



MICROTIME

Linux For Creator-S3C2410

User's Guide

Ver : 1.1

Taipei :

4F, No.222, SEC.4, CHENG-TEH Rd. Taipei, Taiwan, R.O.C.

TEL : 886-02-2881-1791

FAX : 886-02-2882-0836

Kaushung :

9F. -6, No. 91, Chung-Shan 2nd Rd., Chianjen Chiu, Kaohsiung, Taiwan, R.O.C.

TEL : 886-07-330-0818

FAX : 886-07-334-8286

<http://www.microtime.com.tw>

E-mail : sales@microtime.com.tw

Copyright Notice

Copyright ©2004 Microtime Computer Inc. All right reserved.

No part of this publication may be reproduced, translated into another language, store in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise without the prior written consent of Microtime Computer Inc.

Disclaimer

This information in this document is subject to change without notice.

Every precaution has been taken in the preparation of this publication. Microtime assumes no responsibility for error or omission. Neither is any liability assumed for damages resulting from the use of the information contained herein.

Trademarks

DOMINGO are trademarks of Microtime computer Inc.

All other product names are trademarks or registered trademarks of their respective owners.

Reversion History

V1.0 : 2004-10-08

V1.1 : 2004-12-23

目錄

前言

第一章 [系統需求](#)

第二章 [建立 Creator-S3C2410 的 Linux Kernel Image](#)

2-1 [安裝 Red Hat 9.0](#)

2-2 [安裝 Toolchain 及 Linux 套件](#)

2-3 [安裝 Linux Kernel Source 及 patch 檔](#)

2-4 [編譯 Linux Kernel Image](#)

2-5 [驗證 Linux Kernel Image](#)

2-5-1 [Linux Samba 伺服器的設定](#)

2-5-2 [將 Linux kernel image 下載到 RAM 的驗證方式](#)

2-5-3 [將 Linux kernel image 燒錄到 Flash 的驗證方式](#)

第三章 [如何修改 Creator-S3C2410 的 Kernel Address 和 Clock Parameter](#)

附錄 A [Samba 伺服器的安裝與設定](#)

附錄 B [Root File System 的建立與修改](#)

B-1 [建立 ext2 file system](#)

B-2 [建立 jffs2 file system](#)

前言

Embedded Linux 是 Linux 的一個應用，利用 Embedded Linux 技術發展嵌入式的最大好處是，我們可以將整個發展重心從硬體與組合語言程式拉到應用程式層面，目前網路上有相當多的 Embedded Linux 的相關資料可用。Linux Kernel 是一個強大、可信賴、具彈性與擴充性的作業系統核心（OS Kernel），Linux Kernel 實作了許多現代化作業系統理論，並且支援完整的硬體驅動程式、網路通訊協定與多處理器的架構，同時，Linux 的開放源碼發展模式，可以很容易掌握所有的技術細節，使我們可以有效的發展自己的 Embedded Linux 系統。

首先跟大家約略說明新華電腦提供的”完整的嵌入式系統發展工具解決方案”，一個完整的嵌入式系統發展工具解決方案包含了以下：

1. 線路實體模擬器(In-Circuit Emulator, ICE)：就是模擬 CPU 動作的硬體裝置，利用 PC 下載 obj code 用來模擬執行的狀況，也是一個極高效率的除錯工具。也就是我們所提供的 Carrier ICE。
2. 系統發展平台(Development Platform)：由硬體廠商所提供的設計樣本，下游的產品開發者可以做軟體開發與功能驗證，加速自己的研發進度。也就是我們提供的 Creator 母板與各子板。
3. 應用程式整合開發除錯工具(Debugger)：方便的整合開發環境，也有助於程式碼的管理、開發及除錯。我們提供的 Dominigo 就是一個很好的應用程式整合開發、除錯環境(IDE)。
4. 編譯器(Compiler)等工具：就是編譯 Linux 核心、模組及使用者應用程式的開發環境。在此我們利用 Linux，使用者可以在 Linux 的工作環境下編譯 Linux 核心、模組等...

Creator-S3C2410 嵌入式系統中 Linux Host 端所使用的作業系統是 Red Hat9 Linux，本文的主旨在說明如何在 Linux Host 端建立 Linux kernel image、Root Filesystem 及如何利用 Windows Host 端執行安裝與下載等工作.....做一詳盡的介紹。

在開始介紹之前，要請使用者特別注意本文第一章的系統需求，對於作業系統的安裝一定要符合需求，因為我們接到部分使用者反映的問題，最後發現大部分都是因為忽略系統需求而引起的，所以這一點請使用者要特別注意。

第二章說明如何安裝 Red Hat9 Linux、編譯 Linux kernel image 及操作等...，其整個章節的說明也是根據安裝、操作的步驟依序編排而成，所以使用者按照規劃的章節依序閱讀，便可詳細了解如何安裝、編譯等相關議題，希望藉著本文的介紹使用者能更快進入 Embedded Linux 的世界。

註 1：

我們通常將 Linux 的 execution mode 稱為 kernel-space 與 user-space，他們各自具有自己

的 memory mapping，也就是說，他們各自擁有自己的 address space，如模組是屬於 kernel space 而應用程式是屬於 user space。

第一章 系統需求

Creator-S3C2410 嵌入式系統的系統需求為下列幾點，以下分別說明其版本及使用時機：

1. Linux Host 端：
作業系統：Red Hat9 Linux。
主要是做為編譯器(Compiler)的使用，編譯 Linux kernel image、Root Filesystem、device driver、使用者應用程式等...。
2. Windows Host 端：
作業系統：WIN 2000/NT/XP/98。
提供應用程式整合開發環境(IDE)—Domingo 的安裝環境。
3. Linux Target 端：
作業系統：Embedded Linux
就是 Microtime 提供的 Creator 母版、Creator-S3C2410 子板等....。
4. 安裝系統必須包含光碟機。

第二章 建立 Creator-S3C2410 的 Linux Kernel Image

首先必須在 Linux Host 端安裝 Red Hat9 Linux 作業系統，因為 Linux kernel image、Root Filesystem、device driver、是用者應用程式都必須在 Linux 的環境下編譯。

要建立 Linux kernel image，首先你必須取得 Linux Kernel 的原始程式碼，由於 Linux 是開放源碼的發展模式，所以我們可以在網路上輕易取得原始程式碼版本，原始程式碼的檔案名稱一般是以標記成 linux-x.y.z.tar.gz 為代表，其中的 x.y.z 是版本編號。

取得原始程式碼後，要做的是解開原始程式碼包裝，通常在/usr/src 目錄下，用`tar -zxvf linux-x.y.z.tar.gz' 來解開原始程式碼的包裝,當它完成以後，在/usr/src 將會出現一個新的`linux' 子目錄，接下來要做是做配置核心的工作及編譯 Linux kernel image。

一般而言 Linux 啟動時包含三個部分以下概略介紹：

- ◆ Boot Loader：開機管理程式(Boot Loader)是當電腦啟動時第一個執行的軟體程式，它負責載入並傳遞控制到作業系統的核心軟體。
- ◆ Root Filesystem：Linux 啟動時第一個掛載的檔案系統，掛載的目錄是在根目錄下。
- ◆ Kernel：Linux 作業系統的核心。

在建立 Linux 核心時，我們可以個別建立 Boot Loader、Root Filesystem、Linux kernel image，也可以將這三個部分合併成一個 Image，這全看你要如何規劃。Creator-S3C2410 嵌入式系統採用前者的規劃方式。

Linux Host 端在建立 Embedded Linux 核心時，會在 /usr/src/creator/s3c2410/linux/arch/arm/boot 路徑下產生 zImage 檔，而 Root Filesystem 則 Creator-S3C2410 嵌入式系統 Linux kernel 安裝光碟有提供。以下將詳細說明如何建立 Linux kernel image。

2-1 安裝 Red Hat 9

在 Linux Host 端編譯 Linux kernel image、Root Filesystem、device driver、使用者應用程式必須在 Linux 的環境下編譯，所以首先要先安裝 Linux 作業系統，因為 Linux 作業系統的供應商及版本很多，為了說明及使用上的一致性，所以我們使用 Red Hat 9，也建議使用者安裝這個版本。

2-2 安裝 Toolchain 及 linux 套件

要編譯 Linux kernel image 之前必須先安裝 Toolchain，以提供在編譯時所需的工具，我們在這裡使用的是 GNU 爲了 ARM 所做的 Cross Compiler Toolchain，什麼是 Cross Compiler 呢? Cross Compiler 就是在「某個類型的處理器平台」中產生在「其他種類處理器」可執行檔的編譯器。GNU Cross Compiler Toolchain 就是 GNU 提供的開發工具鍊。

