第35卷 第22期 计 算 机 工 程 2009年11月
Vol.35 No.22 Computer Engineering November 2009
·工程应用技术与实现· 文章编号：1000—3428(2009)22—0242—02 文献标识码：A 中图分类号：TP316.89

基于ARM智能卡的卡片操作系统

黄一平[1]，农丽萍[1]，唐汉雄[2]，苏检德[1]
(1. 广西师范大学物理与电子工程学院，桂林 541004；
2. 广西师范大学网络中心，桂林 541004)

摘 要：研究和分析卡片操作系统(COS)性能要求，提出以ARM智能卡为硬件平台的COS设计思路及方法，论述智能卡STK扩展应用及OTA技术实现原理。该系统利用空中下载技术实现卡片STK菜单的远程更新，结构合理、功能齐全、兼容性好。
关键词：智能卡；卡片操作系统；OTA技术

Card Operating System Based on ARM Smart Card

HUANG Yi-ping[1], Nong Li-ping[1], TANG Han-xiong[2], SU Jian-de[1]
(1. College of Physics and Electronic Engineering, Guangxi Normal University, Guilin 541004;
2. Network Center, Guangxi Normal University, Guilin 541004)

【Abstract】This paper researches and analyzes the performance requirements of Card Operating System(COS), and proposes the ideas and methods of COS based on the hardware platform of ARM smart card. The implementation principles of STK application and OTA technology are discussed. The system utilizes OTA technology to implement STK menu of card remote updating. The structure of this system is reasonable, has complete function and high compatibility. It will be widely applied in the future.
【Key words】 smart card; Card Operating System(COS); Over The Air(OTA) technology

1 概述

目前手机应用已成为互联网应用向移动通信领域渗透的突出代表，是最具前景的电子商务应用之一。据市场研究公司IE Market Research(IEMR)最新发表的研究报告“2008年一季度移动预测：2007－2010年中国市场”表明：到2010年，中国手机用户数量将从2007年的5.40亿增长到7.38亿。从中国信息化开展以来，作为GSM网络个人鉴权模块的SIM卡在中国的需求呈现爆炸式的增长[1]。随着芯片行业的发展以及技术的逐渐成熟，一批新公司先后进入了智能卡领域，使竞争空前激烈，因此，针对智能卡的卡片操作系统(Card Operating System, COS)开发已成为竞争的核心。同时，利用智能卡技术提供服务或进行增值业务推广的行业逐渐增多，使跨行业多应用的智能卡成为智能卡技术的发展方向[2]。最初的COS普遍采用8位51单片机作为其处理核心，由于卡片空间和处理器性能的限制，COS功能单一，因此无法提供更复杂的应用和更可靠的安全性[3]。智能卡的核心是COS，它相当于一台微型计算机，不仅有数据存储功能，而且具有数据安全保护、各种加密运算等功能[4]。

如何将最新的市场需求反映到产品中，如何提供更复杂更可靠的应用，这些都是行业研发人员和市场人员共同关注的焦点。依据ARM智能卡硬件平台实现一款高效、可靠、低代码量的COS，通过对需求进行抽象、提升，以模块化和对象化的思路构建系统框架，以达到最高的抽象化和系统移植的可能。

2 ARM智能卡硬件平台

ARM智能卡芯片是一款接触式的CPU智能IC芯片。CPU采用ARM公司的SC100，该CPU基于32位的ARM7架构设计并采用RISC指令。芯片内置ROM，RAM和Flash作为程序和数据的存储。另外，Flash还作为数据的断电存储。该芯片集成了定时器、中断控制器、系统控制、RSA、DES、安全控制、硬件7816接口等模块，主要针对高端GSM电信卡、3G SIM卡和Java卡应用。其接口符合带触点集成电路卡ISO 7816硬件标准，共8个触点，包括VCC, GND, RST, CLK, IO1, IO2及2个保留接口，具体硬件接口如图1所示。

