

麒麟 OS 环境下基于 Qt 的驱动程序学习系统^①



王雪文¹, 刘杰¹, 兰雨晴^{2,3}

¹(首都师范大学信息工程学院, 北京 100048)

²(北京航空航天大学计算机学院, 北京 100191)

³(麒麟软件有限公司, 北京 100190)

通讯作者: 王雪文, E-mail: 18810879172@163.com

摘要: 随着基础软件国产化的浪潮, 麒麟操作系统已经应用于各个行业. 本文基于 Qt 提供的强大的 C++ API 框架, 开发运行于国产麒麟 OS (Operating System) 上的内核驱动程序学习系统, 弥补了麒麟 OS 环境下缺乏驱动程序学习软件的不足. 主要研究内核中几百个驱动程序的执行流程, Linux 内核驱动程序的一般体系结构、内核驱动程序的详细分类、驱动程序配置与调试技术以及应用程序、内核、硬件、驱动程序之间的关系等驱动程序学习机理. 在实现过程中系统基于模块化设计思想, 采用分层实现软件架构. 具体实现网络请求、远程文件下载与上传、内核驱动程序一般架构、驱动程序详细分类、驱动程序学习机理、视频学习、具体驱动程序、系统设置 (包括计算机系统信息和系统升级检测两个子模块) 等 8 大模块. 最后采用交叉编译方式, 使其部署到 X86、ARM、MIPS 三种不同的架构平台的计算机上.

关键词: 麒麟 OS; Qt 技术; Linux 内核; 驱动学习系统; 交叉编译

引用格式: 王雪文, 刘杰, 兰雨晴. 麒麟 OS 环境下基于 Qt 的驱动程序学习系统. 计算机系统应用, 2021, 30(3): 70-78. <http://www.c-s-a.org.cn/1003-3254/7849.html>

Driver Learning System Based on Qt in Environment of Kylin OS

WANG Xue-Wen¹, LIU Jie¹, LAN Yu-Qing^{2,3}

¹(Information Engineering College, Capital Normal University, Beijing 100048, China)

²(School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

³(KylinSoft, Beijing 100190, China)

Abstract: Amid the wave of basic software localization, Kylin Operating System (OS) has been applied to industries. With the powerful C++ API framework provided by Qt, this study develops a kernel driven learning system running on the domestic Kylin OS to make up for the lack of driver learning software. This study mainly focuses on the execution process of hundreds of drivers in the kernel, the general architecture of Linux kernel drivers, the detailed classification of kernel drivers, driver configuration and debugging technology, the relationship among applications, kernel, hardware, drivers, and other driver learning mechanisms. The system is implemented based on modular design with layered software architecture. It enables network request, remote file download/upload, general architecture of kernel driver, detailed driver classification, driver learning mechanism, video learning, specific driver, system settings (including two sub modules of computer system information and system upgrade detection). Finally, it is applied to the computers of X86, ARM, and MIPS through cross compilation.

Key words: Kylin OS; Qt technology; Linux kernel; driver learning system; cross compilation

① 基金项目: 国家自然科学基金面上项目 (61672361); 北京市教委-市自然科学基金联合资助项目 (KZ201910028039)

Foundation item: General Program of National Natural Science Foundation of China (61672361); Joint Fund of Beijing Municipal Education Commission and Natural Science Foundation (KZ201910028039)

收稿时间: 2020-07-08; 修改时间: 2020-08-11, 2020-08-21; 采用时间: 2020-09-04; csa 在线出版时间: 2021-03-03

1 引言

2018年10月,倪光南院士在首届新智慧城市建设峰会上,再次提到发展国产操作系统的重要性。他提到,“网络信息与网络安全越来越具备时代特征,自主可控替代变得尤为重要”。2020年3月20号,在麒麟软件“遨天”计划发布会上,倪光南强调,在构建安全可控的国产信息技术体系过程中,操作系统是软件之魂,大国重器。麒麟OS是基于Linux的二次开发,Linux kernel是整个操作系统的最底层,也是操作系统最基础的部分,OS是不能够直接控制硬件的,设备驱动是操作系统和硬件设备间的粘合剂,没有驱动程序大多硬件设备无法工作,驱动程序在内核中占据约50%的代码,起着至关重要的作用。众所周知,驱动程序很难开发和调试,这对硬件产品的发布具有负面影响,因为上市时间通常由驱动程序开发和验证计划控制^[1]。在这样的形势下麒麟操作系统需要大量的新进内核驱动程序维护和开发人员,但麒麟OS下还没有供研发人员和用户使用的驱动程序学习软件。本文旨在研究、设计和开发麒麟OS环境下基于Qt的内核驱动程序学习系统。此系统采用SQLite数据库、FTP服务器、HTTP和TCP协议、Qmedia视频框架、SQLTableModel表格模型、交叉编译等,在驱动程序信息提取部分,借助多模式串匹配AC算法,为使算法的执行时间减少,提出了一种改进的AC算法简称为S_AC。此系统有效弥补了驱动程序学习部分的短缺,经验证系统能稳定运行。麒麟OS环境下基于Qt的内核驱动程序学习系统,能体现出其巨大的价值,降低企业新人的学习周期,能够节省工程师对内核驱动程序的维护时间,降低企业运营成本。更深一层讲,有助于推进国产操作系统版本的发布和维护,从而有利于国产操作系统的推广和普及,更好的保障国家信息、金融、国防等行业的安全,使网络信息与网络安全越来越具备时代特征,实现真正的自主可控、安全可靠。

