



信息工程系

教 案

课程名称： 嵌入式应用技术

教 师： 陈凯斌

总 学 时： 36

理论学时： 0

实训学时： 36

上课班级： 电子信息工程技术 24 级

授课学期： 2025-2026 第 2 学期

第 1 章：智能终端基础开发环境搭建(4 课时)

教学目标

1、知识目标：

了解嵌入式智能终端的基本组成和开发流程。

掌握智能终端基础开发环境的搭建方法。

理解程序编译与移植的基本概念。

2、能力目标：

能够独立搭建一个基本的嵌入式开发环境。

能够编写并运行简单的“Hello World”程序。

3、素质目标：

培养学生对嵌入式开发的兴趣和动手能力。

引导学生思考开发环境搭建过程中可能遇到的问题及解决方案。

4、课程思政目标：

培养自主创新能力：通过搭建开发环境，引导学生认识到掌握核心技术的重要性，激发自主学习和创新的动力，树立科技自立自强的信念。

增强团队协作意识：在环境搭建过程中，鼓励学生分工合作，培养团队协作精神和集体荣誉感，体会团队合作的力量。

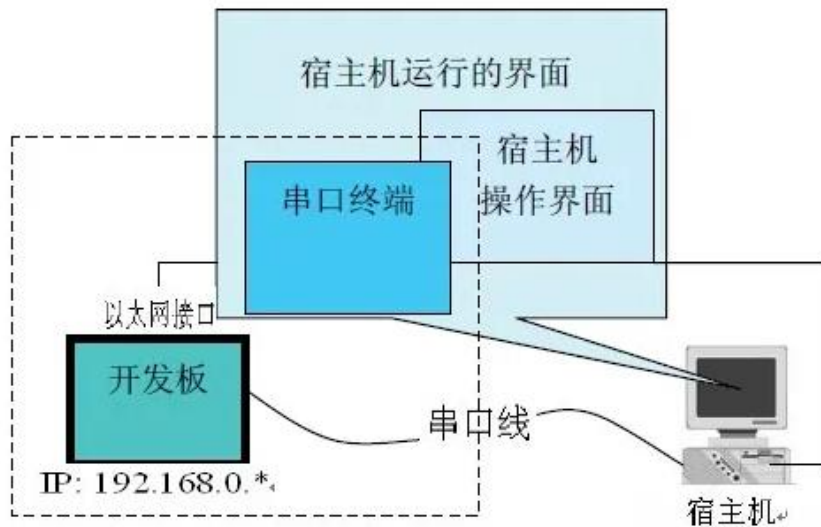
树立严谨治学态度：强调开发环境搭建的规范性和稳定性，培养学生严谨细致、追求卓越的科学态度。

重难点内容

1、嵌入式智能终端的开发要素：

嵌入式系统的基本组成（硬件、软件、操作系统）。

开发流程概述（需求分析、设计、编码、调试、测试）。



2、开发环境的搭建：

硬件选择（开发板等）。

操作系统、虚拟机等安装（如 virtualBox，Linux，wsl，超级终端，xshell 等）。

开发工具的安装与配置（如 GCC，Vim，arm-linux-gcc 等）。

3、程序编译与移植：

编译器的选择（GCC、arm-linux-gcc 等）。

编译过程（预处理、编译、汇编、链接）。

程序移植技巧（跨平台开发、兼容性问题）。

教学方法与过程

讲授法：通过 PPT 讲解嵌入式系统的开发流程和开发环境的搭建步骤。

演示法：现场演示开发环境的搭建过程，包括硬件连接和软件安装。

实践法：学生在实验课上亲自搭建开发环境，编写并运行“Hello World”程序。

1、导入：通过展示一些常见的嵌入式智能终端产品（如智能家居设备、智能手表等），引出本章的主题。提问学生对嵌入式系统的初步了解，激发学习兴趣。

2、新课讲解：使用 PPT 讲解嵌入式智能终端的开发要素和开发流程。详细介绍开发环境的搭建步骤，包括硬件选择和操作系统安装。讲解程序编译与移植的基本概念。

3、演示：在计算机上现场演示开发环境的搭建过程，包括安装 Linux 操作系统、配置开发工具等。展示如何编写一个简单的“Hello World”程序，并进行编译和运行。

4、实践指导：

学生在实验室中分组进行开发环境的搭建。

教师巡视，解答学生在搭建过程中遇到的问题。

指导学生编写“Hello World”程序，并进行编译和运行。

5、总结：

总结本章的重点内容，包括开发环境搭建的关键步骤和程序编译的基本流程。
强调在搭建过程中可能遇到的常见问题及解决方法。

6、布置作业：

搭建基本的嵌入式开发环境，并记录搭建过程中遇到的问题和解决方案。
编写简单的“Hello World”程序，并在所搭建的环境中编译运行。

- 1、完成VirtualBox和linux的安装
- 2、熟悉相关的图形界面和工具，比如如何调出终端，如进行文件的上传和下载
- 3、查看相关工具是否有安装，比如vim, gcc等
- 4、自行复习下相关的linux命令，如cd, ls等

使用C语言完成下面的任务，对每个任务使用GCC工具编译出对应的可执行文件，在虚拟机内运行

- 1、输出你的名字的拼音字符串，比如：小明同学的程序要输出：xiaoming
- 2、使用循环结构计算1到100的累加和，并直接打印结果
- 3、判断2024年是否为闰年，并打印结果
- 4、使用嵌套循环打印九九乘法表

教材：《物联网智能终端应用程序开发》第 2 章 2.1 节。

第 2 章：项目开发的常用命令与工具使用(4 课时)