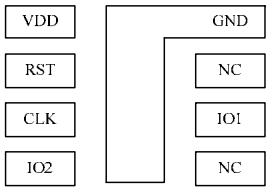

图1 ARM智能卡硬件接口

3 卡片操作系统的设计

COS作为智能卡内软件的核心部分，是智能卡所支持全部应用的基础，设计的首要原则是最大限度满足应用需求。依据GSM机卡接口标准GSM11.11、GSM扩展应用标准GSM11.14以及安全规则，基于ARM智能卡硬件平台的COS不仅要实现与手机通信的功能，而且要能够单独管理自身的SIM卡开发工具包(SIM Tool Kit, STK)应用，扩展空中下载接口，实现动态下载。在设计过程中，采取模块化设计方法，将整个COS分为一般性功能模块、扩展性功能模块、空中下载模块3个部分。

基金项目：广西研究生教育创新计划基金资助项目(2008106020809M264)
作者简介：黄一平(1983－)，男，硕士研究生，主研方向：计算机网络，智能卡系统；农丽萍，硕士研究生；唐汉雄、苏检德，副教授
收稿日期：2009-04-23 **E-mail：**pingarm@qq.com

3.1 COS 的一般性功能

COS 的一般性功能是指对 ISO 7816 硬件接口和 GSM 11.11 通信层和指令协议栈的实现。依据 GSM 11.11 的内容和操作系统的通用模块设计，可以分为 T=0 通信协议模块、SIM 卡文件系统处理模块以及 GSM 11.11 协议栈处理模块等。其中，T=0 协议为通用操作系统的 I/O 模块，主要实现智能卡和手机之间的底层数据通信；智能卡文件系统处理对应于通用操作系统的文件系统；GSM11.11 协议栈中提供的若干指令用于实现智能卡内文件的操作、权限的校验、算法的执行等基本过程。

3.2 COS 的扩展性功能

COS 扩展性功能是指对 GSM11.14 协议栈的实现，是实现 STK 应用的基础。STK 是在原 SIM 卡被动式的操作系统上衍生出的 SIM 卡主动交互式操作系统。它允许智能卡中的应用与支持该应用的手机进行交互操作，为手机的用户接口(User Interface, UI)提供符合 GSM11.14 规范的字节流，手机 UI 把这些字节流解析成能够识别的菜单元素，从而构建一个与手机菜单系统一致的应用菜单。STK 扩展功能主要由 4 个应用协议数据单元(Application Protocol Data Unit, APDU)指令实现，分别为：Terminal Profile, Envelope, Fetch 和 Terminal Response，用于实现 STK 应用和 STK 单元过程。其中，STK 应用相当于一系列的 STK 单元操作，根据逻辑关系和数据传递所构成的连续的一系列 STK 单元。一个 STK 单元过程就是从卡片对某指令回复状态字 91 XX 指令、引发终端执行 Fetch 指令获得 STK 数据执行、并将执行结果以 Terminal Response 指令返回的过程。这个过程可以是 ARM 智能卡提供一套菜单添加到手机原有的菜单结构中(SET UP MENU)，提供一套菜单项目列表供用户选择(SELECT ITEM)，或在显示屏上显示文本或图标(DISPLAY TEXT)等，这些应用的组合可以扩展出丰富的 STK 应用。Terminal Profile 指令和 Envelope 指令则保证了 STK 过程的数据下发和触发源。STK 应用操作流程如图 2 所示。

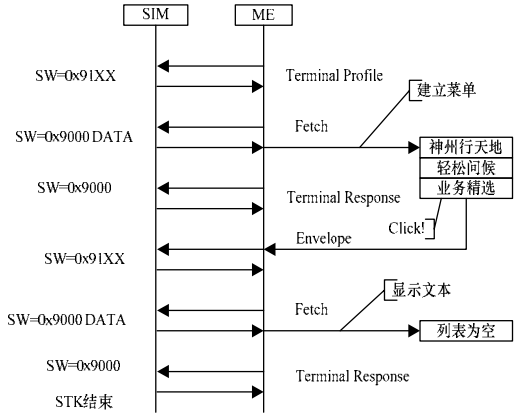

图 2　STK 应用过程示例

STK 应用的各个 STK 单元过程是逻辑连续的，但是执行过程分散在几个指令中。在回复当前指令、终端 Fetch 数据、用户做出选择这个过程中实际上跨越了几个 STK 指令，而且在这几条 STK 指令执行过程中，终端可能会处理一些 GSM11.11 的普通指令，也就是说虽然这段逻辑处在主逻辑下，但在回复 91 XX 状态字节过程中需要重新进入主逻辑。