Creator-S3C2410 嵌入式系統 Linux kernel 安裝光碟中提供的 cross-2.95.3-arm.tar.gz 檔就是 GNU 提供的發展 arm elf 的開發工具鍊，而且已經編譯過，所以我們只要將此檔案解壓縮到適當的路徑即可，請放入 Creator-S3C2410 嵌入式系統 Linux kernel 安裝光碟，請依照下列步驟執行：

◆ 建立 armtools 目錄及複製 Tool Chain 檔，步驟如下：

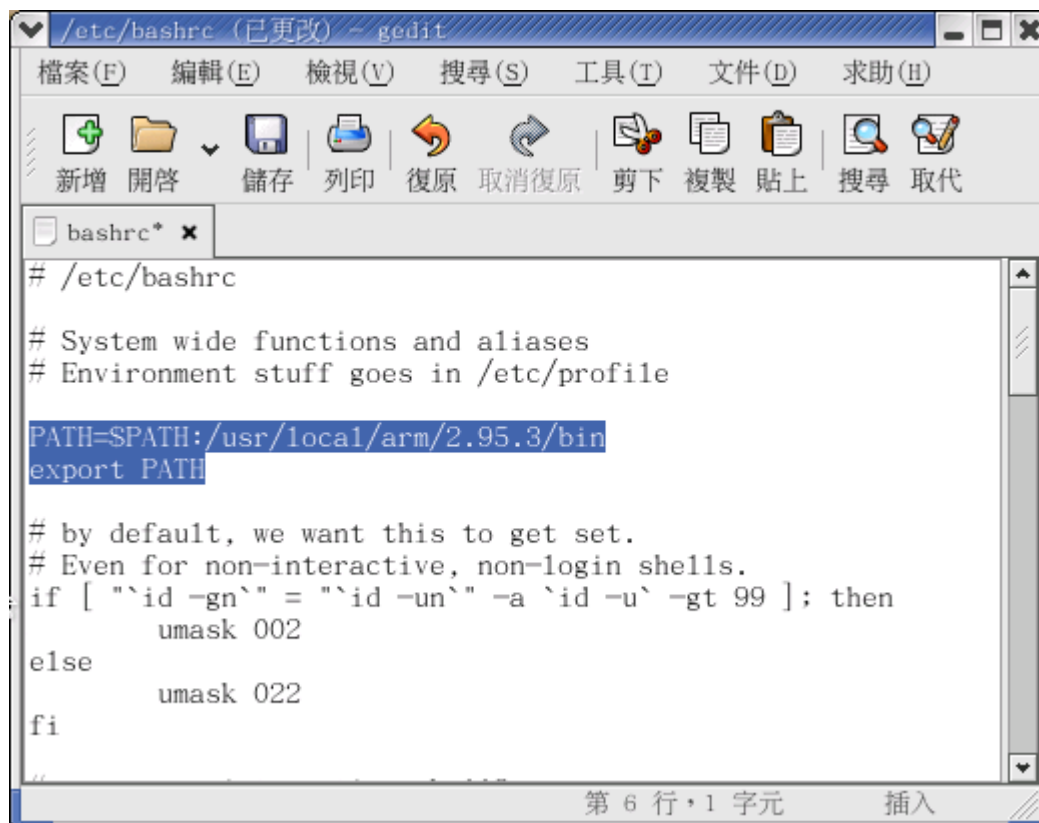
1. 請在 Linux Host 端桌面或”主選單” 下執行”個人資料夾”，利用”個人資料夾”建立”usr/local/arm”路徑。
2. 因為 Linux Host 端的光碟機目錄是在/mnt/cdrom，所以複製/mnt/cdrom/source/目錄下的 cross-2.95.3.tar.bz2 檔到/usr/local/arm 路徑下。

◆ 執行 Linux Host 端的終端機(可在 Linux Host 端桌面或”主選單/系統工具”下執行終端機)，再解壓縮 Tool Chain 檔，如下：

```
cd /usr/local/arm  
bzip2 -dc cross-2.95.3.tar.bz2 | tar xf -
```

◆ 設定 Tool chain PATH，我們將利用 Linux Host 端提供的文字編輯器編輯/etc/bashrc 檔，首先在 Linux Host 端執行”主選單/輔助程式/文字編輯器”此時開啓”文字編輯器”，請開啓目錄/etc 下的 bashrc 檔，加入下列兩行如下：

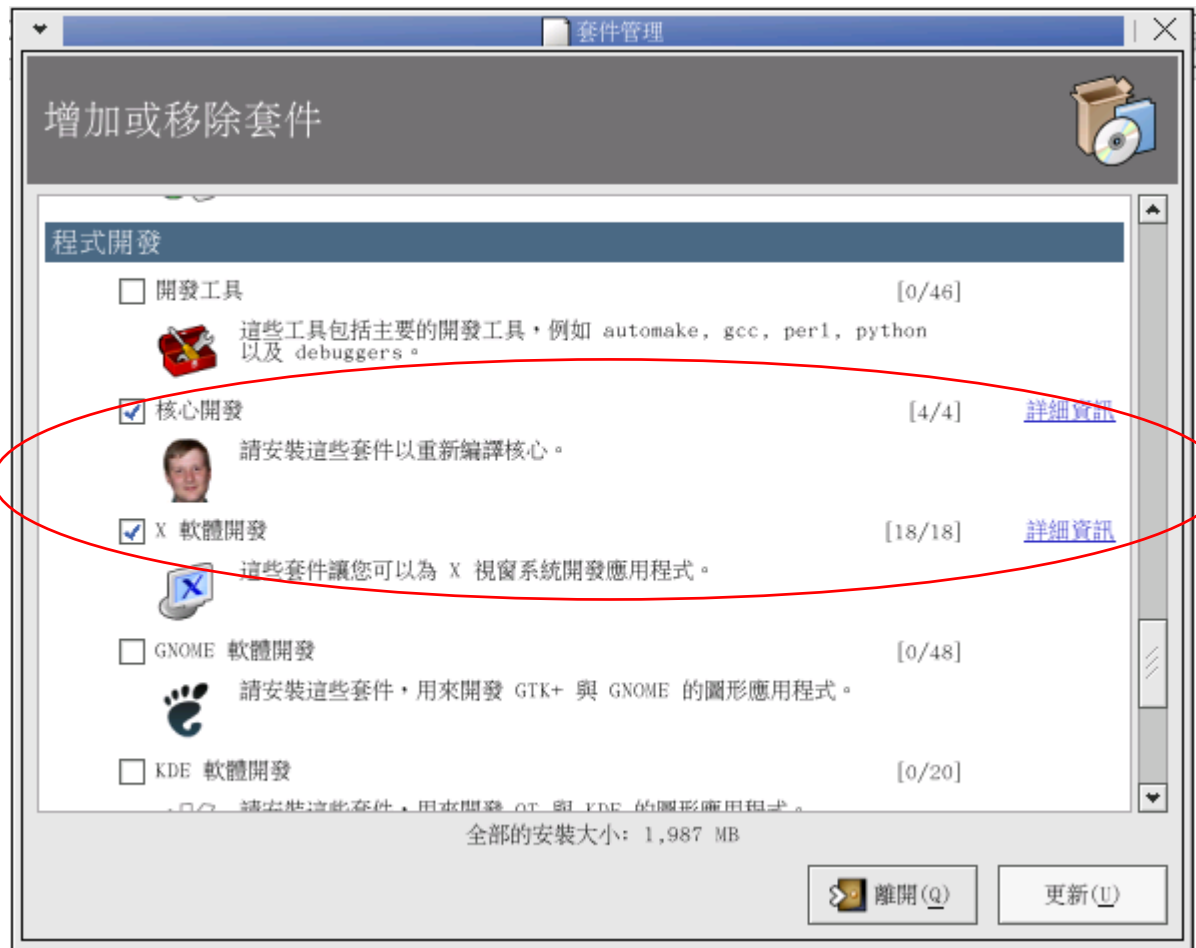
```
PATH=$PATH:/usr/local/arm/2.95.3/bin  
export PATH
```



