Linux图形界面栈

- 如何开始理解？
  - Linux的图形栈工作流程就像是盖房子
  - 工作流程
    - 第一层：该层为地基（驱动和内核），提供了最基础的能力
    - 第二层：该层是施工队入口（显示管理器），决定哪个工程队可以进来干活
    - 第三层：该层是核心建筑框架和法规（X11或Wayland），决定了这栋楼怎么盖（协议），并用钢筋混泥土搭出结构（对于Wayland是合成器的工作，对于X11是WM的工作）
      - 该层为关键决策点
      - 选择 X11 路径，你会进入一个模块化的世界，Window Manager (WM) 是一个可以自由替换的独立组件
      - 选择 Wayland 路径，你会发现 Compositor 是一个集成的、不可分割的整体，它决定了你后续能搭配的“装修风格”（Desktop Environment，桌面环境）
    - 第四层：该层是精装修和家具（DE或最小化组合），用户最终看到和使用的样子
- 【第四层】：用户体验层 (User Experience Layer)
  - 功能：用户直接交互的、可见的应用程序和界面的集合
  - 组件
    - **桌面环境 (Desktop Environment - DE)**
      一个预配置好的、功能完整的软件套件，包含所有【第三层】的组件并提供统一的体验
      - DDE
      - GNOME
      - KDE Plasma
      - XFCE
      - Cinnamon
      - LXQt
      - Budgie
      - .....
    - **最小化组合**
      用户自行组合【第三层】的组件，通常是一个独立的窗口管理器加上各种轻量级工具
      - 如：i3 + polybar + rofi + Thunar...
  - 关键点：桌面环境（DE）是一个概念，而不是一个技术层。它建立在下面的技术层之上。
- 【第三层】：窗口与合成层 (Window & Compositing Layer)
  - 功能：管理屏幕上的窗口（绘制、移动、缩放、叠加）、组合最终图像并输出给下一层。这是用户能感知到的“图形界面”核心
  - 技术/协议
    - **X11**
      说明：一个独立的协议和服务器
      - **X Server**
        说明：负责底层的渲染和输入
      - **窗口管理器 (Window Manager - WM)**
        说明：运行在X Server之上，负责窗口的装饰和行为。这是一个独立且可替换的组件
        - 堆叠式(Stacking): Openbox, Fluxbox
        - 平铺式(Tiling): i3, AwesomeWM, Qtile
    - **Wayland**
      说明：一个协议规范；这个层的功能由一个组件实现
      - **合成器 (Compositor)**
        说明：一个集成了显示服务器、窗口管理器和合成功能的单一组件。Wayland Compositor就是Wayland世界的“WM+X Server”
        - Mutter (GNOME)
        - KWin (KDE Plasma)
        - Sway
        - Hyprland
        - River
- 【第二层】：显示与通信层 (Display & Communication Layer)
  - 功能：充当【第三层】和【第一层】之间的桥梁。负责启动会话、认证、选择不同的图形环境（X11或Wayland）
  - 组件
    - **显示管理器 (Display Manager - DM)**
      提供图形化登录界面，用户选择会话后，由DM来启动对应的Layer 3组件（如plasmax11或gnome-wayland）
    - **从TTY启动（替代方案）**
      用户可以不使用DM，直接从命令行TTY手动执行startx或sway命令来启动图形界面
- 【第一层】：核心图形驱动层 (Core Graphics Layer)
  - 功能：与Linux内核和硬件直接交互，提供最基础的图形渲染能力。
  - 组件
    - **Linux Kernel**
      提供DRM (Direct Rendering Manager)、KMS (Kernel Mode Setting) 等子系统来直接管理显卡和显示输出
    - **图形驱动**
      Mesa (开源驱动，如radeonsi, anv, iris), NVIDIA proprietary, AMDGPU-PRO
  - **抽象库**
    SDL, OpenGL, Vulkan 等应用程序使用的图形API，它们通过驱动与硬件通信