教学目标

1、知识目标：

熟悉 Linux 系统的基本操作和常用命令。
掌握项目开发中常用的工具和软件。

2、能力目标：

能够熟练使用 Linux 命令进行文件操作、权限管理和进程控制。
能够使用文本编辑器、编译器和调试工具进行简单的程序开发。

3、素质目标：

培养学生的自主学习能力和团队协作精神。

4、课程思政目标：

培养工匠精神：通过熟练掌握常用命令和工具，引导学生树立精益求精的工匠精神，注重细节和质量，追求卓越。

增强自主学习能力：鼓励学生主动探索和学习新的命令和工具，培养自主学习和终身学习的意识，适应快速发展的技术环境。

树立网络安全意识：在使用工具和命令时，引导学生关注网络安全和信息安全问题，增强网络安全意识和责任感。

重难点内容

1、Linux 系统的基本目录与作用：

文件系统的结构（根目录、用户目录、系统目录等）。

常用目录的功能（如/bin、/etc、/home 等）。

2、Linux 系统的常用 shell 命令：

文件操作命令（ls、cp、mv、rm 等）。

权限管理命令（chmod、chown 等）。

进程控制命令（ps、top、kill 等）。

3、Linux 系统的常用工具：

文本编辑器（vi、nano 等）。

编译器（gcc、make 等）。



教学方法与过程

讲授法：通过 PPT 讲解 Linux 系统的基本操作和常用命令。

演示法：现场演示常用命令和工具的使用。

实践法：学生在实验课上进行命令操作练习。

1、导入：提问学生对 Linux 系统的了解程度，引入本章主题。展示 Linux 系统在嵌入式开发中的重要性。

2、新课讲解：使用 PPT 讲解 Linux 文件系统的结构和常用目录的功能。讲解常用 shell 命令的分类和使用方法。介绍开发中常用的工具及其功能。

3、演示：现场演示常用命令的操作，如文件操作、权限管理和进程控制。展示如何使用文本编辑器编写代码，如何使用编译器编译代码。

4、实践指导：学生在实验室中分组进行命令操作练习。指导学生使用文本编辑器编写一个简单的 C 程序，并使用编译器和调试工具进行编译和调试。教师巡视，解答学生在操作过程中遇到的问题。

5、总结：总结本章的重点内容，包括常用命令和工具的使用方法。强调在实际开发中如何高效使用这些工具。

6、布置作业：完成一系列 Linux 命令操作练习，包括文件管理、权限设置等。使用文本编辑器编写简单的脚本，并在 Linux 系统中运行。

教学资源

参考资料：

教材：《物联网智能终端应用程序开发》第 2 章。

Linux 命令大全

Linux 开发工具使用指南

实验设备：计算机、Linux 操作系统。

第 3 章：嵌入式终端的文件 IO 分析与编程操作(12 课时)