修改完畢後，請將 Linux Host 端作業系統登出再登入，已確定路徑已經加入環境變數 PATH 中。

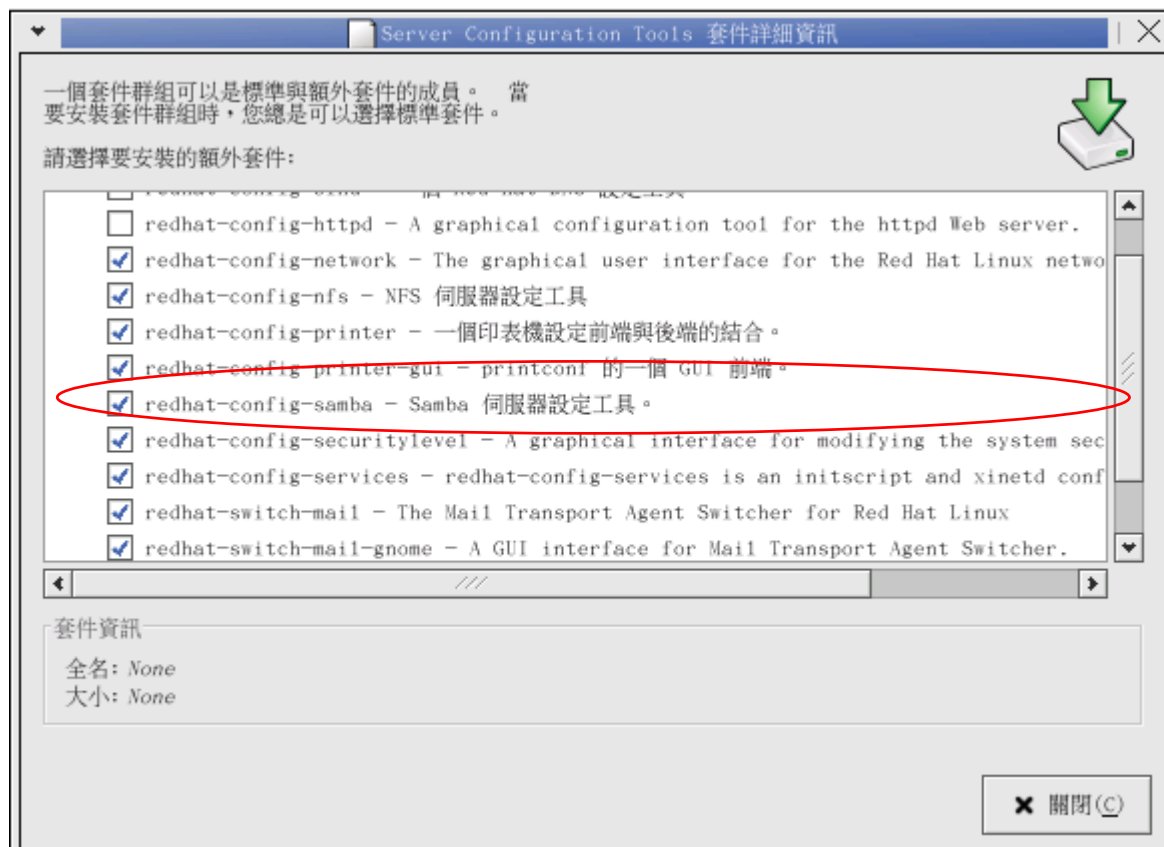
接下來我們要為 Linux Host 端作業系統安裝一些套件，請在 Linux Host 端”主選單”下選擇”系統設定”的”新增移除應用程式”，在新增移除應用程式對話框中做以下設定：

- ◆ 選擇”程式開發”下的”核心開發”及”X 軟體開發”中所有套件。



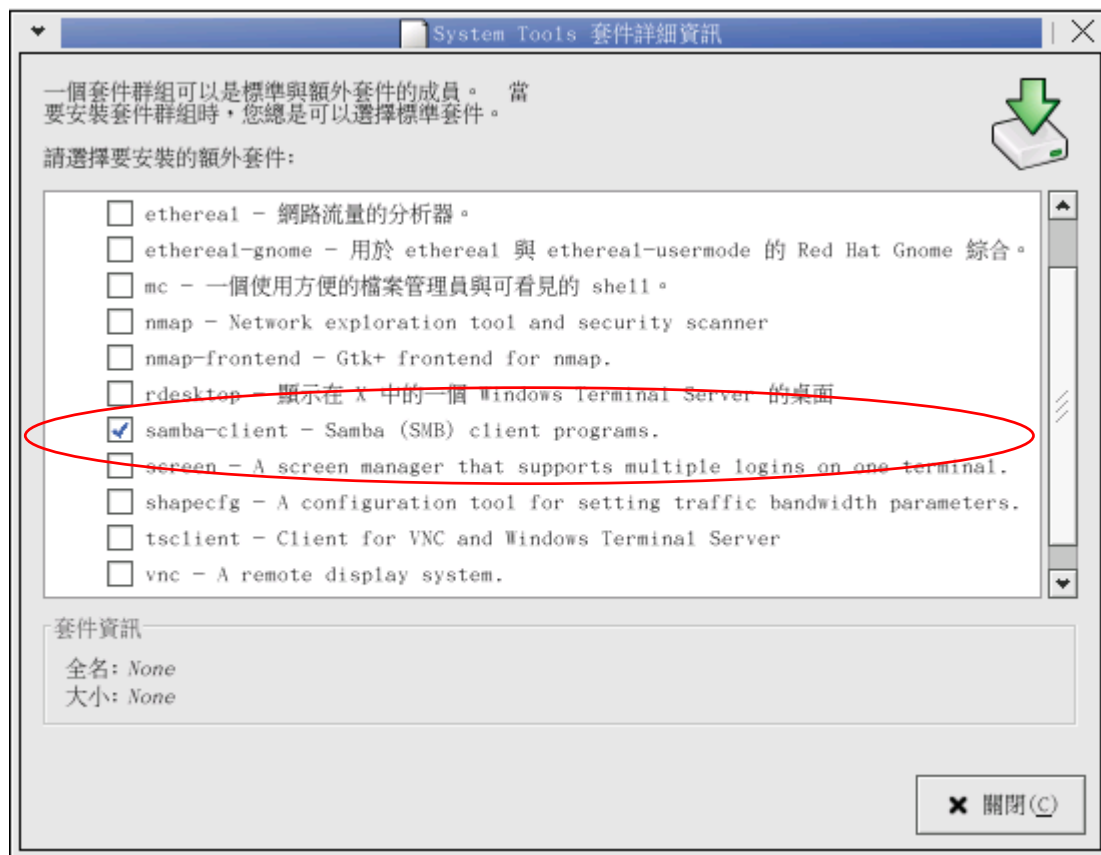
選擇完成後，請按更新鈕。

- ◆ 選擇伺服器→伺服器設定工具→redhat-config-samba - Samba 伺服器設定工具。



選擇完成後，請按更新鈕。

- ◆ 選擇系統→系統工具→samba-client - Samba (SMB) client program。



選擇完成後，請按更新鈕。

2-3 安裝 Linux Kernel Source 及 patch 檔

接下來要將 Linux kernel source 解壓縮及做部分檔案修改，已確保 Linux kernel source 可以在 Linux Host 端能正確的編譯，請放入 Creator-S3C2410 嵌入式系統的 Linux kernel 安裝光碟，請開啓 Linux Host 端終端機依照下列步驟執行：

- ◆ 建立 Linux kernel source 存放目錄及複製 Linux kernel source 檔，步驟如下：

1. 請在 Linux Host 端桌面或”主選單” 下執行”個人資料夾”，利用”個人資料夾”建立”usr/src/creator/s3c2410”路徑。
2. 因爲 Linux Host 端的光碟機目錄是在/mnt/cdrom，所以複製/mnt/cdrom/source/目錄下的 linux-2.4.18.tar.gz、patch-2.4.18-rmk7.gz、patch-2.4.18-rmk7-sw18 檔到 /usr/src/creator/s3c2410 路徑下。

- ◆ 解開 Linux kernel source 檔，請在 Linux Host 端終端機中，執行如下：

```
cd /usr/src/creator/s3c2410
tar -zxvf linux-2.4.18.tar.gz
gzip -dc patch-2.4.18-rmk7 | (cd linux; patch -p1)
gzip -dc patch-2.4.18-rmk7-sw18 | (cd linux; patch -p1)
(註：其中”patch -p1”的’1’是阿拉伯數字的 1)
```

- ◆ 複製 Creator-S3C2410 patch 檔，因爲 Linux Host 端的光碟機目錄是在/mnt/cdrom，所以複製/mnt/cdrom/source/目錄下的 patch-2.4.18-creator2410.tar.gz 檔到 /usr/src/creator/s3c2410 路徑下。

- ◆ 解開 Creator-S3C2410 patch 檔，請在 Linux Host 端終端機中，執行如下：

```
cd /usr/src/creator/s3c2410
tar -zxvf patch-2.4.18-creator2410.tar.gz
```