2 相关简介

2.1 国产操作系统、麒麟OS简介

为保证国家信息安全,必须大力发展自主可控技术,即信息系统从硬件到软件都是自主研究、创造、开发、维护,从而实现真正的信息安全可控。发展国产操作系统是势在必行,很有前景的工作,发展国产操作系统是一件利国利民、大有裨益的事业。现在中美贸

易摩擦局势紧张,众所周知从2018年到现在中美一直打贸易战,美国针对中国的一些制造业以及高科技产业加收高额关税,对中国的一些企业影响较大,比如华为,阿里巴巴等大公司,特朗普曾说,美国最不能接受的事情是在高科技产业上被中国取代或反超。有了真正属于自己的操作系统,我们在很多方面就可以不用受制于人,可以保证我们的金融、国防、信息等行业的安全。20世纪90年代初,国家有关部门就开始倡导发展自主操作系统,国产操作系统企业努力了多年,目前在产业规模、支撑应用能力等多个方面有了很大的提高^[2]。

麒麟OS是名气较大和使用人数最多的国产操作系统。据统计麒麟OS已经连续八年国内Linux市场占有率第一。麒麟操作系统系列产品主要以操作系统技术为核心,安全可信为特色,其产品已经在政府、国防、金融、教育、财税、公安、审计、交通、医疗、制造等行业得到深入应用,应用领域涉及我国信息化和民生各个方面,多个领域已经进入核心应用部分。

麒麟操作系统具有高安全、高可靠、高可用、跨平台以及强大的中文处理能力等优点,被越来越多的用户所接受。2020年3月麒麟软件发布了“5年‘遨天’行动计划”,旨在打造国家操作系统的最强力量,持续为用户提供高质量服务,让操作系统带动我国信息产业市场规模和技术创新的发展。麒麟“遨天”行动将带动万亿产业规模,服务国家关键基础设施建设,推动网安产业变革进步,用五年时间,培育一支规模逾万人的自主操作系统精英团队,积极推动联合创新和开放发展,走一条内生安全、融入移动、注重体验、支撑体系的特色发展道路。麒麟OS作为国内领先国产操作系统,在促进基础软件国产化以及信息技术自主可控方面义不容辞。

在操作系统方面,暂时还做不到完全自主开发,也没有这个必要,开源Linux内核功能完善,安全性更强^[3],多用户多任务独立运行,具有良好的用户界面,设备独立性强,同时具有良好的可移植性。开源Linux是一个很好的基础,针对它进行二次开发,可以给我们节省不少人力和财力。中国工程院院士倪光南曾表示,国产操作系统的竞争力很大程度上取决于其生态系统的建设。其实和国产芯片一样,包括麒麟OS在内的国产操作系统面临的不仅是技术问题,更重要的是生态。如果生产出的OS没有生态支持,就很难推广。所以接下来需要大力发展生态,适配更多的软硬件,并加大自主创

新,为开源社区贡献更大的力量,同时增加我们在开源基金会、开源社区的话语权。

2.2 内核驱动简介

内核,是一个操作系统的核心,是操作系统的基础,它负责整个硬件的驱动,以及提供各种系统所需的核心功能。计算机真正工作的东西其实是硬件,如果内核不认识某个最新的硬件,那么硬件也就无法被驱动,计算机也就无法使用该硬件^[4]。

设备驱动程序(device driver),是一种可以使计算机和设备进行相互通信的特殊程序。一般而言,驱动程序负责以下3个任务:(1)配置和管理一个或多个设备;(2)将来自内核的请求转换为对硬件的请求;(3)将结果从硬件传递到内核。

内核代码3000多万行,其中驱动程序占了大约50%的代码。正是因为驱动程序的重要性以及其相对较难开发的特点,国内外越来越多专家、学者对Linux内核及设备驱动投入大量精力研究。清华大学信息科学与技术国家实验室的Bai等^[5]提出了从原始被动驱动程序代码到自动生成高效的主动驱动程序的方法AutoPA,AutoPA使用源代码系统来实现驱动程序转换,还使用改进的主动驱动程序体系结构来减少性能下降。驱动程序对于操作系统来说至关重要。

