

行政院國家科學委員會專題研究計畫 成果報告

子計畫一：應用於系統單晶片嵌入式 LINUX 作業系統之實作

計畫類別：整合型計畫

計畫編號：NSC91-2218-E-009-002-

執行期間：91年08月01日至92年07月31日

執行單位：國立交通大學資訊科學學系

計畫主持人：張瑞川

報告類型：完整報告

處理方式：本計畫可公開查詢

中 華 民 國 92 年 10 月 20 日

應用於系統單晶片嵌入式 Linux 作業系統之實作

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 91-2218-E-009-002

執行期間： 89 年 8 月 1 日至 92 年 7 月 31 日

計畫主持人：張瑞川

共同主持人：

計畫參與人員：黃昭維、林衍良、蔡建東、宋孟霖、莊豐穗

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、

列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立交通大學資訊科學系

中 華 民 國 92 年 10 月 6 日

摘要

隨著系統單晶片技術的發展，近年來嵌入式作業系統的相關研究受到相當程度的重視。本計劃將以功能強大，高穩定度的 Linux 作業系統為基礎，發展適用於系統單晶片嵌入式 Linux 作業系統。我們研究及實作以唯讀記憶體為基礎的 Linux 核心，在原地執行功能，即時作業支援，Flash 檔案系統以及系統單晶片電源管理等系統功能。我們也將重組 Linux 成為模組化核心以適應不同系統單晶片之系統需求。

關鍵詞：Linux；嵌入式系統

Abstract

With the development of System on Chip (SoC) technology, researches on embedded, real-time systems has received lots of attention. Based on the power, reliability of Linux, the goal of this project is to develop an Linux system that is suitable for SoC. We plan to research and implement a ROM-based Linux kernel, Execution-In-Place(XIP) function, real-time support, Flash memory file system, and power management for SoC. We also plan to modulate Linux kernel so that it can fit various requirements of different SoC.

Keywords: Linux; Embedded System

目錄

中文摘要.....	II
Abstract.....	III
計畫之背景及目的.....	1
研究方法.....	5
結果與討論.....	7
參考文獻.....	11
計畫成果自評.....	20

計畫之背景及目的：

Linux 是目前世界上最受到重視的作業系統。著名的軟體公司如 IBM , SGI , Informix , Oracle , Corel 等均將其重要的軟體移植到 Linux 上。此外, Linux 的成長率也遠超過 Microsoft 的 Windows/NT。更由於 Linux 是 Open Source 政策下的作業系統, 透過網際網路有成千上萬的工程師同時改進其品質及增強功能, 使得其前景大為看好, 有與 Microsoft 的 Windows/NT 分庭抗禮的實力。

然而, 在 embedded 應用方面, Linux 僅支援少數以 VME BUS 為基礎的應用系統, 並未有解決方案提供完整的支援。尤其對系統單晶片嵌入式系統方面的應用更需重新設計 Linux 核心架構, 以適應不同 SoC 所需的作業系統核心 IP。同時即時作業支援(real-time system support)對 SoC 嵌入式系統的應用亦相當重要。目前雖已有幾所國外大學在研究如何在 Linux 作業系統上加入即時作業能力, 但是其研究並未完全考量嵌入式系統的需求。

交通大學資訊科學系系統實驗室多年來的研究重點是微核心作業系統(Micro Kernel Operating Systems)。我們以 Mach 微核心作業系統為主要研究工具, 在國科會長期的支持下, 我們對微核心作業系統有相當深入的研究, 並曾將 Mach 修改移植至不同的硬體架構, 如 Convex 超級電腦、Acer Frame 3000 MP、和以高速網路 Myrinet 為基礎的叢集系統, 並對 Mach 核心程式和虛擬記憶體系統作相當程度的研究和改進。但是微核心架構的優點如可模擬多種作業系統功能(multiple personality)、可延展性(scalability)、和可支援多種計算機架構等一直沒有完全實現。隨著 IBM 公司在 1996 年底, 放棄以 Mach 微核心作業系統整合其多個作業系統產品線, 和 Microsoft 為了增進 NT 系統效能, 漸漸改變 NT 作業系統的架構, 使得什麼是新一代作業系統核心架構一直對此領域的研究人員廣泛

的討論著。除了微核心架構一再被討論和改進外，可擴充式作業系統架構(extensible kernel architecture)和物件導向式系統核心架構等亦被提出。

在應用程式的開發步驟裡，利用軟體元件(componentware)幾乎已成為程式設計不可或缺的一環。使用軟體元件不但可使程式開發更為簡單、軟體產能更為提高，同時也達成了軟體的再使用性(reuse)，例如微軟公司的 VBX,OLE,COM,DCOM 及 Active X, Sun Micro 的 Java Bean 等技術，對應用軟體的開發有關鍵性的影響力。