2-4 編譯 Linux Kernel Image

在開始建立 Linux kernel image 之前，約略介紹建立步驟的各個命令及說明，如下：

make dep : 建立相依的屬性關係！

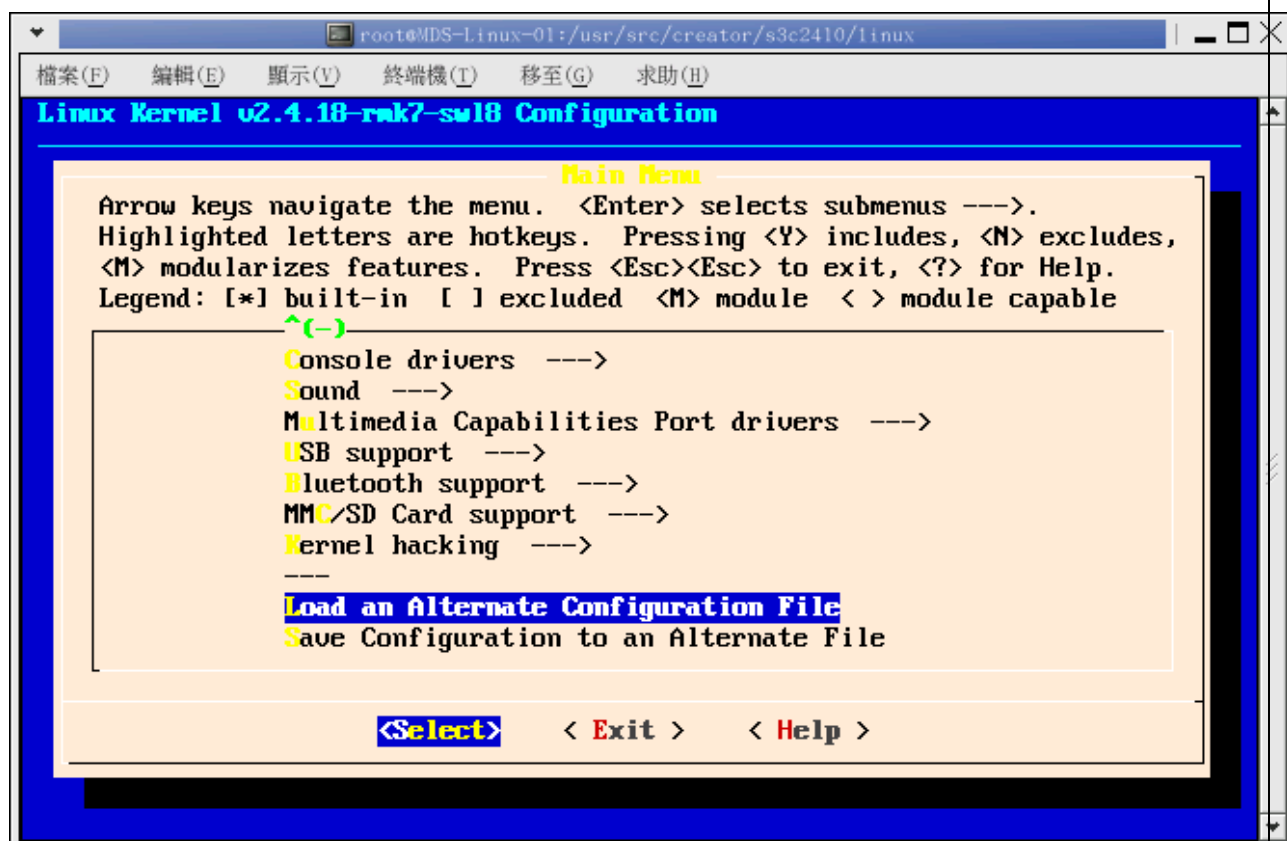
make clean : 將舊的資料去除掉！

make zImage : 開始編譯核心，核心經過壓縮，且以 RAM 爲記憶體 0 的位置。

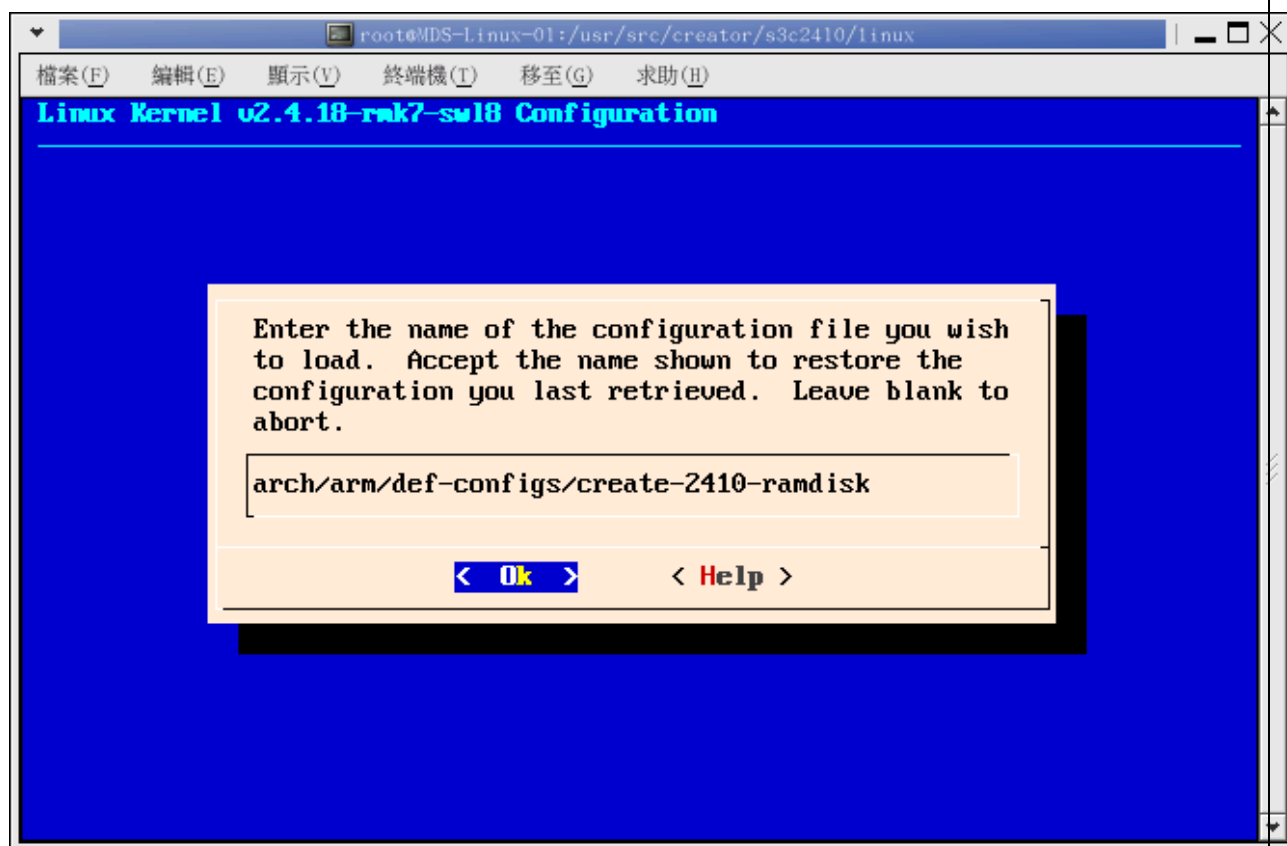
現在，我們要開始建立 Linux kernel image，請 Linux 終端機下執行，操作步驟如下：

- ◆ 開啓 Linux Host 端終端機。
- ◆ 設定 Embedded Linux 的 configuration，如下：

```
SHELL> cd /usr/src/creator/s3c2410/linux
SHELL> make menuconfig
```



- ◆ 出現"Linux Kernel Configuration"對話框後，選擇 "Load an Alternate Configuration from File"，則出現 Load an Alternate Configuration from File 對話框，如下：



- ◆ 請輸入”arch/arm/def-configs/create-2410-ramdisk”後，按”OK”鈕回到”Linux Kernel Configuration”對話框，此時按”Save and Exit”鈕，完成 configuration 的設定。
- ◆ 接下來開始建立 Linux kernel image，請在 Linux Host 端終端機執行如下：

```
cd /usr/src/creator/s3c2410/linux
make dep
make clean
make zImage
```

- ◆ 完成建立 Linux kernel image 後，可在/usr/src/creator/s3c2410/linux/arch/arm/boot 路徑下產生 zImage 檔案，爲了提供給別的作業平台使用，所以我們要預先將檔案複製到 export 的目錄下並更名爲 zImage_ram，執行如下：
 1. 請在 Linux Host 端桌面或”主選單” 下執行”個人資料夾”，利用”個人資料夾”建立”/usr/src/creator/nfs/s3c2410”路徑。
 2. 複製/usr/src/creator/s3c2410/linux/arch/boot 目錄下的 zImage 檔到 /usr/src/creator/nfs/s3c2410 路徑下，並修改檔名爲 zImage_ram。
- ◆ 我們已經順利產生 Linux kernel image 現在你可以參考下一章節驗證所產生的 Linux kernel image 是否正確。

