本质来讲，只要能将硬件调配至就绪（Ready）状态，便可算作一个基础的操作系统。最简易的操作系统仅负责驱动与管控硬件，当用户要使用硬件时，需借助应用软件或者 shell 程序，向操作系统发出指令，驱使硬件工作。如今提及的操作系统，除上述基本功能外，往往还囊括日常办公所需的各类应用软件。
3. Unix 的起源可以追溯到贝尔实验室（Bell lab.），最初由 Ken Thompson 运用组译语言编写，后在 1971 - 1973 年期间，Dennis Ritchie 使用 C 语言对其改写，这才正式命名为 Unix 。1977 年，Bill Joy 发布了 BSD（Berkeley Software Distribution），这类系统被归为类 Unix （Unix like）操作系统。
4. 1984 年，Andrew Tannenbaum 打造出 Minix 操作系统，该系统不仅提供源码，还附带相关软件；同年，Richard Stallman 发起 GNU 计划，大力推广自由软件理念，主张软件能够 “自由获取、复制、修改与再次发行”，同时确立了 GPL（General Public License）授权模式。依据此模式，任何 GPL 授权软件都禁止单纯售卖软件本身，也不允许私自更改软件授权。
5. 1991 年，芬兰人 Linus Torvalds 开发出 Linux 操作系统。概括而言，Linux 的成功得益于 Minix（类 Unix 系统）、GNU 计划、互联网、POSIX 标准以及虚拟开发团队这些要素。Linux 本质上是较为基础的操作系统，其官方开发网站是 http://www.kernel.org ，它最底层的数据部分被称作 “核心（Kernel）” 。
6. 当前，Linux 发展出两种版本分支：稳定版本多为偶数版本号，例如 2.6.X 系列，适用于商业及家庭使用场景；另一种是开发版本，如 2.5.X 系列，适合用于开发特殊功能的场景。Linux 发行版（Linux distributions）则是整合 Linux 内核（Kernel）、工具（Tools）、自由软件（Free Software）、文档（Documentations）以及可完整安装程序后，所形成的一套完备操作系统。

4. 课后习题

一、单选题（10 道）

1. 1GByte 等于多少 MBytes?
   A. 1000
   B. 1024
   C. 512
   D. 2048
2. 操作系统要管理硬件，以下哪项不属于其基本管理范畴?
   A. 管理网络带宽
   B. 管理内存
   C. 管理装置
   D. 负责行程管理
3. Unix 最初是由谁利用组译语言写成的?
   A. Dennis Ritchie
   B. Bill Joy
   C. Ken Thompson
   D. Andrew Tannenbaum
4. 以下哪个操作系统是在 1984 年制作出来的?
   A. Linux
   B. BSD
   C. Minix
   D. Windows
5. GNU 计划由谁提倡发起?
   A. Linus Torvalds
   B. Richard Stallman
   C. Bill Joy
   D. Ken Thompson
6. Linux 操作系统最底层的数据被称作什么?
   A. Shell
   B. Kernel
   C. BIOS
   D. Driver
7. 目前 Linux 稳定版本的版本号特征通常是什么?
   A. 奇数版
   B. 偶数版
   C. 小数版
   D. 无规律
8. 1Byte 等于多少 bits?
   A. 2
   B. 4
   C. 8
   D. 16
9. 以下哪个属于类 Unix 操作系统?
   A. Minix
   B. Windows
   C. macOS（早期版本）
   D. BSD
10. GPL 授权模式规定，软件不可进行以下哪种操作?
    A. 自由修改
    B. 单纯仅贩卖其软件

C. 自由复制
D. 自由再发行

二、多选题（5 道）

1. 操作系统的主要功能包括以下哪些？
   A. 管理内存
   B. 驱动硬件
   C. 制作办公文档
   D. 负责行程管理
2. 以下哪些事件对 Linux 的成功有推动作用？
   A. Minix 的存在
   B. GNU 计划
   C. Internet 的发展
   D. POSIX 标准
3. Linux 发行版通常包含以下哪些部分？
   A. Kernel
   B. Tools
   C. Free Software
   D. Documentations
4. 以下哪些是在 1970 年代对操作系统发展有重要意义的事件？
   A. Unix 在 1971 - 1973 年间由 C 语言改写
   B. 1977 年 BSD 的释出
   C. 1979 年 AT&T 收回 Unix 版权
   D. 1970 年计算机首次应用于商业办公
5. 自由软件在 GNU 计划倡导下，拥有以下哪些自由？
   A. 自由取得
   B. 自由复制
   C. 自由修改
   D. 自由再发行

三、判断题（5 道）

1. 最早的操作系统可以直接供用户日常办公使用，无需其他软件。
2. Linux 是由芬兰人 Linus Torvalds 在 1991 年独立完成开发的，没有借助任何前人成果。
3. BSD 是由微软公司开发的一款操作系统。
4. 计算机磁盘容量单位换算和普通的十进制换算一样，都是以 1000 为倍数。
5. GNU 计划中的 GPL 授权软件可以随意修改软件授权。