3 驱动程序学习系统的设计及实现

3.1 系统总体结构设计

系统总体结构如图1所示,系统分为展示层、功能层、数据层。其中,展示层即Qt界面层,多个UI用来展示界面信息;功能层主要是8大功能模块:网络请求、远程文件下载与上传、内核驱动一般架构、驱动详细分类、驱动学习机理、视频学习、具体驱动、设置(计算机信息和系统升级检测)。数据层主要用于服务器学习文件和视频文件的存储,以及SQLite、MySQL数据库中内核驱动程序信息、驱动程序详细分类的存储。在展示层通过点击功能按钮,触发Qt的信号和槽机制,点击按钮时会发出一个或多个信号,当信号发出时,与之相连的槽函数会自动回调,从而完成展示层与功能层的连接,将每个点击事件、滚动事件等与各个功能模块贯通。数据层与各功能模块密不可分,用来存储各个功能模块的数据信息,比如具体驱动模块的几百条驱动信息存储在SQLite数据库,共享的驱动程序学习文件以及较大的视频学习文件存储在远程FTP服务器上,当用户需要时则可进入对应功能模块进行下

载查看。数据层的数据信息通过各功能模块加工、改造间接展现在展示层的UI界面上,以使用户查看。

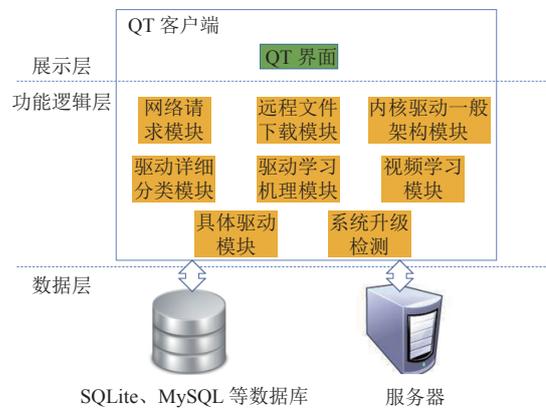


图1 系统总体结构图

开发流程图如图2所示,需求分析是软件计划阶段的重要活动,也是软件生存周期中的一个重要环节,在设计驱动程序学习系统之前,首先要做的就是需求分析,确定系统实现的功能,完成的工作。系统设计包括架构设计、功能设计、应用程序接口设计等,在系统设计过程中首先确定本发明的架构,架构图是顶层设计,如果没有顶层设计,统一合理地把软件的功能需求和非功能需求分配到软件部件当中,那么最后将不是一个好的软件。其次是功能设计,功能设计根据系统的需求将功能落地,再次是应用程序接口设计。再就是编码设计,即编程实现,为了适应X86、ARM、MIPS三种架构平台,需要交叉编译脚本,进行交叉编译。最后是应用部署,包括部署、测试和优化设计,在本地机器上安装,进行测试,根据实际问题反馈,然后优化设计,再进行安装测试,直到最优。

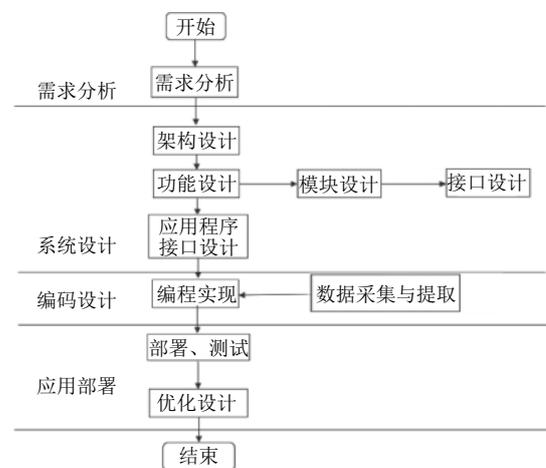


图2 系统开发流程图

3.2 系统功能模块设计

系统的功能图如图3所示,该驱动程序学习系统的软件部分包含网络请求、远程文件下载与上传、内核驱动一般架构、驱动详细分类、驱动学习机理、视频学习、具体驱动、设置(计算机信息和系统升级检测)8大功能模块,以下具体介绍各个模块的作用以及实施机制。