2-5 驗證 Linux Kernel Image

要驗證編譯好的 Linux kernel image 是否正確無誤，唯一的方法就是將編譯好的 Linux kernel image 下載到 Linux Target 端，啓動 Embedded Linux 後再做相對應的檢查是否正確無誤。在這個章節理我們將介紹兩種方法來驗證 Linux kernel image，第一是將 Linux kernel image 及 Root Filesystem 下載到 Linux Target 端的 RAM 中再啓動 Embedded Linux，第二是將 Linux kernel image 及 Root Filesystem 燒錄到 Linux Target 端的 Nand flash 中再啓動 Embedded Linux。這兩種方式各有其好處，但最終的目標就是讓 Linux kernel image、Root Filesystem 在 Linux Target 端執行。

在驗證之前我們會發現目前所編譯好的 Linux kernel image 及 Root Filesystem 都在 Linux Host 端作業系統中，但是我們現在做驗證的環境卻是在 Windows Host 端，所以我們要想辦法從 Linux Host 端取得編譯好的 image，在這裡我們將在 Linux Host 端安裝 Samba 伺服器，使 Linux Host 端的檔案可以在不同的作業系統平台上分享。

2-5-1 Linux Samba 伺服器的設定

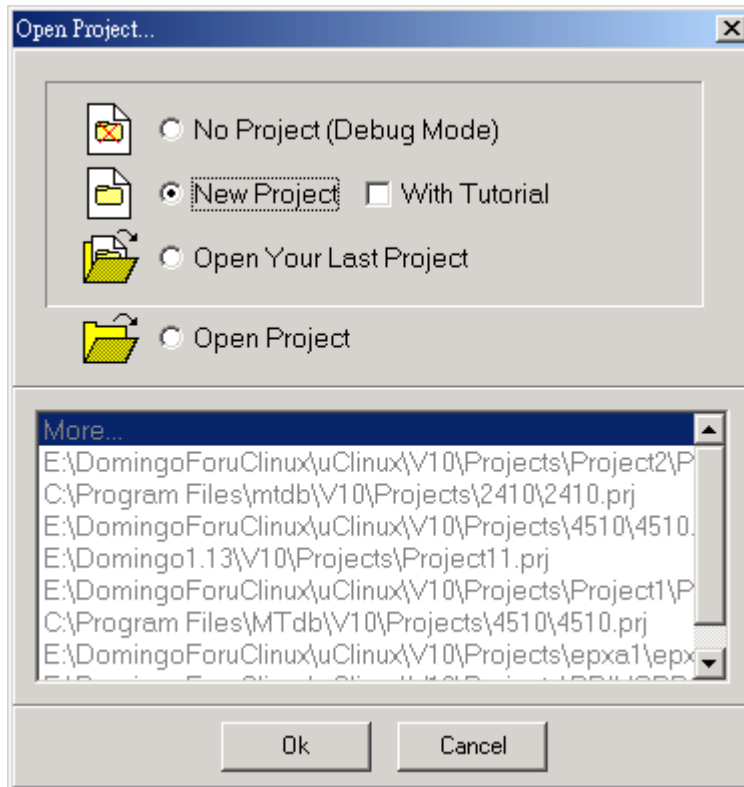
Samba 伺服器可以讓我們在 Linux Host 端上的檔案，在不同的作業平台分享，所以在我們的驗證過程中也將利用 samba 伺服器，讓我們可以在 Windows Host 端的作業系統下讀取 Linux Host 端的檔案。

Linux Host 端的 Samba 伺服器的設定請參考”[附錄 A Samba 伺服器的安裝與設定](#)”。

2-5-2 將 Linux kernel image 下載到 RAM 的驗證方式

這個部分將利用 Domingo for Linux 提供的 load module 功能將 Linux kernel image、Root Filesystem 下載到 Linux Target 端的 RAM 中再啟動 Embedded Linux，請依下列步驟執行：

- ◆ 將 Linux kernel image 下載到 Linux Target 端 RAM 的驗證方式，就是由 RAM 啟動 Embedded Linux。所以要先確認 Linux Target 端 Creator-S3C2410 子板的相關接腳正確無誤，請檢查 Creator-S3C2410 子板的 J19 pin 的 1、2 腳位應相連，而 J3 pin 腳應空接。
- ◆ 確認 Windows Host 端的 Domingo for Linux 已安裝無誤。
- ◆ 確認 Windows Host 端與 ICE、Linux Target 端連結無誤，詳細連結方法請參考”Creator PreSOCes Development Kit User`s Guide”一書。
- ◆ 開啓 Domingo for Linux，此時出現”Open Project...”對話框，因為我在現在只是要做 Download Linux Kernel Image 的動作，所以選擇”No Project(Debug Mode)”，之後按”OK”鈕。



- ◆ 再來會出現 Hardware Option...對話框，其中設定如下：

TargetMachine	選擇	WINEZ(WIN2K)ARM
Processor	選擇	S3C2410X
Memory Model	選擇	Little Endian
Non-cacheable Memory Address(Reserved 8 Bytes) 設定為 40000000		

Hardware Options...

TargetMachine:
WINaZ (WIN2K) ARM

Processor:
S3C2410X

Project Setup

Parallel interface (print port)
378
☐ PC adapted SMC super I/O

Option
TIME OUT : 256 (DEC)

Memory Model
☒ Little Endian
☐ Big Endian

Clock Select
☐ RTCK
☒ ICE Clock 20 MHz

Peripheral Configuration Manager(PCM)
☒ To Initialize PCM when Connect Config PCM...

☒ Coprocessor Option
Non-cacheable Memory Address(Reserved 8 bytes) : 40000000

Advanced Options
System Recovery Counts (6~255) : 15 (DEC)

Ok Cancel

再來設定 "Peripheral Configuration Manager(PCM)" group box 。
其中 "To Initialize PCM when Connect" CheckBox 選擇 Checked，之後按
"Config PCM" 鈕，開啓 Config Peripheral Configuration Manager 對話框，設定
PCM 檔。

Config Peripheral Configuration Manager

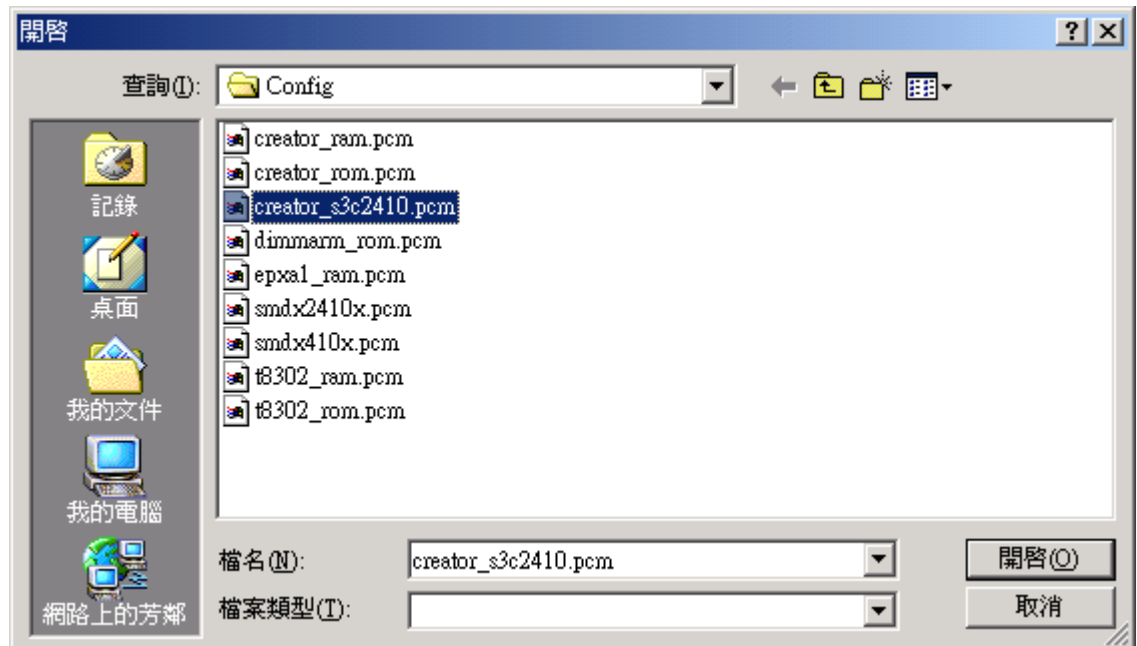
No	Register	Address	Value	Size
01	WTCON	53000000	0	4
02	BWCON	48000000	22111120	4
03	BANKCON0	48000004	1700	4
04	BANKCON1	48000008	2700	4
05	BANKCON2	4800000C	700	4
06	BANKCON3	48000010	700	4
07	BANKCON4	48000014	700	4
08	BANKCON5	48000018	700	4
09	BANKCON6	4800001C	18005	4
10	BANKCON7	48000020	18005	4
11	REFRESH	48000024	8E0459	4
12	BANKSIZE	48000028	B2	4