教学目标

1、知识目标：

掌握 Linux 系统的文件 I/O 原理和相关的标准库函数。

学会通过编程访问普通文件和目录。

2、能力目标：

能够使用文件 I/O 函数进行设备文件的读写操作。

能够完成完整的嵌入式 Linux 交叉编译实验。

3、素质目标：

培养学生的编程能力和问题解决能力。

4、课程思政目标：

培养数据安全意识：通过文件 IO 操作，引导学生关注数据的保密性、完整性和可用性，树立数据安全意识和责任意识。

增强逻辑思维能力：在编程实践中，培养学生的逻辑思维和问题解决能力，提高分析问题和解决问题的能力。

树立规范操作意识：强调文件操作的规范性和合法性，培养学生遵守法律法规和行业规范的意识。

重难点内容

Linux 系统的文件 I/O 函数：

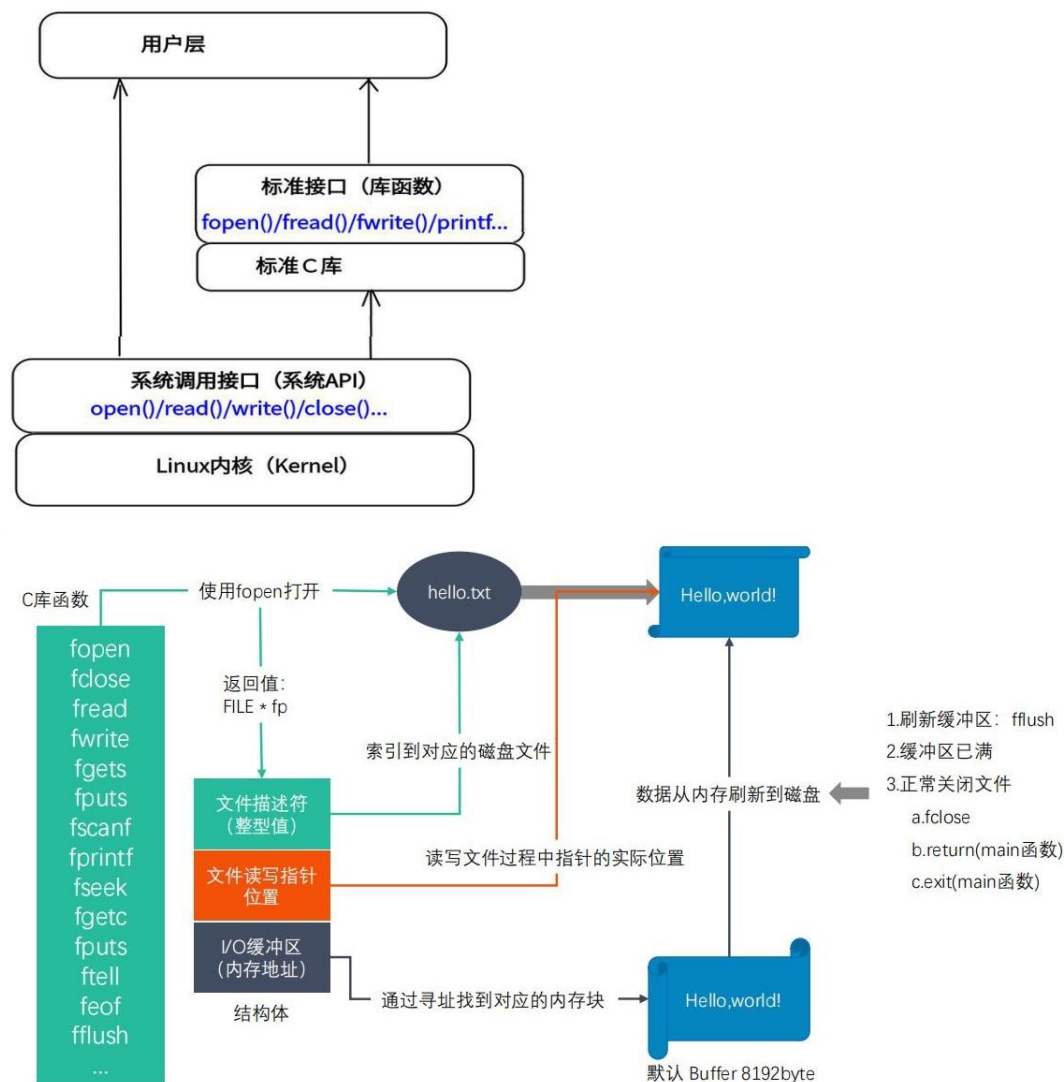
系统调用（open、read、write、close 等）。

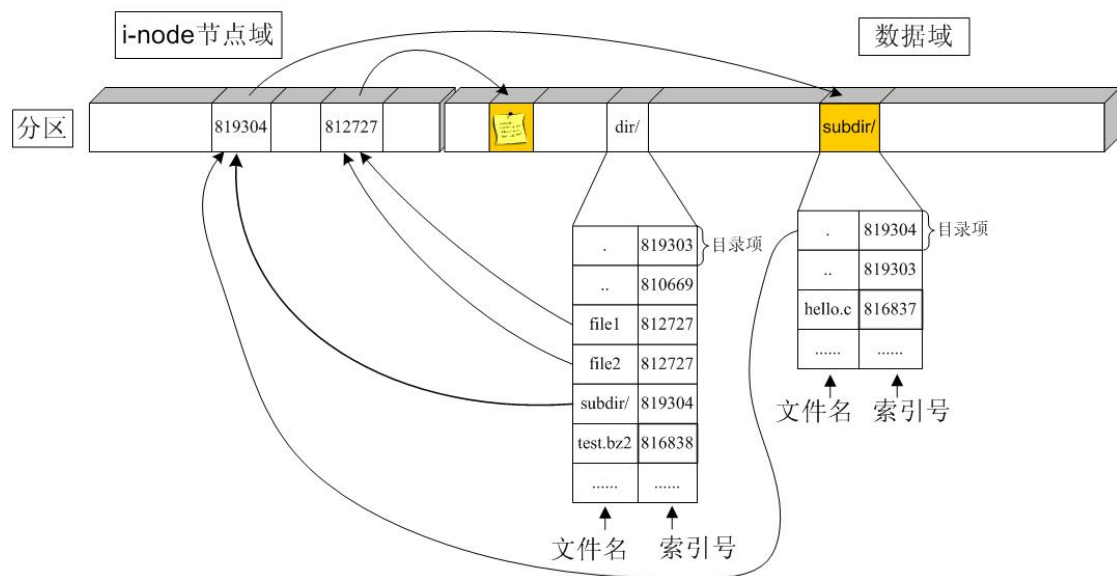
库函数（fopen、fread、fwrite、fclose 等）。

常见设备文件的编程访问：

字符设备（如串口、键盘等）。

块设备（如硬盘、SD 卡等）。





教学方法与过程

讲授法：通过 PPT 讲解文件 I/O 函数的使用方法。

实例分析法：通过实例代码讲解文件、目录等的访问方法。

实践法：学生编写小程序模拟文件的访问。

- 1、导入：通过展示嵌入式设备（如智能手表、智能家居设备等），引出设备访问的重要性。
- 2、新课讲解：使用 PPT 讲解文件 I/O 函数的分类和使用方法。介绍字符设备和块设备的基本概念及其访问方法。
- 3、实例分析：展示简单的访问文件、目录等的程序，讲解代码逻辑。实验原理的讲解和流程梳理。
- 4、实践指导：学生在实验室中分组编写程序，实现对特定文件的读写操作。教师巡视，解答学生在编程过程中遇到的问题。
- 5、总结：总结本章的重点内容，包括文件 I/O 函数的使用方法和设备文件的访问方法。强调在设备访问过程中需要注意的问题。
- 6、布置作业：编写相关程序，使用标准 I/O 实现对特定文件的读写操作。分析并记录文件访问过程中遇到的问题和解决方案。

实验

交叉编译 本地运行

```
root123@bammyga_pc:嵌入式linux交叉编译实验$ gcc 1.c -o 1
root123@bammyga_pc:嵌入式linux交叉编译实验$ arm-linux-gnueabi-gcc 1.c -o arm_1
root123@bammyga_pc:嵌入式linux交叉编译实验$ ./1
hello_world
root123@bammyga_pc:嵌入式linux交叉编译实验$ ./arm_1
-bash: ./arm_1: cannot execute binary file: Exec format error
root123@bammyga_pc:嵌入式linux交叉编译实验$ |
```

文件传送

连接好开发板和pc后，使用超级终端或者Xshell进行传送

注：可查看开发板上没有同名文件后再传送，如有同名文件，可删除后再传

教学资源

教材：《物联网智能终端应用程序开发》第3章。

参考资料：

Linux 设备文件编程

嵌入式设备访问实例

实验设备：计算机、Linux 操作系统、开发板。

第4章：嵌入式终端的设备访问(鼠标、触摸屏)(5课时)

教学目标

1、知识目标：

熟悉 Linux 系统的输入子系统。

学会编程获取鼠标、触摸屏数据。

2、能力目标：

能够使用 Linux 的 input 子系统进行触摸屏数据的读取。

能够编写程序响应鼠标、触摸屏输入。

3、素质目标：

培养学生的编程能力和问题解决能力。

4、课程思政目标：

培养创新思维：通过设备访问编程，引导学生思考如何优化设备交互体验，培养创新思维和用户体验意识。

增强社会责任感：强调设备驱动开发对社会生活的影响，引导学生关注技术对社会的贡献，增强社会责任感。

树立质量意识：在设备访问开发中，注重代码质量和稳定性，培养学生对产品质量的严格要求。

重难点内容

Linux 系统的输入子系统：

输入子系统的框架、驱动程序和 API 的使用。

Linux 的 input 子系统编程：

事件处理（如触摸事件、按键事件等）。

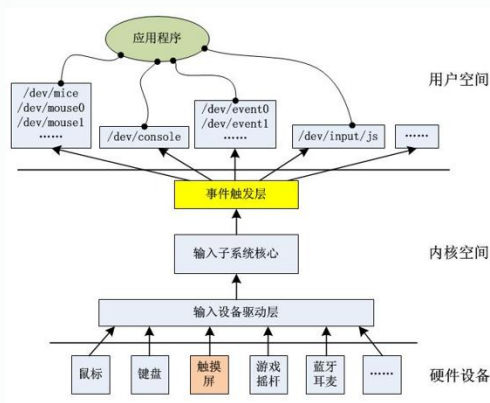
数据读取（如坐标数据、压力数据等）。

Linux 系统的输入子系统

在 Linux 的内核中，对输入设备的使用，通过三层架构来管理，分别是：输入设备驱动层、输入子系统核心层，以及事件触发层。

各自的工作分别是：

- 1.输入设备驱动层：每一种设备都有其特定的驱动程序，被装载到操作系统的设备模型框架内，封装硬件所提供的功能，向上提供规定的接口。
- 2.核心层：将收集由设备驱动层发来的数据，整合之后触发某一事件。
- 3.事件触发层：可以通过在用户空间读取相应设备的节点文件来获知某设备的某一个动作。



鼠标数据读取

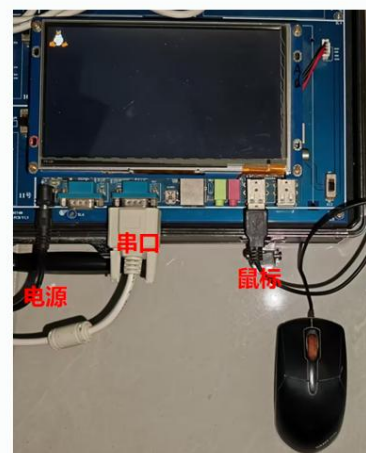
- 1、做好硬件连接后，判别输入设备节点eventX
1) 在超级终端或者xshell上，输入命令 `ls /dev/event*`，确认该目录下是否存在eventX设备节点

```
[root@EmbedSky /]# ls /dev/event*  
/dev/event0 /dev/event1 /dev/event2 /dev/event3 /dev/event4
```