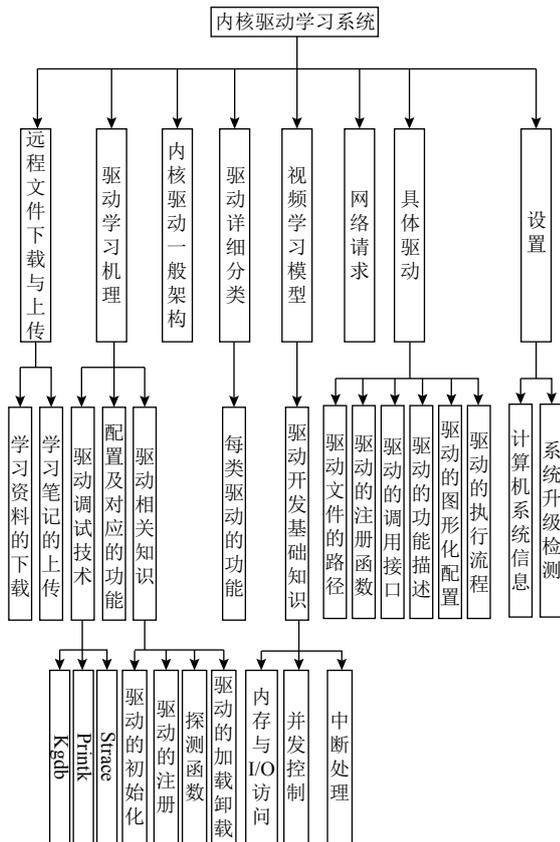


图3 系统功能模块

3.2.1 远程文件上传与下载模块主要功能及实现机制

远程文件的上传与下载模块有两个子功能模块,分别是学习资料的下载和学习笔记的上传,具体实施是创建FTP服务器账号和密码,在服务器上创建学习资料库用于用户选择下载所需的驱动学习资料,创建用户个人库,用于用户学习笔记等文件的存储。远程文件的上传与下载模块的Qt界面设计如图4所示。用户通过服务器地址、用户名和密码登录FTP服务器,建立FTP的连接,在Qt界面的Tree Widget控件中显示FTP服务器上可以公用下载的资源,同时可以显示资源文件名称、大小、拥有者、所属组以及修改日期等。用户选好资料点击下载按钮,下载到本地计算机,通过Progress Bar控件可以显示下载进度,选择本地的学习

笔记,通过点击上传按钮,上传至服务器的用户个人空间,同时Progress Bar控件显示上传进度。



图4 远程文件的上传与下载模块的Qt界面

下面是连接FTP服务器的代码,先通过服务器地址和端口号建立连接,再通过用户名和密码进行登录。

```
void Download::on_connectButton_clicked()
{
    ui->fileList->clear();
    currentPath.clear();
    isDirectory.clear();
    ui->progressBar->setValue(0);
    ftp = new QFtp(this);
    connect(ftp, & QFtp::commandStarted, this,
            & Download::ftpCommandStarted);
    connect(ftp, & QFtp::commandFinished, this,
            & Download::ftpCommandFinished);
    connect(ftp, & QFtp::listInfo, this,
            & Download::addToList);
    connect(ftp, & QFtp::dataTransferProgress, this,
            & Download::updateDataTransferProgress);
    QString ftpServer = ui->ftpServerLineEdit->text();
    QString userName = ui->userNameLineEdit->text();
    QString passWord = ui->passWordLineEdit->text();
    ftp->connectToHost(ftpServer, 21);
    ftp->login(userName, passWord);
}
```

下面代码是下载按钮的槽函数,下载前先进行编码转换,因为文件名可能是中文,然后通过文件名进行下载。

```

void Download::on_downloadButton_clicked()
{
//在使用 get() 函数前需要进行编码转换
QString fileName =
    ui->fileList->currentItem()->text(0);
QString name =
    QLatin1String(fileName.toUtf8());
file = new QFile(fileName);
if (!file->open(QIODevice::WriteOnly))
{
    delete file;
    return;
}
ui->downloadButton->setEnabled(false);
ftp->get(name, file);
}

```

3.2.2 驱动学习模块主要功能及实施机制

驱动学习机理模块包含驱动程序调试技术、内核配置及对应的功能、驱动程序编写相关知识 3 个子模块, 各子模块的内容通过 Text Browser 控件显示, 方便用户浏览。其中驱动程序调试技术包括内核追踪 kgdb、打印调试 printk、调试工具 strace 等, kgdb、printk、strace 等都是常用的内核开发及维护的调试技术, 但对于新进开发人员来说并不那么容易掌握, 网上资源零乱且有限, 该模块对各个调试技术有具体介绍和应用举例, 以使用户学习。

内核配置及对应功能子模块包括内核常用配置及相应注解, 以及每项配置的作用, 用户通过此功能模块可以了解并掌握常用内核配置, 以便在实际操作中运用。

驱动编写子模块包括驱动程序的初始化、驱动程序的注册、探测函数、驱动程序的加载和卸载, 同时还有关于 MakeFile 文件编写的介绍。

3.2.3 内核驱动一般架构模块主要功能及实施机制

内核驱动一般架构模块功能设计如图 5 所示, 在此功能模块中绘制内核的一般架构图, 包含用户空间、内核空间、硬件部分, 点击详细介绍按钮, 触发点击事件, 进入点击事件槽函数, 用户可以看到架构图中系统调用接口、进程管理、内存管理、虚拟文件系统、网络接口、进程间通信、块设备、字符设备、网络设备以及设备驱动程序等的详细介绍, 有助于用户深入理解内核驱动一般架构图。