Move Up Move Down

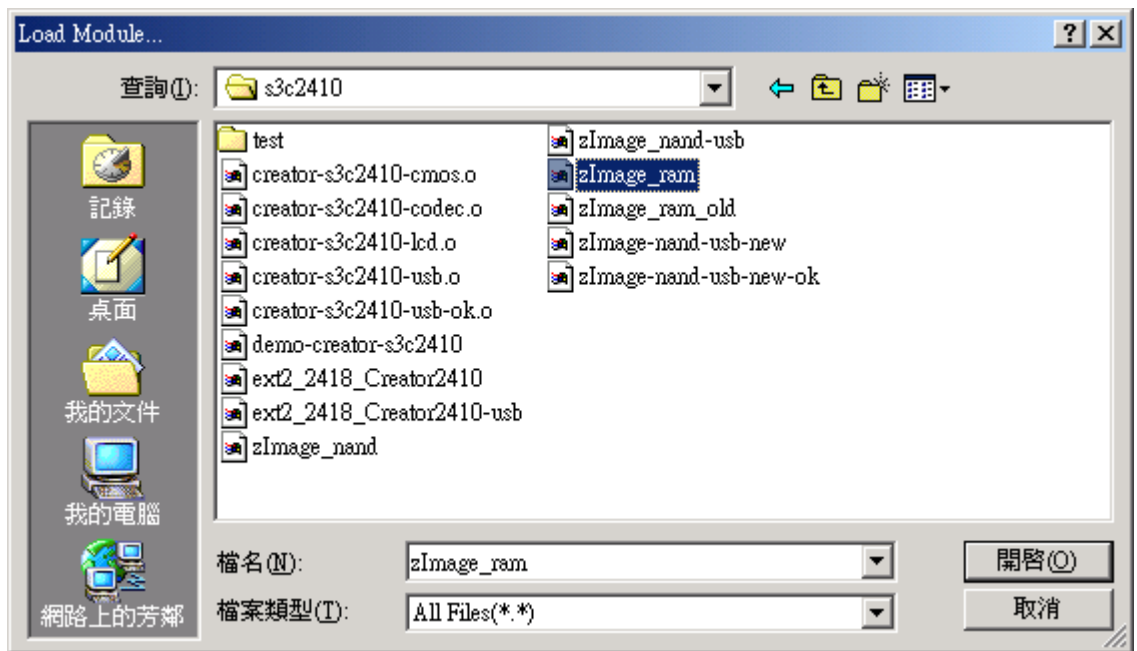
Append Insert Modify Delete Delete All

Import... Export... OK Cancel Apply

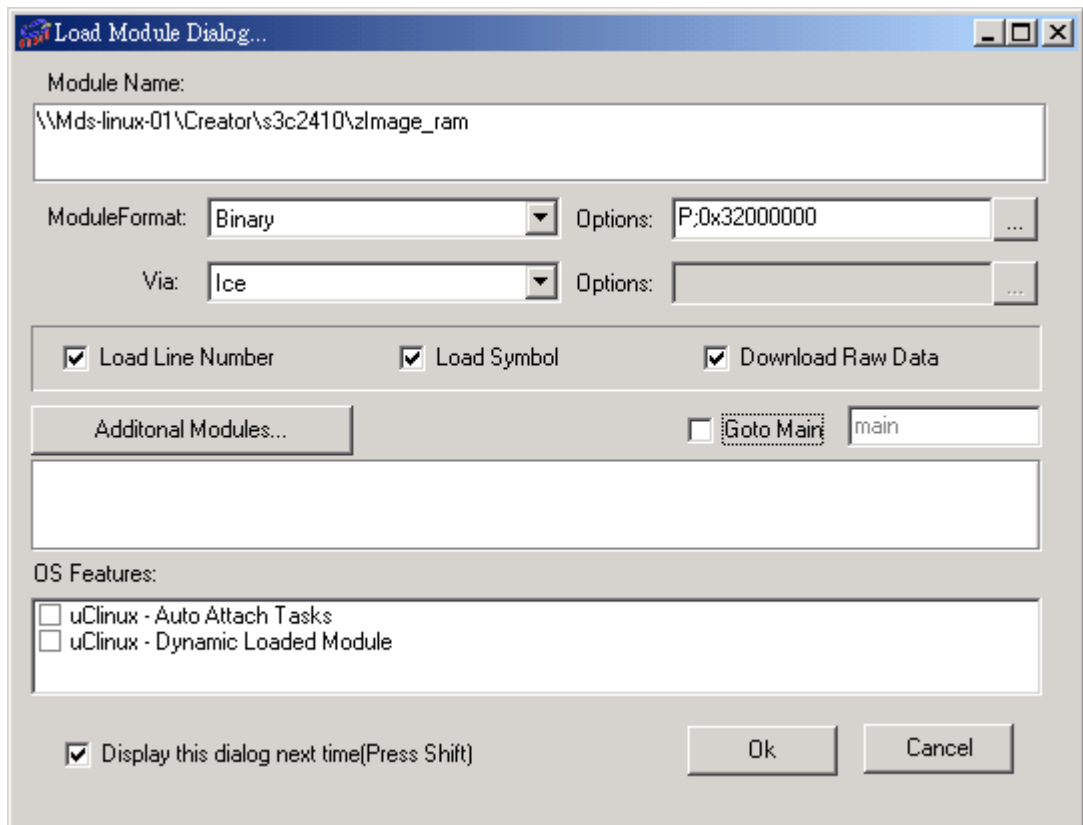
要設定 PCM 檔，請按”Import...”鈕，選擇 Domingo for Linux 安裝路徑下 \LinuxXXX\ARM\ARM7\Config\ creator_s3c2410.PCM 檔即可(XXX：表示版本)，如下：



- ◆ 以上完成了 HardWareOptions...的設定，程式進入 Domingo for Linux 主畫面後，此時 Domingo for Linux 與 ICE 要設定為連結狀態，請選擇 Debug\Connect 使 Domingo for Linux 與 ICE 連結。
- ◆ 連結成功後，要下載 Linux Host 端的 Linux kernel image 檔，在下載之前要確認 Linux Host 端的 Samba 伺服器已安裝完成(安裝請參考”附錄 A Samba 伺服器的安裝與設定”)，我們假設 Linux Host 端因 Samba 伺服器的設定已分享在”Microtime”工作群組之下，Linux Host 端的電腦名稱爲”Mds-linux-01”。所以請選擇 Domingo for Linux 中 menu 的”Debug\LoadModule”，選擇網路上的芳鄰中的 Microtime 群組下的\\Mds-linux-01\Creator\nfs\s3c2410\zImage_ram 檔，如下：