- 2) 执行命令 `cat /dev/event1`，然后按下鼠标左右键、键盘按键或者触摸屏，查看超级终端是否有数据输出；若有数据输出，则event1为对应键盘、鼠标或者触摸屏等输入设备。（ctrl + c退出查看）

```
[root@EmbedSky /]# cat /dev/event4  
i k ' '
```

- 3) 执行命令 `cat /dev/event2`（只改变后面数字编号），利用上面步骤方法，判断其他输入设备的节点。



实验二

触摸屏数据读取

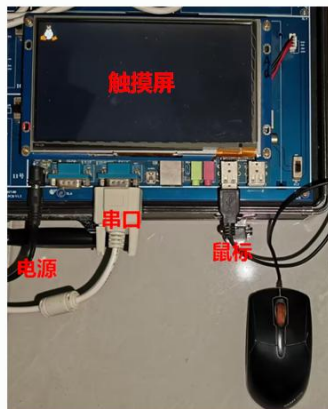
1、做好硬件连接后，判别输入设备节点eventX

1) 在超级终端或者xshell上，输入命令 `ls /dev/event*`，确认该目录下是否存在eventX设备节点

```
[root@EmbedSky ~]# ls /dev/event*  
/dev/event0 /dev/event1 /dev/event2 /dev/event3 /dev/event4
```

2) 执行命令 `cat /dev/event3`，然后按下触摸屏，查看超级终端是否有数据输出；若有数据输出，则event3为对应输入设备。(ctrl + c退出查看)

```
[root@EmbedSky ~]# cat /dev/event3  
#  
0  
n  
uJU  
eSshellU  
gS  
jsU  
u
```



教学方法与过程

讲授法：通过 PPT 讲解输入子系统的编程方法。

实例分析法：通过实例代码讲解鼠标、触摸屏数据的获取方法。

实践法：学生编写小程序模拟鼠标、触摸屏数据的获取。

1、导入：

展示智能终端设备（如智能手机、平板电脑等）的触摸屏操作，引出本章主题。

2、新课讲解：

使用 PPT 讲解 Linux 输入子系统的基本概念和框架。

介绍鼠标、触摸屏数据的类型和读取方法。

3、实例分析：

展示一个简单的鼠标、触摸屏数据获取程序，讲解代码逻辑。

讲解如何处理触摸事件并获取坐标数据。

4、实践指导：

学生在实验室中分组编写程序，实现对鼠标、触摸屏输入的响应。

教师巡视，解答学生在编程过程中遇到的问题。

5、总结：

总结本章的重点内容，包括输入子系统的编程方法和鼠标、触摸屏数据的获取方法。

强调在鼠标、触摸屏数据获取过程中需要注意的问题。

6、布置作业：

编写一个程序，实现对鼠标、触摸屏输入的响应。

分析并记录触摸屏数据获取过程中遇到的问题和解决方案。

教学资源

教材：《物联网智能终端应用程序开发》第 6 章 6.2 节。

参考资料：

Linux 输入子系统

触摸屏编程实例

实验设备：计算机、Linux 操作系统、开发板、触摸屏。

第 5 章：嵌入式终端的 LCD 应用开发(3 课时)

教学目标

1、知识目标：

掌握智能终端的 LCD 控制技术。

学会编程实现 LCD 显示。

2、能力目标：

能够使用 Linux 的 DRM 显示系统进行 LCD 控制。

能够编写程序在 LCD 上显示图像。

3、素质目标：

培养学生的动手能力和创新思维。

4、课程思政目标：

培养审美意识和人文关怀：通过 LCD 应用开发，引导学生关注界面设计的美观性和易用性，培养审美意识和人文关怀。

增强创新意识：鼓励学生在应用开发中尝试新的技术和设计思路，培养创新意识和实践能力。

树立用户至上理念：强调用户体验的重要性，培养学生以用户为中心的设计理念和责任感。

重难点内容

图像原理与 BMP 图

BMP 图片编程处理

颜色最简单的方式就是用RGB或RGBA表示, 如图所示:

RGB	RGB	RGB	RGB	RGB
RGB	RGB	RGB	RGB	RGB
RGB	RGB	RGB	RGB	RGB
RGB	RGB	RGB	RGB	RGB
RGB	RGB	RGB	RGB	RGB

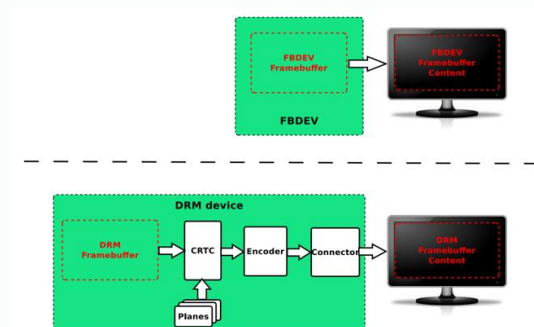
RGBA	RGBA	RGBA	RGBA	RGBA
RGBA	RGBA	RGBA	RGBA	RGBA
RGBA	RGBA	RGBA	RGBA	RGBA
RGBA	RGBA	RGBA	RGBA	RGBA
RGBA	RGBA	RGBA	RGBA	RGBA

Linux 的 DRM 显示系统:

DRM 框架、驱动程序和 API 的使用。

LCD 设备的编程访问:

Linux 的DRM显示系统



传统的显示框架

简单, 但是只能提供最基础的显示功能;
无法满足当前上层应用和底层硬件的显示需求;

目前主流的显示方案

为了适应当前日益更新的显示硬件;
软件上能支持更多高级的控制和特性;

DRM(Direct Render Manager) 即直接渲染管理器。是linux下最新的图形渲染架构, 它是为了解决多个程序对 Video Card 资源的协同使用问题而产生的。它向用户空间提供了一组 API, 用以访问操纵 GPU。概括的几个基本要素: 帧缓冲(FrameBuffer)、绘图控制器(CRTC)、显示面(Planner)、编码器(Encoder)、连接器(Connector)、显示设备 (Display Device)。