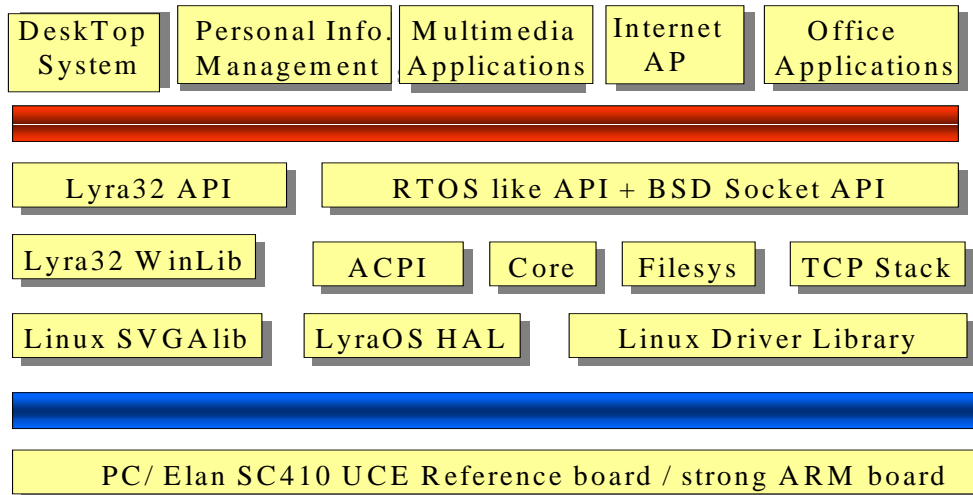
但是作業系統核心程式架構和發展方法並沒有太大的進展，相關程式開發和除錯工具和 20 年前相比仍然沒有太大的改變。所以作業系統的維護和開發新功能的成本仍然很高。

所以在系統軟體方面，如何能利用軟體元件的觀念，以加速即時核心程式的研發，及降低後續維護成本，是學術界研究作業系統發展重要的研究方向。

我們在國科會前期計劃支持下探討新的即時核心程式架構，並製作出完全模組化的 Lyra 即時作業系統和發展環境，整個系統包括模組化的核心程式、輸出系統驅動程式模組、檔案系統、通訊協定程式，支援 Java API 的 virtual machine，和以瀏覽器(Browser)為基礎的桌面系統等，如圖一。

Lyra 即時作業系統雖可做為研究作業系統架構和效能測試的研究平台，但是受限人力、物力無法完成完整的 API，以及沒有完整的程式發展系統，所以無法完全因應 SoC 各種不同嵌入式應用領域的需求。

本計劃預計以三年的時間，配合總計劃以及各子計劃 IP 的研發，在保留 Linux 作業系統 API 以支援各項多媒體應用程式開發的狀況下，重新探討 Linux 核心程式架構。我們將利用軟體元件(componentware)的觀念重新設計具即時作業功能及支援嵌入式作業方式的模組化 Linux 核心程式。



圖一

國內外有關本計劃之研究情形：

國內外許多大學和研究機構均以從事關於即時核心程式及相關發展環境的研究。在國內探討核心程式架構或實作作業系統的包括台灣大學、成功大學、交通大學、中研院資訊所和資訊工業策進會。在國外從 1994 年開始，關於核心架構的探討開始受到包括 ACM SIGOPS 及 USENIX 兩個 Communities 的重視，這其中已經開始著眼於 Componentized 的作業系統只有法國 Chorus/ClassiX 而著眼於 Code-reuse 的只有美國猶他大學的 Fluk OSKit 計劃，他們也只達到使核心程式交易於使用，並移植了一些共享系統軟體的目的而已，尚未能夠做到易於維護，修改甚至於開發新元件。

卡內基美濃大學的 Resource-centric 核心計劃則探討到即時資源的保留技術但所使用的微核心架構的 RT-Mach。哈佛大學的 VINO 核心計劃，則是在核心程式中加入了 Transaction 的觀念，希望系統服務可以藉由動態連結技術，加入核心程式中，而 Transaction 的目的就是確保在外加執行碼到核心程式中時，系統仍能安全運作。VINO 的最大困難就是系統效能，在加入了 Transaction 後，核心程式速度大量降低，而影響到整