为了实现 STK 应用单元过程的切换以实现整个 STK 应用的连续，在设计 COS 中考虑了操作系统任务调度的设计思想，依据 Java 虚拟机的方法，采用解释型语言构建 STK 应用，这是系统的核心。Java 虚拟机屏蔽了与具体操作系统平台相关的信息，使 STK 应用菜单只需生成在 Java 虚拟机上运行的目标代码(字节码)，就可以在 COS 上不加修改地运行。Java 虚拟机在执行字节码时实际上是把字节码解释成智能卡内部 COS 上的机器指令来执行的。

每个 STK 任务由若干解释型语言 Bytecode 组成，这些 Bytecode 有的实现具体的 STK 单元，有的控制 STK 单元之间的逻辑，主要分为流程控制和参数控制。为了管理协议栈中的 STK 过程，虚拟机构建于协议栈层次之上，本身是一个逻辑上独立的任务，通过解析存储在数据区的 Bytecode 实现 STK 应用的过程。Java 虚拟机提供一套统一的机制管理 STK 单元指令间的上下文处理、多个任务间基于优先级的任务调度。将 STK 的通用过程归纳为若干 Bytecode，Bytecode 作为数据区的一部分而不是代码区，而程序部分的 Java 虚拟机只需要按规定逻辑将字节码解释执行。这样在统一协议后，一个应用的 Bytecode 可以直接在另一个过程或编译器中引用，同时，Bytecode 作为数据对象又使 STK 应用的动态下载成为可能。

3.3 OTA 空中下载

OTA(Over The Air)通过空中下载机制对 STK 菜单应用进行操作。由于 STK 指令单元的数据以字节码进行存储，在 COS 中只作为数据区，不作为代码区，因此在 Java 虚拟机中，一个 STK 任务可以理解为存储在某个存储区中的一组连续的标准字节码。OTA 菜单下载的实现借助于 STK 功能和数据短信息通道，支持空中下载的 COS 提供可行的人机接口界面供用户发起下载申请，OTA 应用下载服务器根据用户请求，以数据短信息的形式将相应的服务下载内容发给用户手机，并透明地传递给 ARM 智能卡芯片，COS 对下载的短信息内容通过双向认证后进行解析，把 STK 菜单的字节码数据流重新进行组织存储，实现相应的 STK 卡菜单管理。OTA 提供基于数据短信的方式动态下载和管理字节码文件的机制，从而可以动态地实现对 ARM 智能卡内 STK 应用菜单的下载、删除、同步等基本操作。为了与 OTA 服务器接口相结合，在 COS 中需按照中国移动推出的 OTA2 和 OTA3 标准进行字节码函数接口设计。

3.4 COS 整体设计实现

COS 设计从模块上分为硬件接口层、存储字节码的文件系统、GSM 协议栈、Java 虚拟机、OTA 动态下载。从功能的角度，前 3 层对应于 COS 的一般性功能要求，后 2 层对应于 STK 的扩展性功能需求。对于 COS 主流程，系统主要以接收 APDU 指令、处理指令、发送指令处理结果的状态字为基本循环。COS 主流程如图 3 所示。

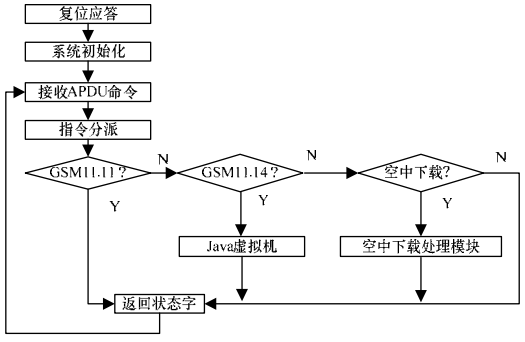

图 3　COS 主流程　(下转第 247 页)