实验

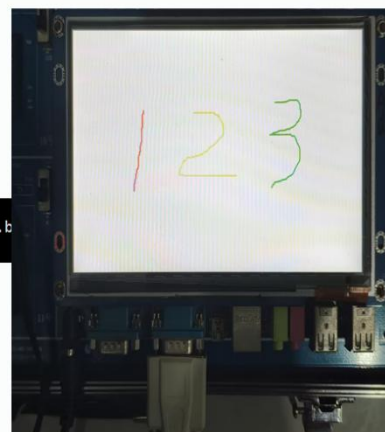
四、虚拟机上交叉编译, 传送可执行文件和bmp文件, 修改权限后执行, 查看显示屏显示内容。

```
$ arm-linux-gnueabi-gcc show_picture.c -o show
$

[root@EmbedSky /]# rz
^Np
rz 3.48 01-27-98 finished.

[root@EmbedSky /]# ls -l bmp_123.bmp show
-rw-r--r-- 1 root root 1152054 Feb 6 2025 bmp_123.b
-rw-r--r-- 1 root root 8432 Feb 6 2025 show

[root@EmbedSky /]# chmod 755 show
[root@EmbedSky /]# ./show
```



教学方法与过程

讲授法：通过 PPT 讲解 LCD 控制技术。

演示法：现场演示 LCD 显示编程过程。

实践法：学生进行 LCD 显示编程实践。

- 1、导入：展示智能终端设备（如智能手机、平板电脑等）的 LCD 显示效果，引出本章主题。
- 2、新课讲解：使用 PPT 讲解图像原理与 BMP 图，Linux 的 DRM 显示系统的基本概念和框架。介绍常见 LCD 设备的接口协议和控制命令。
- 3、演示：现场演示如何使用 DRM API 编写程序控制 LCD 显示。展示一个简单的 LCD 显示程序，讲解代码逻辑。
- 4、实践指导：学生在实验室中分组编写程序，实现在 LCD 上显示静态图像。教师巡视，解答学生在编程过程中遇到的问题。
- 5、总结：总结本章的重点内容，包括 LCD 控制技术和编程方法。强调在 LCD 显示过程中需要注意的问题。
- 6、布置作业：编写一个程序，实现在 LCD 上显示静态图像。分析 LCD 显示过程中可能遇到的问题，并提出解决方案。

教学资源

教材：《物联网智能终端应用程序开发》第 3 章 3.3 节。

参考资料：

Linux DRM 显示系统

LCD 编程实例

实验设备：计算机、Linux 操作系统、开发板、LCD 显示屏。

第 6 章：嵌入式终端的多任务分析和多线程编程(3 课时)

教学目标

1、知识目标：

掌握 Linux 系统的多任务机制。

学会进行多任务分析。

掌握 Linux 系统的多线程编程技术。

学会编写和调试多线程程序。

2、能力目标：

能够分析多任务系统的调度策略。

能够进行多任务系统的性能评估和优化。

能够使用线程创建、同步和通信机制。

能够使用调试工具对多线程程序进行调试。

3、素质目标：

培养学生的系统分析能力和问题解决能力。

4、课程思政目标：

培养系统思维能力：通过多任务和多线程编程，引导学生树立系统思维，学会从整体和局部的角度分析和解决问题。

增强团队协作精神：在多线程编程中，强调线程之间的协同合作，培养学生的团队协作精神和沟通能力。

树立责任感和诚信意识：在编程实践中，引导学生注重代码质量和稳定性，树立对项目负责、对用户负责的责任感和诚信意识。

重难点内容

Linux 系统的多任务机制：

任务调度（如时间片轮转、优先级调度等）。

进程管理（如进程创建、销毁、上下文切换等）。

多任务分析方法：

性能评估（如 CPU 利用率、响应时间等）。

资源分配（如内存分配、I/O 分配等）。

Linux 系统的多线程编程：

线程创建（`pthread_create` 等）。

线程同步（互斥锁、条件变量等）。

线程通信（共享内存、消息队列等）。

多线程程序的调试：

调试工具的使用（如 `gdb`）。

常见问题诊断（如死锁、竞态条件等）。



教学方法与过程

讲授法：通过 PPT 讲解多任务机制和分析方法。

案例分析法：通过实际案例讲解多任务系统的优化方法。

实践法：学生进行多任务系统的分析和调试。

1、导入：

通过展示多任务操作系统（如 Windows、Linux 等）的运行效果，引出本章主题。

2、新课讲解：

使用 PPT 讲解 Linux 系统的多任务机制，包括任务调度和进程管理。

介绍多任务分析的基本方法，如性能评估和资源分配。

使用 PPT 讲解 Linux 系统的多线程编程技术，包括线程创建、同步和通信机制。

介绍多线程程序调试的基本方法。

3、案例分析：

展示一个多任务系统的实际案例，讲解如何分析其调度策略。

讲解如何通过性能评估和资源分配优化多任务系统。

展示一个多线程程序的实例代码，讲解代码逻辑。

讲解如何使用调试工具对多线程程序进行调试。

4、实践指导：

学生在实验室中分组进行多任务系统的分析和调试。

学生在实验室中分组编写多线程程序，实现并发处理任务。

教师巡视，解答学生在分析和调试过程中遇到的问题。

5、总结：

总结本章的重点内容，包括多任务机制和分析方法、多线程编程技术和调试方法。

强调在多任务系统优化和多线程编程过程中需要注意的问题。

6、布置作业：

分析一个多任务系统的调度策略，并提出改进建议。
编写一个简单的多任务程序，并分析其性能。
编写一个多线程程序，实现并发处理任务。
分析并记录多线程编程过程中遇到的问题和解决方案。

教学资源

教材：《物联网智能终端应用程序开发》第 4 章。

参考资料：

Linux 多任务机制

多任务系统优化案例

实验设备：计算机、Linux 操作系统。

第 7 章：多线程数字相册实验(3 课时)