體效能。

華盛頓大學的 SPIN 核心計劃和 VINO 一樣，希望簡易的核心程式是來自於動態連結的外加程式執行碼。由於使用者程式可以輕易外加核心執行碼，核心程式可以高度符合使用者程式需求，而整體效能亦可提升。個人認為這是一個很好的方法，用來產生一個易於替換，修改的核心程式，且用於應用程式的軟體元件技術，說不定可以很簡單的搬到核心程式中。但是 SPIN 並沒有真正區分、制定系統軟體元件，如何把環環相扣的各系統模組確實模組化是一個很重要的研究問題。此外，MIT 的 Exokernel 核心計劃，則是將系統軟體模組便成為應用程式程式庫一部分，目的是減少系統及使用者程式間的控制變換次數，以提高效能。在觸及如何提高軟體再用性，Exokernel 核心的系統架構並沒有解決任何問題，而諸如 UIUC 大學的 Choice 核心程式計劃，是引進物件導向的技術到核心程式中。個人認為純物件導向的核心架構，並不太可行。例如繼承只會增加核心程式的複雜度，無助於核心程式的模組化。

許多現有的商用系統如 PSOS⁺，LynxOS，VTRX，Chorus 等嵌入式作業系統亦是我們參考研究的對象。但是商用作業系統的原始程式並不公開，無法掌握其核心技術。最近亦有國內外的資訊業者，如 Lineo 及資策會，正在進行將 Linux 改成嵌入式，但是他們大都僅小幅更改 Boot 部分而未重新設計 Linux 核心。

研究方法：

首先我們將研究作業系統架構的新技術尤其是用於可延展性核心 (extensible kernel) 及現有嵌入式即時作業系統的核心程式。我們也將研究軟體 components 和 composition 的機制並探討用於建構作業系統核心的架構。

同時我們亦將研究 Linux 核心原始程式，以及現有增加 Linux 即時處理能力的方法和架構，以界定將 Linux 移植至 Embedded 平台所需更動的部分。

根據初步研究，預期需要修改的子系統包括：

(1) Linux Loader (LILO)

Linux Loader 目前的工作模式是從磁碟式硬碟載入 Linux 核心，我們將修改 LILO 以配合未來 SoC 從 flash 或 ROM 上載入核心程式的需求。

(2) 初始子系統 (Initialization Subsystem)

Linux 的初始子系統需要作硬碟讀寫動作，我們將配合 SoC 要求在保留原功能的原則下修改這部分原始碼。

(3) 記憶體管理子系統 (Memory Management)

Linux 的記憶體管理系統支援虛擬記憶體及整合 I/O buffer cache 等先進功能，但是對 SoC 嵌入式系統的應用，記憶體管理子系統需要作大幅度的修改，以配合 CPU 有時並不支援硬體記憶體管理功能及硬碟作備援等功能。我們已將增加在原地執行 (Execution-In-Place) 的功能。

(4) 核心其它部分，包括檔案系統，輸出入系統等亦將配合修改。

以上研究工作將於第一年完成，預計將可在 Evaluation board 上執行 Embedded Linux 基本功能。

第二階段計劃，我們將在保持 Linux API 的狀況下重新建構適合 SoC

嵌入式整合系統的核心，我們希望利用第一年的研究成果及前期實作 LyraOS 的經驗，從新設計 Linux 核心程式，以提供可替換的系統模組。預計我們將 Linux 核心分割成開機模組，工作元管理模組，即時排成管理模組，插段(interrupt)模組，工作元間通訊模組，記憶體管理模組，通訊協定模組以及檔案系統模組。同時我們考量嵌入式系統的特殊需求和少量的記憶體，固定的工作天及檔案系統模組。同時我們考量嵌入式系統的特殊需求如少量的記憶體，固定的工作元組及電源管理等。

第三階段計劃，我們將著重在 SoC 系統偵錯，測試的作業系統支援，同時我們也將配合總計劃和其它子計劃作系統整合與驗證測試。

結果與討論：

在第一年度的計劃中，我們的研究成果如下：

- (1) Embedded Linux Loader & Initialization System
- (2) Embedded Linux Memory Management System
- (3) Embedded Linux File System
- (4) Execution-In-Place Function
- (5) Port Embedded System to Evaluation Board

茲將其研究成果分述如下：

➤ Embedded Linux Loader & Initialization System

一般硬體平台上系統的啟動有以下程序：

- 硬體啟動 power on
- 中央處理器擷取 0x0 位址處的程式碼開始執行
- Loader 將 Linux 核心從磁碟式硬碟讀到記憶體
- Loader 將程式執行權轉交給 Linux
- Linux 進行系統初始化動作並進入 kernel task

我們將 Embedded Linux 上的 loader 作適度修改，另其可以由 flash 或是 rom 來載入 Linux 核心而不限定只能由磁碟式硬碟上載入，而系統初始化的部分也根據 SOC target platform 做相關的修改。

➤ Embedded Linux Memory Management System

為了因應嵌入式系統上普遍記憶體受限的環境，我們採用最保守的 memory pool 管理方式。我們在編譯時期讓使用者指定要建立 memory pool 的 chunk/block size 以及數量，讓使用者可以針對應用程式對記憶體的利用特性加以指定最合適的 memory pool。我們發現對大多數應用程式而言，這樣的動態記憶體配置方式以足敷所需，且執行效率亦有不錯的表現。