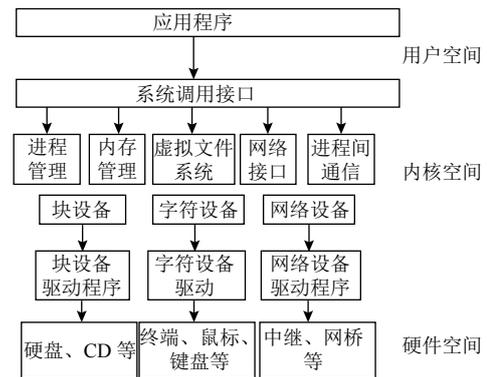


图 5 内核驱动一般架构模块功能设计

3.2.4 具体驱动模块主要功能及实施机制

具体驱动模块包含内核中几百种驱动程序的驱动程序名称、文件路径、功能描述、注册函数、程序执行流程等详细信息, 这些数据信息存储在数据层的 SQLite 数据库, 此模块用到了 QSqlTableModel 表格模型以及视图, 用于显示和处理数据库数据。下面代码的功能是输入驱动程序名称, 查询某个具体驱动的详细信息。

```

void juti::on_checkBtn_clicked()
{
if(ui->lineEdit->text().isEmpty()){
    QMessageBox::information(this, tr("提示!"),
tr("请先输入模块或驱动名称!"),
    QMessageBox::Ok); ui->lineEdit->setFocus();
} else {
    QString name = ui->lineEdit->text();
// 根据模块名称进行筛选
// 一定要使用单引号
model->setFilter(QString("名称 = '%1'").arg
(name));
model->select();
}
}

```

3.2.5 视频学习模块主要功能及实施机制

视频学习模块, 将驱动程序开发的基础知识的相关视频, 存储到 FTP 服务器上, 用户点击视频学习, 程序将自动连接服务器获取相关视频, 并加载到 Qt 视频界面中的视频播放列表, 用户通过鼠标点击触发事件, 选择播放的视频, 在此模块中用到了 Qt 的事件系统-鼠标事件、定时器、信号和槽机制等。下面代码是此模块各个信号和槽函数的 connect() 连接。

```

QObject::connect(playAudio, SIGNAL(clicked()),
    this, SLOT(playVideo()));
QObject::connect(stopAudio, SIGNAL(clicked()),
    this, SLOT(stopVideo()));
QObject::connect(pauseAudio, SIGNAL(clicked()),
    this, SLOT(pauseVideo()));
connect(brightslider, & QSlider::sliderMoved,
    videoWidget, & QVideoWidget::setBrightness);
connect(videoWidget, & QVideoWidget::brightness
    Changed, brightslider, & QSlider::setValue);
connect(contrastslider, & QSlider::sliderMoved,
    videoWidget, & QVideoWidget::setContrast);
connect(videoWidget, & QVideoWidget::contrast Changed,
    contrastslider, & QSlider::setValue);
connect(hueslider, & QSlider::sliderMoved,
    videoWidget, & QVideoWidget::setHue);
connect(videoWidget, & QVideoWidget::hueChanged,
    hueslider, & QSlider::setValue);
QObject::connect(listView, SIGNAL(clicked(
    QModelIndex)), this, SLOT(showClick(QModelIndex)));
QObject::connect(sliderDuration, SIGNAL(costomSlider
    Clicked()), this, SLOT(sliderProgressClicked()));
QObject::connect(sliderDuration, SIGNAL(sliderMoved()),
    this, SLOT(sliderProgressMoved()));
QObject::connect(sliderDuration, SIGNAL(sliderReleased()),
    this, SLOT(sliderProgressReleased()));

```

3.2.6 网络请求模块主要功能及实施机制

网络请求模块, 主要功能是请求网络, 确认网络是否连接, 是由 QNetworkRequest 类表示. 下面是判断网络是否连接的代码, 能 ping 通百度说明网络已连接, 通过 QTable 显示.

```

QString network::Iplive()
{
    QTcpSocket tcpClient;
    tcpClient.abort();
    tcpClient.connectToHost("202.108.22.5", 80);
    //1000 毫秒没有连接上则判断不在线
    QString m;
    if(tcpClient.waitForConnected(1000))
    {
        m="网络已连接";
    }
}