- ◆ 選擇 kernel image 檔之後，出現”Load Module Dialog...”對話框，如下：

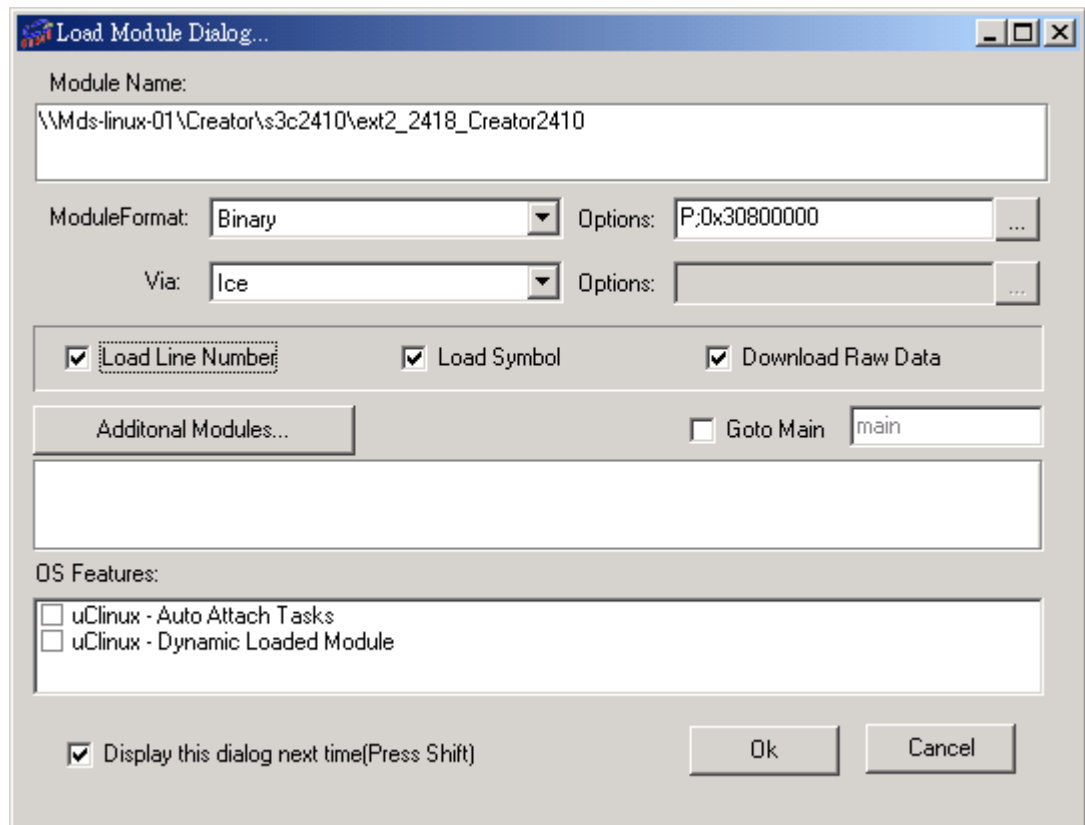


請設定，Options = P,32000000，Goto Main = unchecked。

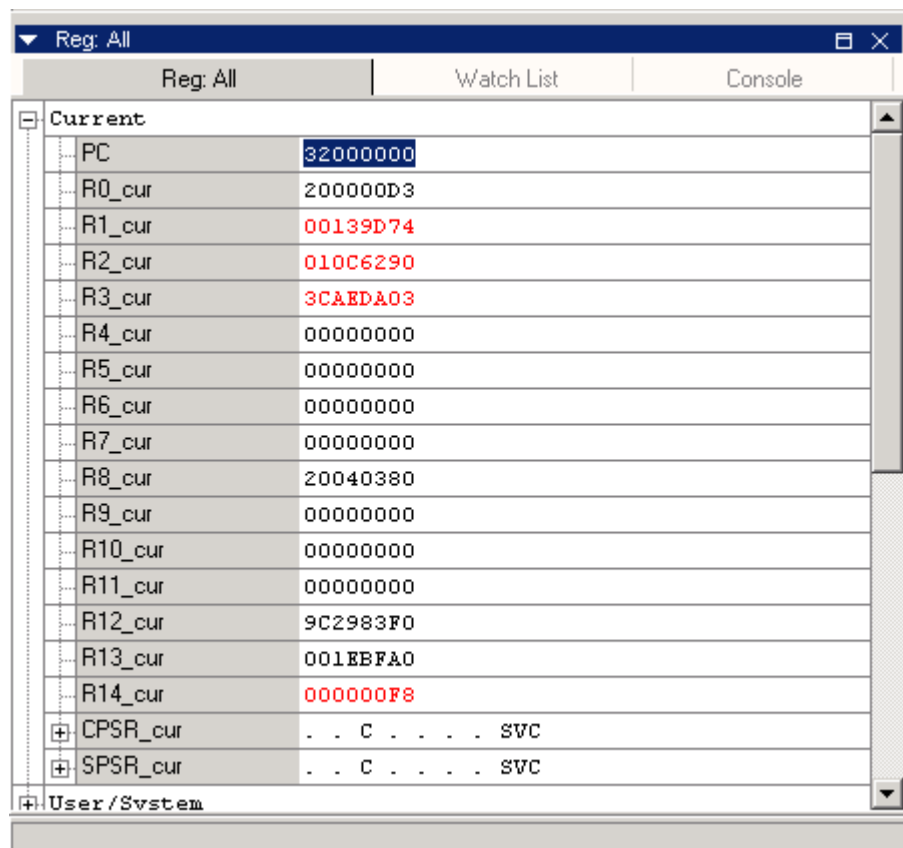
按 OK 後，便將 zImage_ram 檔下載到 0x32000000 位置。

- ◆ 下載 Linux Host 端的 Root Filesystem 請選擇”Debug\LoadModule”，選擇網路上的芳鄰中的 Microtime 群組下的 \\Mds-linux-01\Creator\nfs\s3c2410\ext2_2418_Creator2410 檔(假設 Linux Host 端因 smaba 伺服器設定在”Microtime 群組下，電腦名稱爲”Mds-linux-01”)，之後出現”Load Module Dialog...”對話框，請設定，Options = P,30800000，Goto Main =

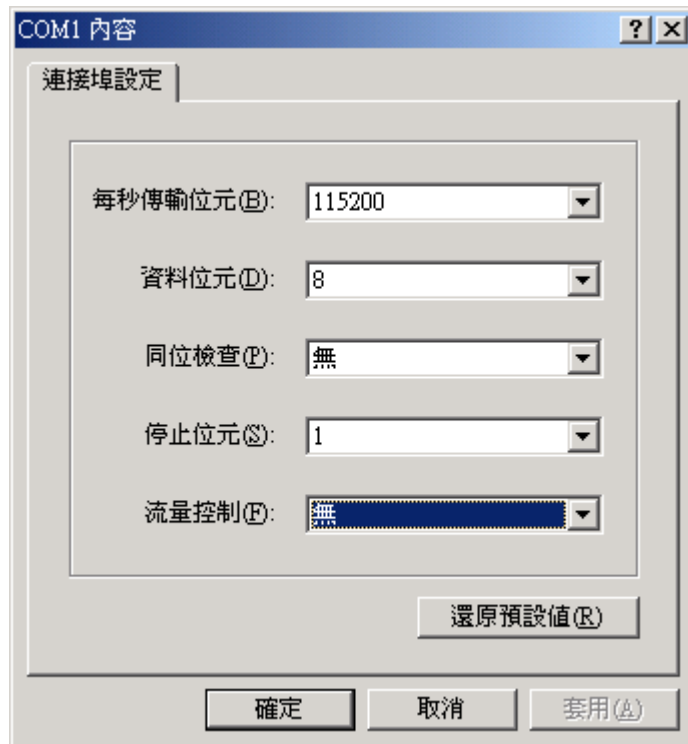
unchecked。按 OK 後，便將 Root Filesystem 檔下載到 0x30800000 位置，如下：



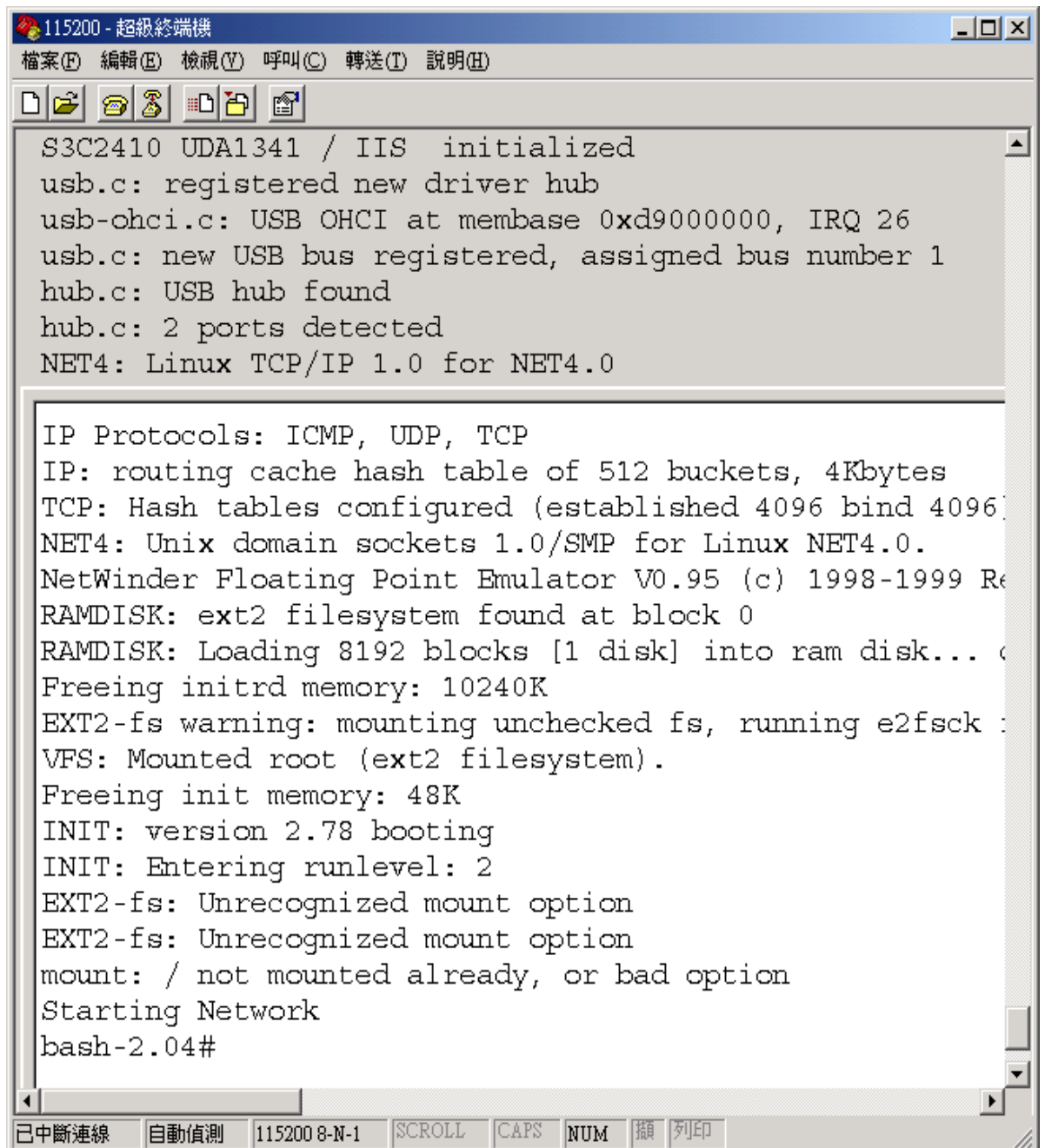
- ◆ 現在 Linux kernel image 及 Root Filesystem 都已經下載到 Linux Target 端的 RAM 中，此時觀察 Domingo for Linux，processor 應為暫停，接下來要把 Register 的 PC 值設為 32000000，此時執行的指標便會指到 Linux kernel image 開始處，如下：