➤ Embedded Linux File System

雖然 Linux 本身對檔案系統的支援相當廣泛，從本身的 ext2，到 DOS 上的 FAT，Win32 系統上的 VFAT，以及 NTFS 等各主要作業系統上的檔案系統均提供支援。為了符合嵌入式系統的使用環境，我們將檔案系統作相當幅度的簡化，使其符合系統的儲存環境。我們以類似 ramdisk 的方式在記憶體中模擬 compact flash 或其他型態記憶體在儲存時的應用情況。

➤ Execution-In-Place Function

配合執行碼儲存系統的設計，我們使程式能夠在其儲存的位置上直接執行。就系統內部而言，程式碼可直接擷取自線性定址空間，因此此執行方式省下了傳統計算架構中將程式碼影像檔載入記憶體的步驟與空間。

➤ Port Embedded Linux to Evaluation Board

接下來最後的工作就是將我們發展好的 Embedded Linux port 到 Evaluation Board 上以測試我們的 kernel 是否正常 work，我們所使用的 Evaluation 為 StrongArm SA-1110 的 Assabet，再依我們的 Evaluation Board 的硬體做相關的硬體驅動程式，如 LCD controller, video driver, pcmcia driver 等等。並實際傳入 ramdisk 在 Evaluation 上執行應用程式，同時測試網路功能也可正常 work，這就是我們最後的工作。

在第二年度的計劃中，我們的研究成果如下：

- (1) Linux 核心程式元件化
- (2) Linux 電源管理系統
- (3) Linux 即時功能

茲將其研究成果分述如下：

➤ Linux 核心程式元件化

雖然在應用程式的開發步驟裡，利用軟體元件幾乎已成為程式設計不可或缺的一環。但核心程式方面，如何能利用軟體元件的觀念，以加速即時核心程式的研發，一直是作業系統發展重要的研究方向。我們利用前幾年開發元件化系統核心的經驗，將 Linux 原先的各子系統之間的介面釐清。並進一步將各子系統包成模組元件。每一元件提供若干服務點(service points)。元件間就經由這些服務點來溝通。到目前為止，我們已經釐清的元件介面包含：初始化，記憶體管理，檔案系統管理，及執行緒管理。

➤ Linux 電源管理系統

因為嵌入式設備大多用電池當作其電力來源，所以嵌入式系統設計的一個重要原則就是要省電。目前很多硬體都有提供省電的機制，例如：CPU, memory, disk, monitor 等等。但卻沒有核心程式去很有效的利用這些硬體特性來讓整個系統的耗電量降至最低。國內外對於電源管理相關的研究主要分成兩部分：動態電源管理(Dynamic Power Management)及動態電壓調整(Dynamic Voltage Scaling)兩類。前者是利用硬體的省電模式特性。當系統發現暫時用不到某硬體時，就將其切換到省電模式。如此一來就可減少耗電量。後者主要是探討 CPU 的省電。CPU 提供多種工作電壓。工作電壓低時雖然效能較差，但由於電源消耗與電壓平方成正比，所以降低電壓對於省電很有幫助。在我們的 Linux 系統中，我們以前者為主。因為我們最主要的介面是 Wireless LAN 的介面，所以我們與 Wireless LAN MAC 層有一個 Power Save 介面。藉由此介面，我們可以使 Wireless LAN 的硬體進入省電模式。同時，我們也要在系統整合時調整我們使用 CPU 的頻率，藉以讓整個系統可已有較低的耗電量。

➤ Linux 即時功能

由於 Linux 原始的設計是針對一般應用，所以它並不適合支援即時的應用程式。於是國外有一些研究就開始改進 Linux 以達到支援即時的應用程式的目的。

的。例如：KURT, RTLinux, RTAI 及 TimeSys Linux/RT 等等。在我們的計畫中，我們已經分析了 RTLinux 的架構及其優點。接著我們評估 RTLinux 中哪些功能是我们的 Linux 核心程式所需要的，再將這些部分移植到我們以 Evaluation board 為基礎的 Linux 上。

在第三年度的計畫中，我們的研究成果是整合 Linux 與 WLAN MAC。

在第三年度的計畫中，我們著重於各子計畫的 IP 整合。我們以 FPGA 板為 SOC 平台的雛形，實際驗證各個子計畫 IP 的功能正確性與效能。在所有子計畫中，我們這個嵌入式 Linux 的子計畫要與 WLAN IP 整合。到目前為止，WLAN 的 MAC 及 Baseband 部分是實作在 FPGA 上。我們使用的 FPGA 是 ARM Integrator 的 Logic Module。所以我們可用 ARM Integrator AP 為平台，做為軟體 (Core Module)與硬體 (Logic Module)間溝通的平台。在軟體方面，主要是要用 MAC 所提供的暫存器介面 (Register Interface) 來實作 WLAN 驅動程式。我們的驅動程式架構在 PCMCIA 上，並以由 CPU 對資料暫存器 (data register) 做讀寫來收送資料。到目前為止，此驅動程式已經實作完成。我們並已整合此驅動程式到我們所移植的核心程式中，與 FPGA 上的 WLAN MAC/Baseband 作 peer-to-peer 的測試。

