重庆大学本科学生毕业设计（论文）

基于ARM的EPOS操作系统的移植



学 生：冯志敏

学 号：20121955

指导教师：洪明坚

专 业：软件工程专业

重庆大学软件学院

二O一六年六月

**Graduation Design (Thesis) of Chongqing University**

**Transplantation of EPOS operating system based on ARM**



**Undergraduate: Feng Zhimin**

**Instructor: Associated Prof.Hong Mingjian**

**Major: Software Engineering**

**School of Software Engineering**

**Chongqing University**

**June 2016**

# **摘 要**

随着开源硬件在嵌入式领域的广泛使用，越来越多基于ARM体系结构的开发板产品也随之诞生，其中比较流行的一款开源硬件—Raspberry PI(树莓派)，其体积仅信用卡大小，搭载ARM架构处理器，运算性能和智能手机相仿。本文将EPOS操作移植到Raspberry PI开发板上。

本文是将X86体系结构下的EPOS操作系统移植到以Broadcom BCM2835 ARMv6 为处理器的开源硬件Raspberry PI B+上，主要是将EPOS由X86系统结构修改成ARM体系结构，大致修改内容包括引导项(Bootloader)、内存管理单元(MMU)、中断处理、系统定时器、多线程的切换以及PI开发板上的UART驱动。

本文实现的功能为：将EPOS操作系统编译后生成kernel.img文件，将kernel.img文件拷贝到FAT文件格式的SD卡中，将SD卡插入Raspberry PI B+卡槽内，开启电源后，通过UART串口向PC机发送一条消息，同时初始化多任务。

关键词：嵌入式， Bootloader，UART，MMU，Raspberry PI B+

# ABSTRACT

With the widely use of the open source hardware in the embedded field, more and more based on ARM architecture development board products emerge, which more popular a open source hardware - Raspberry PI (raspberry pi), the volume of only credit card size, equipped with ARM processor architecture, the computational performance and smart phones similar。In this paper, the EPOS operation is transplanted to the PI Raspberry development board.

In this paper, the x86 architecture of EPOS operating system transplantation by Broadcom BCM2835 armv6 processors of open source hardware raspberry PI B +, mainly is the EPOS by x86 system structure modified to arm architecture, roughly modified content includes a guide (bootloader), memory management unit (MMU), interrupt processing, system timer, multi thread switching and PI development board UART driver.

In this paper, the realization of the function is: EPOS operating system compiler generated after kernel.img file, kernel.img files are copied to the SD card fat file format in, SD card into the raspberry PI B + card slot, after power is turned on, through the UART serial port to the PC sends a message. Meanwhile, the initial of multi task.

**Key words：**Embedded, Bootloader, UART, MMU, Raspberry PI B+

# 目录

[摘要 I](#_Toc358850539)

[ABSTRACT II](#_Toc358850540)

[目录 III](#_Toc358850541)

[1 绪论 1](#_Toc358850542)

[1.1 选题背景及意义 1](#_Toc358850543)

[1.2 国内外研究现状 2](#_Toc358850544)

[1.3 开发板介绍 3](#_Toc358850545)

[1.4 开发环境搭建 3](#_Toc358850545)

[2 系统关键技术 4](#_Toc358850546)

[2.1 ARM体系结构介绍 4](#_Toc358850547)

[2.2 Bootloader设计 4](#_Toc358850548)

[2.2.1 Raspberry PI启动流程介绍 4](#_Toc358850549)

[2.2.2 EPOS初始化以及加载kernel 6](#_Toc358850550)

[2.3 ARM虚拟内存 7](#_Toc358850551)

[2.3.1 MMU工作原理 7](#_Toc358850552)

[2.3.2 CP15协处理器 8](#_Toc358850553)

[2.3.3 EPOS恒等映射 8](#_Toc358850553)

[2.3.4 EPOS pagefault处理 8](#_Toc358850553)

[2.4 ARM中断处理 7](#_Toc358850551)

[2.4.1 中断机制 7](#_Toc358850552)

[2.4.2 中断向量表 8](#_Toc358850553)

[2.4.3 Raspberry PI定时器 8](#_Toc358850553)

[2.5 多任务处理 7](#_Toc358850551)

[2.5.1 多任务工作原理 7](#_Toc358850552)

[2.5.2 多任务切换 8](#_Toc358850553)

[3 模块详细设计 13](#_Toc358850557)

[3.1 系统引导及其初始化 13](#_Toc358850558)

[3.2 UART串口驱动设计 13](#_Toc358850559)

[3.3 虚拟内存映射 14](#_Toc358850560)

[3.3.1 内存映射 14](#_Toc358850561)

[3.3.2 激活MMU 14](#_Toc358850562)

[3.3.3 EPOS恒等映射 16](#_Toc358850563)

[3.4 EPOS中断处理 14](#_Toc358850560)

[3.4.1 初始化中断向量表 14](#_Toc358850561)

[3.4.2 中断现场保护 14](#_Toc358850562)

[3.3.3 定时器中断处理 16](#_Toc358850563)

[3.5 多任务 14](#_Toc358850560)

[3.5.1 创建多任务 14](#_Toc358850561)

[3.5.2 任务切换 14](#_Toc358850562)

[4 效果展示 21](#_Toc358850566)

[4.1 源码编译操作 21](#_Toc358850567)

[4.2 系统运行结果 21](#_Toc358850568)

[致谢 40](#_Toc358850584)

[参考文献 41](#_Toc358850585)

# **绪论**

## 选题背景及意义

多任务操作系统内核是源于Linux操作系统，并进行简化，使用户更容易理

## 国内外研究现状

在国内，国防科技大学计算机学院着手研究Unix操作系统，并把银河巨型

## 开发板介绍

## 开发环境搭建

# **系统关键技术**

1. ARM体系结构介绍
2. Bootloader设计
3. Raspberry PI启动流程介绍
4. EPOS初始化以及加载kernel
5. ARM虚拟内存
6. MMU工作原理
7. CP15协处理器
8. EPOS恒等映射
9. EPOS pagefault处理
10. ARM中断处理
11. 中断机制
12. 中断向量表
13. Raspberry PI 定时器
14. 多任务处理
15. 多任务工作原理
16. 多任务切换

# **模块详细设计**

1. 系统引导及其初始化
2. UART串口驱动设计
3. 虚拟内存映射
4. 内存映射
5. 激活MMU
6. EPOS恒等映射
7. 虚拟内存映射
8. 初始化中断向量表
9. 中断现场保护
10. 定时器中断处理
11. EPOS中断处理
12. 创建多任务
13. 任务切换

# **效果展示**

1. 源码编译操作
2. 系统运行结果