教学目标

1、知识目标：

通过实验加深对多线程编程的理解。
掌握数字相册应用的多线程设计方法。

2、能力目标：

能够实现多线程图片加载与切换控制。
能够编写和调试多线程程序。

3、素质目标：

培养学生的实践能力和团队合作精神。

4、课程思政目标：

培养创新实践能力：通过数字相册实验，引导学生将理论知识与实际应用相结合，培养创新实践能力和解决问题的能力。

增强用户体验意识：关注数字相册的用户体验，培养学生以用户为中心的设计理念和人文关怀。

树立环保意识：引导学生思考数字相册对传统纸质相册的替代作用，树立环保意识和可持续发展理念。

重难点内容

任务介绍

功能：在触摸屏上显示图片，当检测到触摸信号时，切换显示另一张图片

实验目标：

- 1、实时采集触摸屏数据
- 2、智能终端LCD显示bmp图片信息

思路：开启两个线程，分别执行：采集触摸屏数据 与 智能终端LCD显示bmp图片

主要技能：bmp图片显示、触摸屏区域判定

数字相册应用的多线程设计：

多线程架构设计（主线程、工作线程等）。

图片加载与切换控制（线程同步与通信）。

多线程程序的调试：

调试工具的使用（如 `gdb`）。

常见问题诊断（如死锁、竞态条件等）。

教学方法与过程

讲授法：通过 PPT 讲解数字相册应用的多线程设计方法。

实践法：学生进行多线程数字相册的开发实践。

案例分析法：通过实际案例讲解多线程程序的调试方法。

1、导入：

展示一个简单的数字相册应用界面，引出本章主题。

2、新课讲解：

使用 PPT 讲解数字相册应用的多线程设计架构。

介绍图片加载与切换控制的实现方法。

3、实践指导：

提供数字相册项目的基本代码框架。

学生分组完成多线程图片加载功能的实现。

教师巡视，解答学生在开发过程中遇到的问题。

4、总结：

总结本章的重点内容，包括数字相册应用的多线程设计方法。

强调在多线程开发过程中需要注意的问题。

5、布置作业：

完成数字相册应用的多线程开发，提交可运行程序及实验报告。

教学资源

教材：《物联网智能终端应用程序开发》第 4 章。

参考资料：

多线程数字相册设计

多线程程序调试指南

实验设备：计算机、Linux 操作系统。

第 8 章：多进程编程与音视频播放器实验（实验可选）(2 课时)

教学目标

1、知识目标：

掌握 Linux 系统的多进程编程技术。

学会编写和调试多进程程序。

2、能力目标：

能够使用进程创建、通信和同步机制。

能够使用调试工具对多进程程序进行调试。

3、素质目标：

培养学生的编程能力和问题解决能力。

4、课程思政目标：

培养技术应用能力：通过音视频播放器实验，引导学生将多进程编程技术应用于实际项目，培养技术应用能力和创新意识。

增强社会责任感：强调音视频播放器对文化传播和社会娱乐的影响，引导学生关注技术对社会的积极贡献，增强社会责任感。

树立知识产权意识：在音视频播放器开发中，引导学生尊重知识产权，遵守法律法规，树立正确的价值观。

重难点内容

Linux 系统的多进程编程：

进程创建（fork、exec 等）。

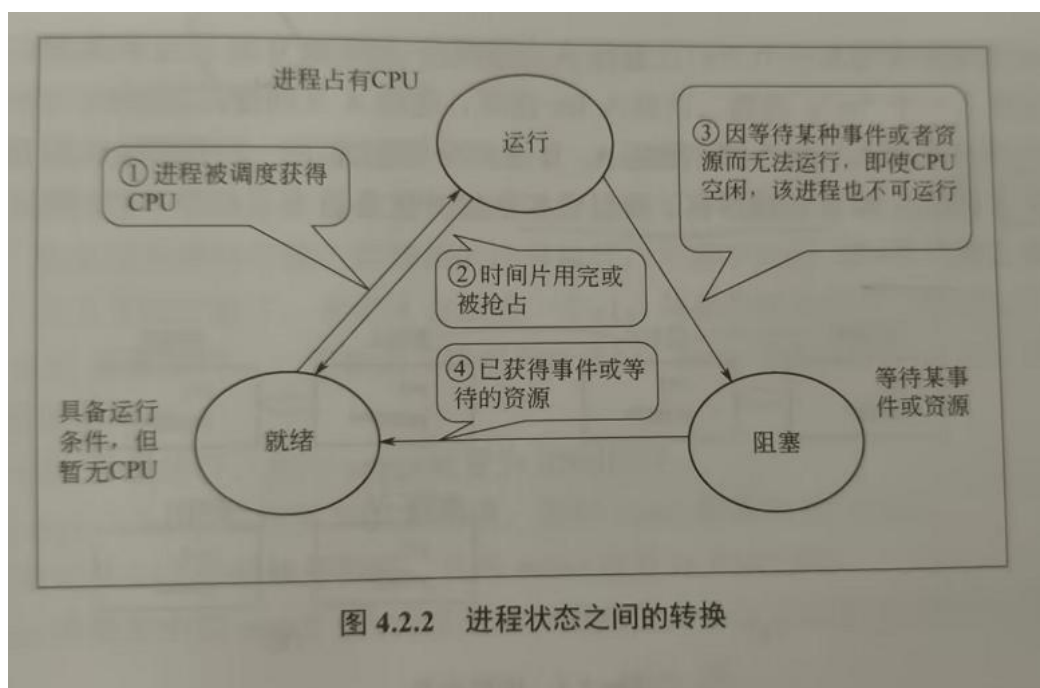
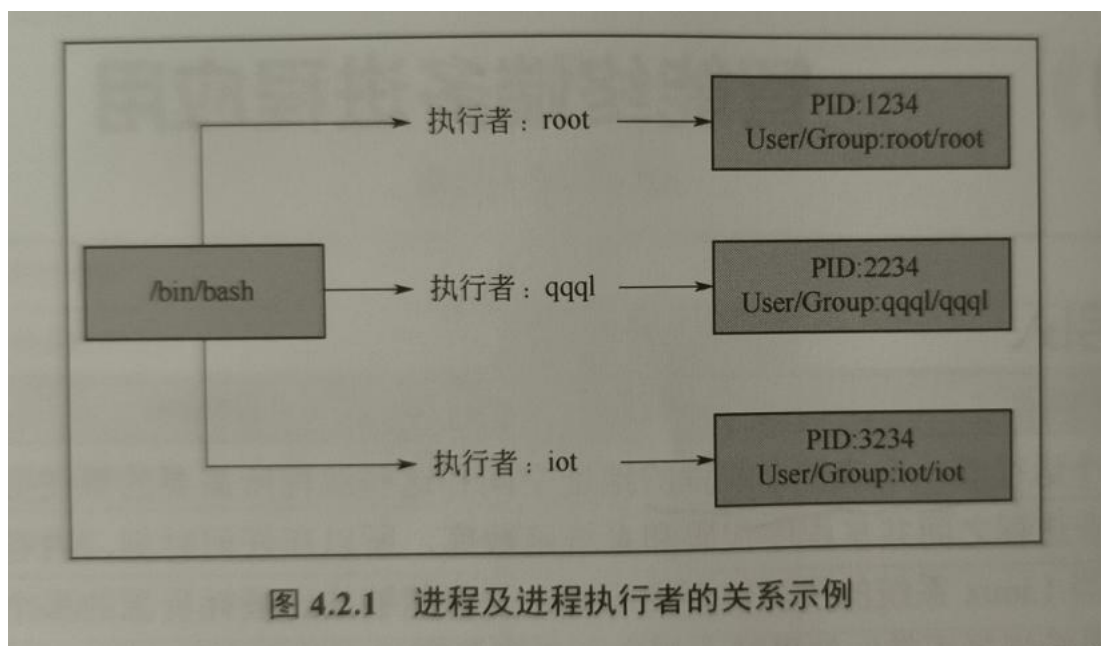
进程通信（管道、消息队列、共享内存等）。