除此之外，我們也負責與 MPEG 子計畫的整合。將 MPEG 子計畫的 IP 與軟體整合到 Linux 環境中以驗證整體功能及效能。透過整合，我們已經可以在 ARM Integrator 上透過 WLAN MAC/Baseband 作 on-line peer-to-peer 的 MPEG 檔傳輸與播放。

參考文獻：

1. S G. Abraham and S. A. Mahlke, "Automatic and Efficient Evaluation of Memory Hierarchies for Embedded Systems," Proc. 32th Annual ACM/IEEE Int'l symposium Microarchitecture, pp. 114-125. 1999.
2. M. Accetta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian and M. Young, "Mach: A New Kernel Foundation for UNIX Development," Proc. of the Summer 1986 USENIX Conference. pp. 93-112, June 1986.
3. David P. Anderson, "Device Reservation in Audio/Video Editing Systems," ACM Transactions on Computer Systems, Vol. 15, No. 2, May 1997, pp. 111-133.
4. M. Auslander, H. Franke, B. Gamsa, O. Krieger, and M. Stumm, "Customization Lite," Proc. 16th ACM Symposium on Operating Systems Principles, Saint-Malo, France, October 1997.
5. J. M. Bass, "Proposals Toward an Integrated Design Environment for Complex Embedded Systems," Proc. Sixth Euromicro Workshop Parallel and Distributed Processing, pp. 273-278, 1998.
6. B. N. Bershad et al., "Eggers. Extensibility, Safety, and Performance in the SPIN Operating System," Proc.15th ACM Symposium on Operating Systems Principles, Copper Mountain, CO, pp. 267-284. December 1995.
7. D. Beuche et al., "The PURE Family of Object-Oriented Operating Systems for Deeply Embedded Systems," Proc. IEEE Object-Oriented Real-time Distributed Computing, pp. 45-53, 1999.
8. S. H. Bokhari and R. Rehman, "Linux and the Developing World," IEEE Software, Vol. 16, No. 1, pp. 58-64, Jan.-Feb. 1999.
9. S. N. Bokhari, "The Linux Operating System," IEEE Computer, Vol. 28, NO. 8, pp. 74-79, Aug. 1995.
10. T. Bollinger, "Linux in Practice: an Overview of Applications," IEEE Software, Vol. 16, No. 1, pp. 72-79, Jan.-Feb.1999.

11. T. Bollinger and P. H. Beckman, "Linux on the Move," IEEE Software, Vol. 16, No. 1, pp. 30-35, Jan.-Feb.1999.
12. I. T. Bowman, R. C. Holt, and N. V. Brewster, "Linux as a Case Study: its Extracted Software Architecture," Proc. Int'l Conf. Software Engineering , pp. 555-563, 1999.
13. R. Campbell, N. Islam, P. Madany, and D. Raila, "Designing and Implementing Choices: An Object-Oriented System in C++," Communications of the ACM, Sep. 1993.
14. Chorus Corporation. CHORUS/ClassiX Release 3 Technical Overview. CS/TR-96-119.13. June 1997.
15. A. G. Dean and J. P. Shen, "Techniques for Software Thread Integration in Real-time Rmbedded Systems", Proc. IEEE Real-Time Systems Symposium, pp. 322-333, 1998.
16. P. Druschel, V. S. Pai, and W. Zwaenepoel, "Extensible Systems are Leading OS Research Astray," Proc. 16th ACM Symposium on Operating Systems Principles, Saint-Malo, France, October 1997.
17. D. R. Eagler, M. F. Kaashoek, and J. O'Toole Jr, "Exokenrel: An Operating System Architecture for Application-Level Resource Management," Proc.15th ACM Symposium on Operating Systems Principles, Copper Mountain, CO, pp. 251-266. December 1995.
18. Hugo Fierz, "The CIP Method: Component- and Model-based Construction of Embedded Systems," Proc. 7th ACM SIGSOFT Symposuim on Foundations of Software Engineering, pp. 375-392, 1999.
19. Brett D. Fleisch, "The Failure of Personalities to Generalize," Proc. 6th Workshop on Hot Topics in Operating Systems, Cap Cod, Massachusetts, May 1997.
20. J. Fleischmann and K. Buchenrieder, "Prototyping Networked Embedded