```

```

}else{
    m="无法联网";
}
return m;
}

```

3.2.7 驱动详细分类模块主要功能及实施机制

内核中有成百上千个驱动程序, 但这些驱动不是杂乱无章而是有规则分类的, 此模块包含 108 类驱动程序的功能描述, 便于用户分类、查看和学习. 用 SQLite 数据库存储这 108 类驱动程序的信息. 图 6 是驱动详细分类模块的数据库设计.

rowid	name	description
1	accessibility	这些驱动提供支持一些辅助设备. 在 Linux 3.9.4 中, 这个文件夹中只有一个驱动就是盲文设备驱动
2	acpi	高级配置和电源接口驱动用来管理电源使用.
3	amba	高级微控制器总线架构是与片上系统 (SoC) 的管理和互连的协议. SoC 是一块包含许多或所有必要的
4	ata	该目录包含 PATA 和 SATA 设备的驱动程序. 串行 ATA (SATA) 是一种连接主机总线适配器到像硬盘
5	atm	异步通信模式 (ATM: Asynchronous Transfer Mode) 是一种通信标准.
6	auxdisplay	这些驱动用于管理液晶显示屏. 液晶显示屏会在按压时显示图标. 注意: 按压会触摸屏幕, 所以请不
7	base	这是个重要的目录包含了固件、系统总线、虚拟化能力等基本的驱动.
8	bcm2	这些驱动用于使用基于 AMBA 协议的总线. AMBA 是由博通公司开发.
9	block	这些驱动提供对块设备的支持. 像软驱、SCSI 磁带、TCP 网络设备等等
10	bluetooth	蓝牙是一种安全的无线个人区域网络标准 (PAN). 蓝牙驱动放在这个文件夹. 它允许系统使用各种蓝
11	bus	这个目录包含了三个驱动. 一个转换 pci 接口协议到 scp 协议. 一个是设备间的互联驱动. 第三个是具
12	cdrom	这个目录包含两个驱动. 一个是 cd-rom, 包括 DVD 和 CDC 的读写. 第二个是 gd-rom (只读光盘). C
13	char	字符设备驱动放在这里. 字符设备每次传输数据时传输一个字节. 这个文件夹里的驱动包括打印机、PS
14	clk	这些驱动用于系统时钟.
15	clocksource	这些驱动用于作为定时器的时钟
16	connector	这些驱动使内核知道当进程 fork 并使用 proc 连接时 (UID/用户ID)、GID(组ID) 和 SID(会话ID).
17	cpufreq	这些驱动改变 CPU 的电源消耗.
18	cpuidle	这些驱动用来管理空闲的 CPU. 一些系统使用多个 CPU. 其中一个驱动可以让这些 CPU 负载相当.
19	crypto	这些驱动提供加密功能.

图 6 驱动详细分类模块的数据库设计

3.2.8 设置模块主要功能及实施机制

设置模块包含计算机信息和系统升级检测两个子模块. 其中升级检测模块, 将系统的新版本放在服务器上, 同时编写一个 JSON 文件, 里面存放新旧版本的信息, 包含版本号、程序下载地址、更新时间版本说明等. Qt 程序通过解析 JSON 文件, 比对版本号, 确定是否有更新, 弹出更新对话框, 用户选择更新与否, 此模块用到了 Qt 技术的网络编程、JSON 解析、信号的自动连接等. 计算机信息模块通过 system() 调用、Linux 管道机制实现对系统信息的获取, 包括计算机名称、本地 IP 和用户名、MAC 地址、CPU 信息、处理器个数、类型及版本等. 图 7 是计算机信息模块设计图.

计算机名称: TextLabel	是否联网: TextLabel
本地 IP 地址和用户名: TextLabel TextLabel	处理器个数: TextLabel
MAC 地址: TextLabel	处理器类型: TextLabel
CPU 信息: TextLabel	处理器版本: TextLabel

图 7 计算机信息展示界面设计图

4 关键技术分析与阐述

4.1 Qt 的信号与槽机制

在系统设计中用到了 Qt 的信号和槽机制、事件机制、GUI 策略、数据处理等技术, 最终完成驱动程序学习系统的设计. 信号和槽机制是 Qt 的核心特征, 可用于两个对象之间的通信. 当某个事件发生时比如点击事件, 会发出一个信号, 当信号发出时, 与之相连的槽函数会自动回调. 图 8 为 Qt 的信号和槽机制.

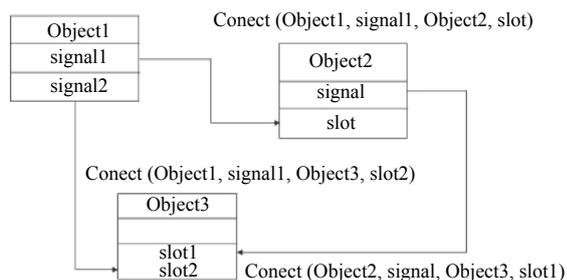


图 8 Qt 的信号和槽机制

信号和槽机制的特点如下:

- (1) 一个信号可以和多个槽相连, 多个信号也可以与一个槽相连;
- (2) 信号也可以连接到信号. 当发出一个信号时, 将触发另一信号;
- (3) 一个类的信号可以和其他类的信号相连.