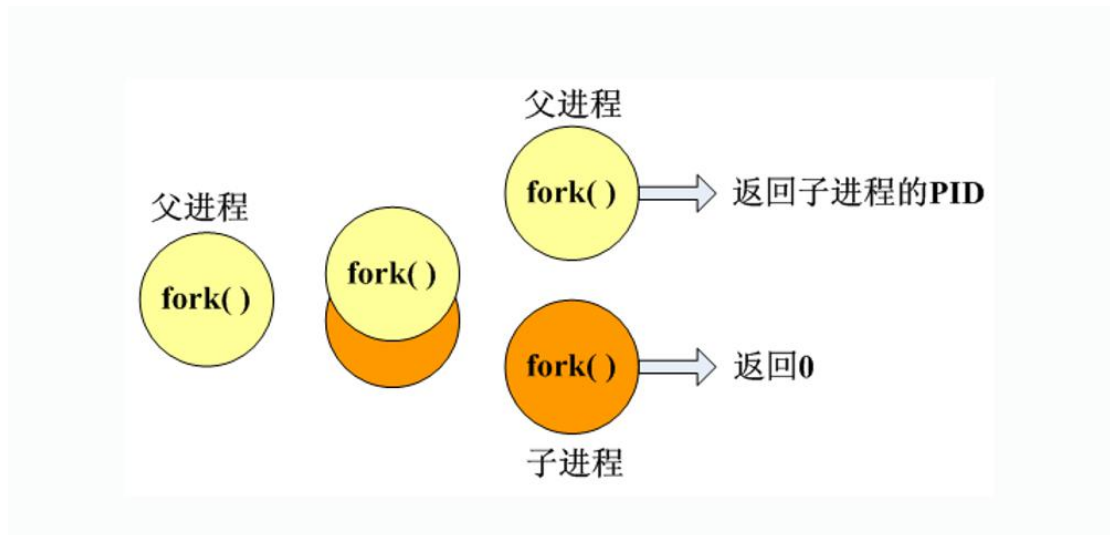
进程同步（信号量等）。

多进程程序的调试：

调试工具的使用（如 gdb）。

常见问题诊断（如僵尸进程、死锁等）。





教学方法与过程

讲授法：通过 PPT 讲解多进程编程的关键技术。

实例分析法：通过实例代码讲解多进程程序的编写和调试方法。

实践法：学生编写多进程程序并进行调试实践。

1、导入：

通过展示多进程程序的运行效果，引出本章主题。

2、新课讲解：

使用 PPT 讲解 Linux 系统的多进程编程技术，包括进程创建、通信和同步机制。

介绍多进程程序调试的基本方法。

3、实例分析：

展示一个多进程程序的实例代码，讲解代码逻辑。

讲解如何使用调试工具对多进程程序进行调试。

4、实践指导：

学生在实验室中分组编写多进程程序，实现并发处理任务。

教师巡视，解答学生在编程和调试过程中遇到的问题。

5、总结：

总结本章的重点内容，包括多进程编程技术和调试方法。

强调在多进程编程过程中需要注意的问题。

6、布置作业：

编写一个多进程程序，实现并发处理任务。

分析并记录多进程编程过程中遇到的问题和解决方案。

教学资源

教材：《物联网智能终端应用程序开发》第 4 章。

附加：音视频播放器实验（可选实验部分）

（一）、音频播放器实验

教学目标

1、知识目标：

掌握音频播放器的开发流程。

学会使用 madplay 音频播放器工具。

理解用户界面设计与交互的基本概念。

madplay用于播放音频

```
[root@EmbeddedSky /player]# madplay -help
Usage: madplay [OPTIONS] FILE [...]
Decode and play MPEG audio FILE(s).

Verbosity:
  -v, --verbose           show status while decoding
  -q, --quiet             be quiet but show warnings
  -Q, --very-quiet        be quiet and do not show warnings
  --display-time=MODE     use default verbose time display MODE
                          (remaining, current, overall)

Decoding:
  --downsample            reduce sample rate 2:1
  -i, --ignore-crc        ignore CRC errors
  --ancillary-output=PATH write ancillary data to PATH

Audio output:
  -o, --output=[TYPE:]PATH write output to PATH with format TYPE (below)
  -b, --bit-depth=DEPTH   request DEPTH bits per sample
  -R, --sample-rate=HERTZ request HERTZ samples per second
  -d, --no-dither         do not dither output PCM samples
```