- Systems,” IEEE Computer, Vol. 32, No. 2, pp. 116-119, Feb. 1999.
21. Bryan Ford, Godmar Back, Greg Benson, Jay Lepreau, Albert Lin and Olin Shivers, “The Fluk OSKit: A Substrate for Kernel and Language Research,” Proc.16th ACM Symposium on Operating Systems Principles, Saint-Malo, France, October 1997.
 22. A. M. Fox, J. E. Cooling, and N. S. Colling, “Integrated Design Approach for Real-time Embedded System,” Proc. IEE Software, Vol. 146, No. 2, pp. 75-85, April, 1999.
 23. P. Green, D. Morris, and G. Evans, “Software Technology for Embedded Systems,” Proc. Eighth IEEE Int’l Workshop Software Technology and Engineering Practice, pp. 402-410, 1997.
 24. Robert Geist and James Westall, “Bringing the High End to the Low End: High Performance Device Drivers of the Linux PC,” Proc. 36th Annual Conf. On Southeast Regional Conference, pp. 251-260, 1998.
 25. S. Goel and D. Duchamp, “Linux Device Driver Emulation in Mach,” Proc. Annual USENIX Technical Conference, pp. 65-73, San Diego, California, January 1996.
 26. A. Gokhale and D. C. Schmidt, “Techniques for Optimizing CORBA Middleware for Distributed Embedded Systems,” Proc. IEEE INFOCOM, Vol 2, pp. 513-521, 1999.
 27. D. Grunwald, “Embedded, General-purpose, and High-performance Systems,” ACM Comput. Surv. Vol. 28, No. 4, Dec. 1996.
 28. J. Hallen, A. Hammarqvist, F. Juhlin, and A. Chrigstrom, “Linux at the Workplace,” IEEE Software, Vol. 16, No. 1, pp. 52-57, Jan.-Feb. 1999.
 29. A. Harbert, W. Lively, and S. Sheppard, “A Graphical Specification System for User-Interface Design,” Vol. 74, pp. 12-20. July 1990.

30. D. Hildebrand, "An Architecture Overview of QNX," Proc. Usenix Workshop on Micro-Kernels and Other Kernel Architectures, pp. 113-126, 1992.
31. S. Hustin, M. Potkonjak, Eric Verhulst, and Wayne Wolf, "Real-time Operating Systems for Embedded Computing," Proc. IEEE/ACM Int'l Conf. on Computer-aided Design, 1998.
32. D. Jensen, "Adventures in Embedded Development," IEEE Software, Vol. 116, pp. 116-118, Nov. 1994.
33. R. Jha et al., "Adaptive Resource Allocation for Embedded Parallel Applications," Proc. Int'l Conf. High Performance Computing, pp. 425-432, 1996.
34. M. B. Jones, D. Rosu, and Marcel-Catalin Rosu, "CPU Reservations and Time Constraints: Efficient, Predictable Scheduling of Independent Activities," Proc. 16th ACM Symposium on Operating Systems Principles, Saint-Malo, France, October 1997.
35. A. Kalavade and P. Moghé, "AsaP--a Framework for Evaluating Run-time Schedulers in Embedded Multimedia End-systems," Proc. 6th ACM Int'l Conf. Multimedia, pp. 291-296, 1998.
36. Orran Krieger, "HFS: A Performance-Oriented Flexible File System Based on Building-Block Compositions," ACM Transactions on Computer Systems, Vol. 15, No. 3, pp. 286-321, August 1997.
37. G. Lamastra et al., "HARTIK 3.0: a Portable System for Developing Real-time Applications," Proc. Fourth Int'l Workshop Real-Time Computing Systems and Applications, pp. 43-50, 1997.
38. David Langan, "EOS: an Object-oriented Operating System for Embedded Real-time Applications," Proc. ACM Conf. computer science, pp. 60-65, Feb. 1993.

39. D. Lefkon, B. Payne, "The Practical Engineer. Making Embedded systems Year 2000 Compliant," IEEE Spectrum, Vol. 356, pp. 74-79, June 1998.
40. W Liu, W. J. Kostis, and V. K. Prasanna, "Communication Issues in Heterogeneous Embedded Systems," Proc. Int'l Workshop Parallel and Distributed Real-Time Systems, pp. 180-183, 1996.
41. G. Luculli, and M. D. Natale, "A Cache-aware Scheduling Algorithm for Embedded Systems," Proc. IEEE Real-Time Systems Symposium," pp. 199-209, 1997.
42. Alan Mackworth, "Constraint-based Design of Embedded Intelligent Systems," ACM Comput. Surv. Vol. 28, No. 4, Dec. 1996. Alan Mackworth, "Constraint-based Design of Embedded Intelligent Systems," ACM Comput. Surv. Vol. 28, No. 4, Dec. 1996.
43. Noah Mendelsohn, "Operating Systems for Component Software Environments," Proc. 16th ACM Symposium on Operating Systems Principles, Saint-Malo, France, October 1997.
44. Clifford W. Mercer, Stefan Savage, and Hideyuki Tokuda, "Processor Capacity Reserves: Operating System Support for Multimedia Applications," Proc. IEEE International Conference on Multimedia Computing and Systems, Boston, Massachusetts, May 1994.
45. J. Miyao, "A Reliable Software Architecture for Complex Embedded Systems," Proc. IEEE Object-Oriented Real-time Distributed Computing, pp. 90-94, 1998.
46. M. Mrav, "Reuse Factors in Embedded System Design," IEEE Computer, Vol. 30, No. 8, pp. 93-95, Aug. 1997.
47. F. Mueller, V. Rustagi, and T. P. Baker, "MiThOS-a Real-time Micro-Kernel Threads Operating System," Proc. IEEE Real-Time Systems Symposium, pp. 49-53, 1995