以下代码是视频学习模块中请求网络读取数据的代码.

```
//新建 QNetworkAccessManager 对象
manager = new QNetworkAccessManager(this);
//关联信号和槽
connect(manager, SIGNAL(finished(QNetworkReply
*)), this, SLOT(replyFinished(QNetworkReply
*)));
void Fit::replyFinished(QNetworkReply *reply)
{
    QString str = reply->readAll(); //读取接收到的
数据
    parseJSON(str);
    reply->deleteLater(); //销毁请求对象
}
```

代码中 connect(manager, SIGNAL(finished(QNetworkReply*)), this, SLOT(replyFinished(QNetworkReply*))) 将信号 finished (QNetworkReply*) 与槽 replyFinished

(QNetworkReply*) 相连, 当触发 finish() 信号是会自动回调 replyFinished 槽.

4.2 Qt 的事件机制

事件是可以被控件识别的操作, 如按下确定按钮、选择某个单选按钮或复选框. Qt 事件是一个 QEvent 对象, 用于描述程序内部或外部发生的动作. Qt 事件产生的类型包括 Qt 自身产生的事件、键盘鼠标事件、绘制事件等, 目前处理 event 最常用的方法是重新实现 paintEvent()、mousePressEvent() 和 keyPressEvent() 等事件处理函数.

下面代码是视频学习模块鼠标点击视频列表选取视频的 mousePressEvent 事件处理函数.

```
void VedioSlider::mousePressEvent(QMouseEvent
*ev)
{
    //应先调用父类的鼠标点击处理事件, 这样可以不
影响拖动的情况
    QSlider::mousePressEvent(ev);
    //获取鼠标的位置, 这里并不能直接从 ev 中取值
    double pos = ev->pos().x() / (double)width();
    setValue(pos * (maximum() - minimum()) +
minimum());
    //发送自定义的鼠标单击信号
    emit costumSliderClicked();
}
```

4.3 QT 多线程技术的应用

在数据库查询以及网络编程模块都用到了 Qt 的多线程技术, 多线程技术可以轻松地开发可移植的多线程 Qt 应用程序, 可以充分利用多处理器的机器, 也可以有效解决在不冻结一个应用程序界面的情况下执行一个耗时操作的问题. Qt 提供了与平台无关的线程类, 在系统中与线程相关的最重要的类是 QThread 类, 该类提供了创建一个新线程以及控制线程运行的各种方法, 它是所有线程类的基础, 该类提供了很多 API 对线程进行操作, 每一个 QThread 对象都代表一个线程, 同时 QThread 类是最常用的 Qt 多线程处理方法. Qt 还提供了 QtConcurrent 模块, QtConcurrent 是高级 API, 该模块中有很多高级函数用于处理一些常见的并行计算模式, 无需使用低级线程原语, 例如: 互斥、读写锁、等待条件或信号量. 在驱动程序学习系统中在查询数据库操作时用到的多线程技术简

单,且并行处理计算量小,故采用 QTread类即可。

如图 9 所示是系统中驱动程序信息查询、操作数据库的多线程编程的流程图。当用户查询某个驱动程序的具体信息时,进入 Qt 数据库入口,加载 SQL 数据库驱动,然连接数据库,之后分有两个分支,一个是 Qt 主线程事件,另一个是数据查询子线程,创建子线程 QTread,重载 QTread: run(), 在 run() 函数中创建 QSqlTableModel 表格模型, 查询数据表, 输出到表格模型, 然后根据用户输入的模块名进行筛选查询某个驱动程序的详细信息, 在主线程中调用子线程, 执行 mythread.start() 函数, 在 Qt 窗口显示。