2、能力目标：

能够开发一个简单的音频播放器应用。

能够实现音频文件的加载、播放、暂停等功能。

3、素质目标：

培养学生的实践能力和创新思维。

重难点内容

音频播放器的开发流程：

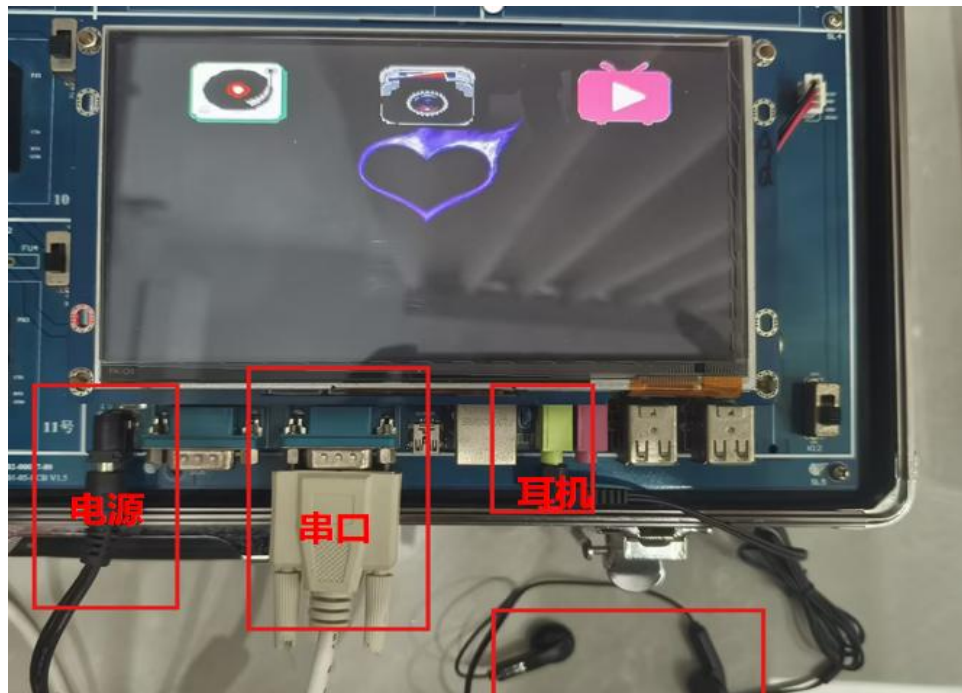
音频文件格式（MP3、WAV 等）。

音频解码与播放（madplay 工具）。

用户界面设计与交互：

界面设计工具（如 Qt Designer）。

交互逻辑实现（播放、暂停、停止等按钮功能）。



教学方法与过程

讲授法：通过 PPT 讲解音频播放器的开发流程。

演示法：现场演示音频播放器的开发过程。

实践法：学生进行音频播放器的开发实践。

1、导入：

展示一个简单的音频播放器应用界面，引出本章主题。

2、新课讲解：

使用 PPT 讲解音频播放器的开发流程，包括音频文件格式、解码工具（madplay）。

介绍用户界面设计的基本概念和工具。

3、演示：

现场演示音频播放器的开发过程，包括界面设计和功能实现。

展示如何使用 madplay 工具进行音频解码和播放。

4、实践指导：

提供音频播放器项目的框架代码。

学生分组完成音频播放器的开发，实现音频文件的加载、播放、暂停等功能。

教师巡视，解答学生在开发过程中遇到的问题。

5、总结：

总结本章的重点内容，包括音频播放器的开发流程和用户界面设计。

强调在音频播放器开发中需要注意的问题。

6、布置作业：

完成音频播放器应用的开发，提交可运行程序及实验报告。

教学资源

参考资料：

音频播放器开发指南

madplay 工具使用手册

实验设备：计算机、Linux 操作系统、音频文件。

(二)视频播放器实验

教学目标

1、知识目标：

掌握视频播放器的开发流程。

学会使用 mplayer 视频播放器工具。

理解用户界面设计与交互的基本概念。

mplayer一个多媒体播放器，可播放音频或者视频

```
[root@EmbedSky /player]# mplayer -help
MPlayer dev-CVS-060102-15:04-4.4.6 (C) 2000-2005 MPlayer Team
CPU: ARM

Usage:  mplayer [options] [url|path/]filename

Basic options: (complete list in the man page)
-vo <drv[:dev]>  select video output driver & device ('-vo help' for a list)
-ao <drv[:dev]>  select audio output driver & device ('-ao help' for a list)
vcd://<trackno>  play (S)VCD (Super Video CD) track (raw device, no mount)
dvd://<titleno>  play DVD title from device instead of plain file
-alang/-slang    select DVD audio/subtitle language (by 2-char country code)
-ss <timepos>    seek to given (seconds or hh:mm:ss) position
-nosound         do not play sound
-fs             fullscreen playback (or -vm, -zoom, details in the man page)
-x <x> -y <y>     set display resolution (for use with -vm or -zoom)
-sub <file>      specify subtitle file to use (also see -subfps, -subdelay)
-playlist <file> specify playlist file
-vid x -aid y    select video (x) and audio (y) stream to play
-fps x -srate y  change video (x fps) and audio (y Hz) rate
-pp <quality>    enable postprocessing filter (details in the man page)
-framedrop       enable frame dropping (for slow machines)
```

2、能力目标：

能够开发一个简单的视频播放器应用。

能够实现视频文件的加载、播放、暂停等功能。

3、素质目标：

培养学生的实践能力和创新思维。

重难点内容

视频播放器的开发流程：

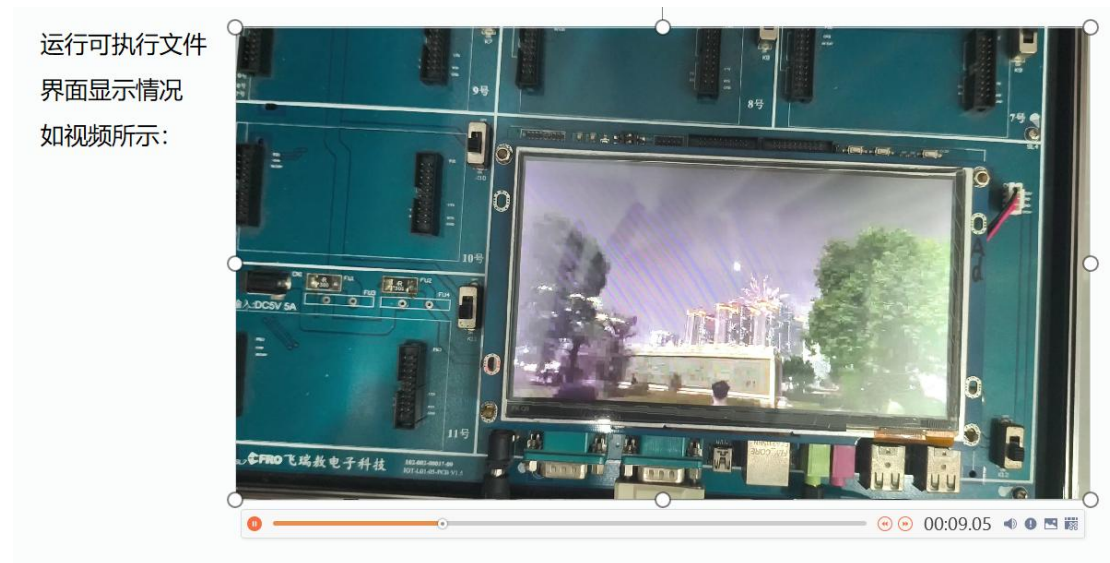
视频文件格式（MP4、AVI 等）。

视频解码与播放（mplayer 工具）。

用户界面设计与交互：

界面设计工具（如 Qt Designer）。

交互逻辑实现（播放、暂停、停止等按钮功能）。



教学方法与过程

讲授法：通过 PPT 讲解视频播放器的开发流程。

演示法：现场演示视频播放器的开发过程。

实践法：学生进行视频播放器的开发实践。

1、导入：

展示一个简单的视频播放器应用界面，引出本章主题。

2、新课讲解：

使用 PPT 讲解视频播放器的开发流程，包括视频文件格式、解码工具（mplayer）。

介绍用户界面设计的基本概念和工具。

3、演示：

现场演示视频播放器的开发过程，包括界面设计和功能实现。

展示如何使用 mplayer 工具进行视频解码和播放。

4、实践指导：

提供视频播放器项目的框架代码。

学生分组完成视频播放器的开发，实现视频文件的加载、播放、暂停等功能。

教师巡视，解答学生在开发过程中遇到的问题。

5、总结：

总结本章的重点内容，包括视频播放器的开发流程和用户界面设计。
强调在视频播放器开发中需要注意的问题。

6、布置作业：

完成视频播放器应用的开发，提交可运行程序及实验报告。

教学资源

参考资料：

视频播放器开发指南

mplayer 工具使用手册

实验设备：计算机、Linux 操作系统、视频文件。