48. D. Mulchandani, "Java for Embedded Systems," IEEE Internet Computing, Vol. 23, pp. 30-39, May-June 1998.
49. Jason Nieh and Monica S. Lam, "The Design, Implementation and Evaluation of SMART: A Scheduler for Multimedia Applications," Proc. 16th ACM Symposium on Operating Systems Principles, Saint-Malo, France, October 1997.
50. S. Petters et al., "The REAR Framework for Emulation and Analysis of Embedded Real-time Systems", Proc. IEEE Int'l Workshop Rapid Systems Prototyping, pp. 100-107, 1999.
51. P. Pop, P. Eles, and P. Zebo, "An Improved Scheduling Technique for Time-Triggered Embedded Systems," Proc. Euromicro Conference, Vol. 1, pp. 303-310, 1999.
52. F. M. Renner, J. Becker, and M. Glesner, "Communication Performance Models for Architecture-Precise Prototyping of Real-time Embedded System," Proc. IEEE Int'l Workshop Rapid Systems Prototyping, pp. 108-113, 1999.
53. J. Rufino, P. Verissimo, and G. Arroz, "Embedded Platforms for Distributed Real-time Computing: Challenges and Results," Proc. IEEE Object-Oriented Real-time Distributed Computing, pp. 147-152, 1999.
54. M. Ryu, J. Park, K. Kim, Y. Seo, and S. Hong, "Performance Re-engineering of Embedded Real-time Systems," Proc. ACM SIGPLAN Workshop Languages, Compilers, and Tools for Embedded Systems, pp. 80-86, 1999.
55. J. Sanders, "Linux, Open Source, and Software's Future," IEEE Software, Vol. 15, No. 5, pp. 88-91, Sept.-Oct. 1998.
56. M. I. Seltzer, Y. Endo, C. Small, and K. A. Smith. "Dealing With Disaster: Surviving Misbehaved Kernel Extensions," Proc. of the second Symposium on Operating Systems Design and Implementation. Seattle, WA, October 1996, pp. 213-227.

57. K. G. Shin et al., "A Distributed Real-time Operating System," *IEEE Software*, pp. 58-68, September, 1992.
58. J. Stankovic and K. Ramamritham, "The Spring Kernel: a New Paradigm for Real-time Operating Systems," *ACM Operating Systems Review*, Vol. 23, No. 3, pp. 54-71, July 1989.
59. John A. Stankovic, "Real-time and Embedded System," *ACM Comput. Surv.* Vol. 28, No. 1, pp. 205-208, Mar. 1996.
60. John A, Stankovic, "Strategic directions in Real-time and Embedded Systems," *ACM Comput. Surv.* Vol. 28, No. 4, pp. 751-763, Dec. 1996.
61. F. J. Suarez, J. Garcia, and D. F. Garcia, "Evaluation of Real-Time and Embedded Systems for Monitoring and Optimizing Industrial Processes Using High Performance Computing," *Proc. Eighth Euromicro Workshop Real-Time Systems*, pp. 127-132, 1996.
62. S. Suzuki, and K. G. Shin, "On Memory Protection in Real-time OS for Small Embedded Systems," *Proc. Fourth Int'l Workshop Real-Time Computing Systems and Applications*, pp. 51-58, 1997.
63. Dave Thomas, "Ubiquitous Applications Embedded Systems to Mainframe," *Commun. ACM*, Vol. 38, No. 10, pp. 112-114, Oct. 1995.
64. L. M. Thompson, "Using pSOS+ for Embedded Real-time Computing," *COMPCON*, pp. 282-288, 1990.
65. Jeffrey M. Thompson, Mats P. E. Heimdahl, and Steven P. Miller, "Specification-based Prototyping for Embedded Systems," *Proc. 7th ACM SIGSOFT Symposium on Foundations of Software Engineering*, pp. 163-179, 1999.
66. H. Tokuda, T. Nakajima, and P.Rao, "Real-Time Mach: Toward a Predictable Real-time System," *Proc. Usenix Mach Workshop*, pp. 73-82, 1990.