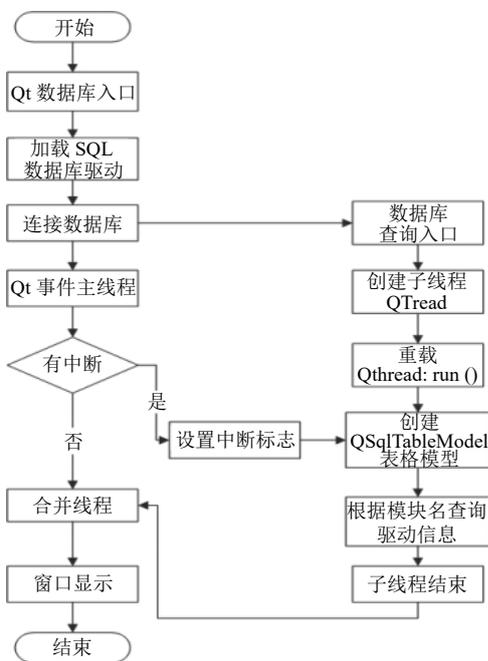


图 9 多线程流程图

4.4 系统运行效果

通过交叉编译方式,使系统运行于 X86、ARM、MIPS 三种平台。当然也运行于 Windows 平台上。图 10 是系统的登录界面,图 11 是系统主界面图。

下面挑选部分功能模块进行结果演示。图 12 是视频学习模块,这个功能模块包含驱动程序开发和案例的多个视频,研发人员和用户可以根据自身因素进行选择学习。

图 13 是具体驱动模块,包含 200 多种常用驱动程序,每种驱动包含驱动程序名称、文件路径、功能描述、注册函数、执行流程等详细信息,用 QSqlTableModel

表格模型、视图显示,输入具体驱动程序名称,就可以查询驱动名称对应的驱动程序的具体信息。



图 10 系统登录界面



图 11 系统主界面

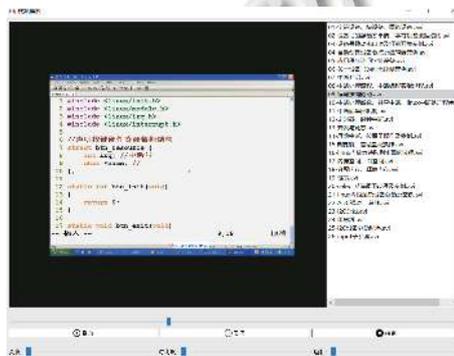


图 12 视频学习模块



图 13 具体驱动信息

图 14 是系统设置模块,此功能模块包含计算机信息、系统更新两个子模块,点击系统更新,就会弹出检查更新提示,若检测到新版本会有提示,用户可以选择去下载或者不更新。下载时通过代码连接远程服务器,更新的版本会放到服务器上。



图 14 系统设置

图 15 是远程文件下载和上传模块,输入服务器地址、用户名、密码登录到 FTP 服务器,用户可以看到远程服务器上可见的目录或文件,然后选择下载。

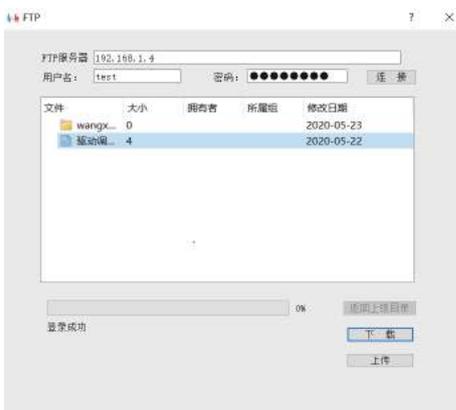


图 15 远程文件下载和上传

5 总结

本文采用 Qt 的进程间通信技术、多线程技术、数据库技术、网络通信技术、多媒体技术、Linux 管道技术等,设计并实现了运行于麒麟操作系统下的内核驱动程序学习系统。在设计与实现中涉及有 Qt 和 C++ 语言、软件算法、Linux 操作系统命令、FTP 通信、TCP 网络通信、计算机硬件系统架构等知识层面。经过测试,系统能够稳定运行,用户端界面效果好。该系统主要用于国产麒麟系统,能体现出其巨大的价值,填补了驱动研究与学习、适配等方面的短板,用于企业将大大减少驱动的维护成本,减少新人的学习时间,降低系统的开发周期等,促进国产 Linux 操作系统的普及与推广。用于高校将有助于学生快速掌握驱动程序学习知识,深入了解驱动程序开发思路,掌握每种驱动程序的执行流程等,同时为国产操作系统储备人才。当然此系统还存在一些潜在的不足,还需继续改进、补充和完善,请批评指正。

参考文献

- 1 Corbet J, Rubini A, Kroah-Hartman G. Linux Device Drivers. 3rd ed. Beijing: O'Reilly, 2005.
- 2 张京林. 国产操作系统产业化现状分析和展望. 科技与创新, 2016, (20): 27-29.
- 3 李勇. 基于麒麟系统的即时通讯系统设计与实现. 自动化技术与应用, 2020, 39(3): 51-55. [doi: 10.3969/j.issn.1003-7241.2020.03.013]
- 4 Passos L, Padilla J, Berger T, et al. Feature scattering in the large: A longitudinal study of Linux kernel device drivers. Proceedings of the 14th International Conference on Modularity. Fort Collins, CO, USA. 2015. 81-92.
- 5 Bai JJ, Wang YP, Hu SM. AutoPA: Automatically generating active driver from original passive driver code. Proceedings of 2018 International Symposium on Code Generation and Optimization. Vienna, Austria. 2018. 288-299.