67. S. Tomoyoshi and T. Kosuke, "Table-based QoS Control for Embedded Real-time Systems," Proc. ACM SIGPLAN Workshop on Languages, Compilers, and Tools for Embedded Systems, pp. 65-72, 1999.
68. K. Weiss, T. Steckstor, and W. Rosenstiel, "Performance Analysis of a RTOS by Emulation of an Embedded System," Proc. IEEE Int'l Workshop Rapid Systems Prototyping, pp. 146–151, 1999.
69. T. S. Woo, S. J. Moon, and K. T. Keun, "Experience with TCP/IP Networking Protocol S/W Over Embedded OS for Network Appliance" Proc. Int'l Workshop Parallel Processing, pp. 556-561, 1999.
70. T. Y. Yen and W. Wolf, "Performance Estimation for Real-time Distributed Embedded System," IEEE Trans. Parallel and Distributed Systems, Vol. 911, pp. 1125-1136, Nov. 1998.
71. K. M. Zuberi and K. G. Shin, "An Efficient Semaphore Implementation Scheme for Small-Memory Embedded Systems," Proc. IEEE Real-Time Technology and Applications Symposium, pp. 25-34, 1997.
72. "Linux Online," <http://www.linux.org/>
73. "Real-Time Linux," http://rtlinux.cs.nmt.edu/~rtlinux/homepage_noframes.shtml.
74. "Linux Document Project," <http://metalab.unc.edu/mdw/index.html>
75. "The ELKS: Embeddable Linux Kernel Subset," <http://www.elks.ecs.soton.ac.uk/>
76. "VIMELinux: Linux for Your Embedded System," <http://www.vmelinux.org/>
77. "Pocket Linux Embedded Box Project," <http://www.cse.unsw.edu.au/~pleb/>
78. "LEM: Linux Embedded," <http://linuxembedded.com/>

計畫成果自評：

本子計畫的目標在於系統單晶片嵌入式 Linux 作業系統的研究與實

作。在三年的計畫中，我們首先釐定了 Linux 的 Hardware Dependent Layer，並移植 Linux 到 Evaluation Board 上。接著，我們也深入研究研究嵌入式作業系統的內部架構。例如：元件化的核心，即時的特性，與電源管理的機制。我們釐定了嵌入式作業系統內元件間的介面，並實作了嵌入式 Linux 的電源管理機制。除此之外，我們也將其他子計畫的 IP(e.g., WLAN MAC and Baseband Logic, MPEG Software)與嵌入式 Linux 進行整合，並在 ARM Integrator Board 上進行整個系統的驗證。經過驗證結果，我們已經可以在 ARM Integrator 上透過 WLAN MAC/Baseband 作 on-line peer-to-peer 的 MPEG 檔傳輸與播放。

整體說來，此計畫的成果與預期之結果大致符合。在此計畫中，我們除了完成此 SOC 雛形平台之外，最重要的，我們也增加了許多嵌入式作業系統的技术與軟硬體共同驗證的經驗，對未來的研究有頗大助益。

附件二

可供推廣之研發成果資料表

可申請專利

可技術移轉

日期：92年10月15日

國科會補助計畫	計畫名稱：應用於系統單晶片嵌入式 Linux 作業系統之實作 計畫主持人：張瑞川
----------------	---

	計畫編號： 學門領域：資訊工程
技術/創作名稱	系統單晶片嵌入式 Linux 作業系統之實作
發明人/創作人	張瑞川
技術說明	中文： 我們釐定了 Linux 的 Hardware Dependent Layer，並移植 Linux 到 Evaluation Board 上。接著，我們也深入研究嵌入式作業系統的內部架構，並實作了嵌入式 Linux 的電源管理機制。除此之外，我們也進行嵌入式 Linux 之 SOC 整合，並在 ARM Integrator Board 上進行整個系統的驗證。
	英文： We have identified the Hardware dependent layer of Linux, and ported it to the evaluation board. Then, we evaluate the internal structure of the embedded Linux, and implemented a power management mechanism. Finally, we integrated the embedded Linux into the ARM Integrator Board and verified the overall system.
可利用之產業 及 可開發之產品	嵌入式系統 作業系統 消費型電子產品
技術特點	Embedded Linux Integration into SOC Power Management on Linux
推廣及運用的價值	Embedded Linux Integration into SOC Power Management Techniques

1. 每項研發成果請填寫一式二份，一份隨成果報告送繳本會，一份送貴單位研發成果推廣單位（如技術移轉中心）。
2. 本項研發成果若尚未申請專利，請勿揭露可申請專利之主要內容。
3. 本表若不敷使用，請自行影印使用。