- ◆ 接下來要啟動 Linux Target 端的 Embedded Linux，在執行之前要先設定 Windows Host 端的超級終端機，以顯示 Linux Target 端的 Embedded Linux 啟動的資訊，終端機的設定如下：



- ◆ 終端機連結完成後，請按 Domingo 的  鈕，此時 processor 為 Free go 的狀態，便會從 0x32000000 處啟動 Linux Target 端的 Embedded Linux，其開啓的相關資訊也會在 Windows Host 端的超級終端機顯示，如下：



```
S3C2410 UDA1341 / IIS initialized
usb.c: registered new driver hub
usb-ohci.c: USB OHCI at membase 0xd9000000, IRQ 26
usb.c: new USB bus registered, assigned bus number 1
hub.c: USB hub found
hub.c: 2 ports detected
NET4: Linux TCP/IP 1.0 for NET4.0

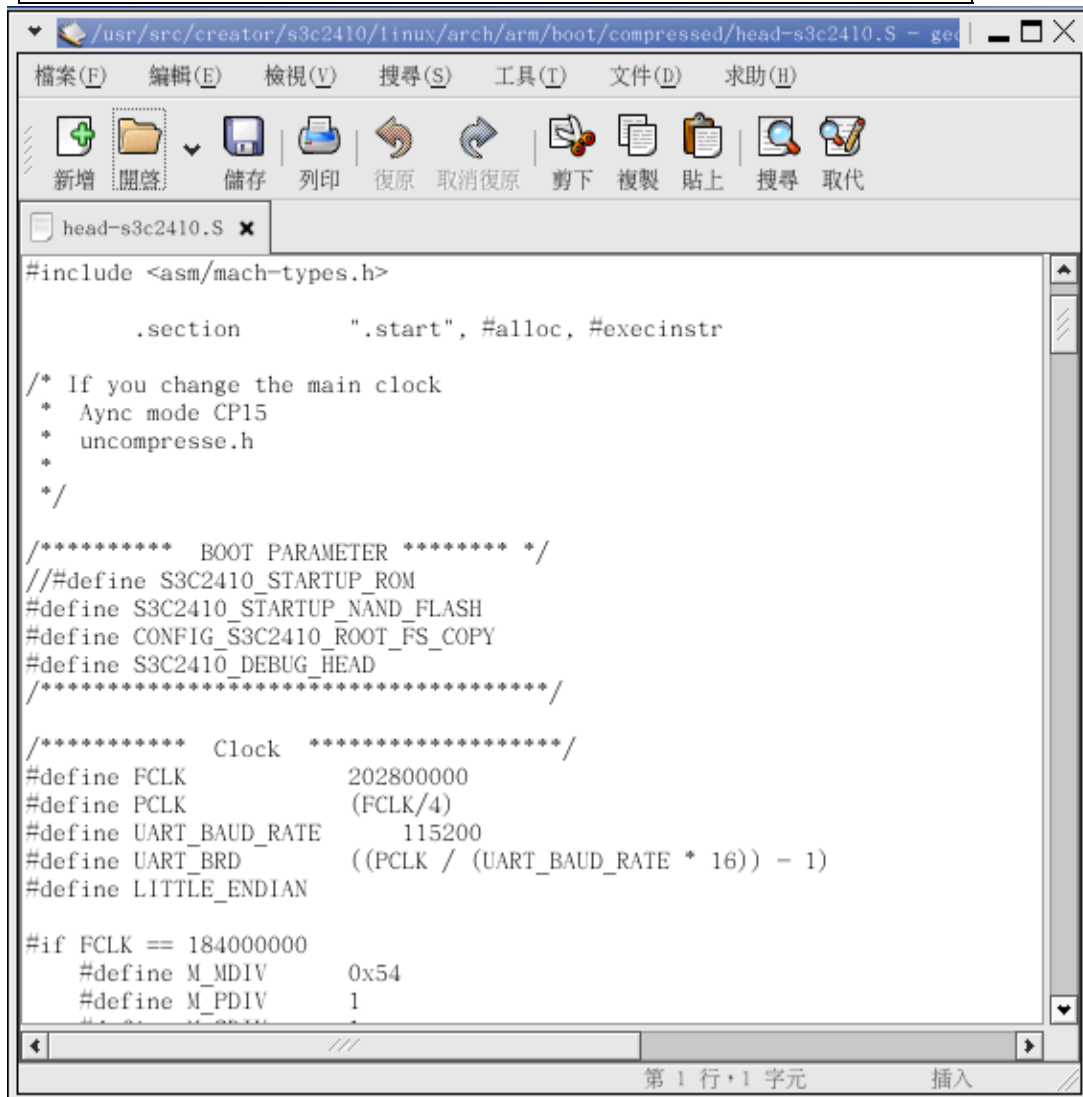
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 4096 bind 4096)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NetWinder Floating Point Emulator V0.95 (c) 1998-1999 R
RAMDISK: ext2 filesystem found at block 0
RAMDISK: Loading 8192 blocks [1 disk] into ram disk...
Freeing initrd memory: 10240K
EXT2-fs warning: mounting unchecked fs, running e2fsck :
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 48K
INIT: version 2.78 booting
INIT: Entering runlevel: 2
EXT2-fs: Unrecognized mount option
EXT2-fs: Unrecognized mount option
mount: / not mounted already, or bad option
Starting Network
bash-2.04#
```

2-5-3 將 Linux kernel image 燒錄到 NAND Flash 的驗證方式

這個部分將利用 Domingo for Linux 提供的 ProgramFlash 功能將 Linux kernel image、Root Filesystem 燒錄到 NandFlash 中，再啟動 Linux Target 端的 Embedded Linux，請依下列步驟執行：

- ◆ 首先要修改 Linux kernel source，我們將利用 Linux Host 端提供的文字編輯器編輯/usr/src/crator-s3c2410-linux/arch/arm/boot/compressed/haed-s3c2410.s 檔，首先在 Linux Host 端執行”主選單/輔助程式/文字編輯器”此時開啓”文字編輯”，請開啓目錄/usr/src/crator/s3c2410/linux/arch/arm/boot/compressed 下的 haed-s3c2410.s 檔，修改檔案如下：
將程式修改為：

```
#define S3C2410_STARTUP_NAND_FLASH
#define CONFIG_S3C2410_ROOT_FS_COPY
#define S3C2410_DEBUG_HEAD
```



```
#include <asm/mach-types.h>

.section      ".start", #alloc, #execinstr

/* If you change the main clock
 * Aync mode CP15
 * uncompress.h
 *
 */

/***** BOOT PARAMETER *****/
// #define S3C2410_STARTUP_ROM
#define S3C2410_STARTUP_NAND_FLASH
#define CONFIG_S3C2410_ROOT_FS_COPY
#define S3C2410_DEBUG_HEAD
/*****/

/***** Clock *****/
#define FCLK      202800000
#define PCLK      (FCLK/4)
#define UART_BAUD_RATE  115200
#define UART_BRD      ((PCLK / (UART_BAUD_RATE * 16)) - 1)
#define LITTLE_ENDIAN

#if FCLK == 184000000
#define M_MDIV      0x54
#define M_PDIV      1
#endif
```

- ◆ 再來要重新編譯 zImage 檔，開啓 Linux Host 端的終端機依下列執行：

```
cd /usr/src/creator/s3c2410/linux
make zImage
```

- ◆ 將編譯好的 zImage 檔複製到 samba 伺服器 export 的路徑並更改檔名為 zImage_nand，執行步驟如下：

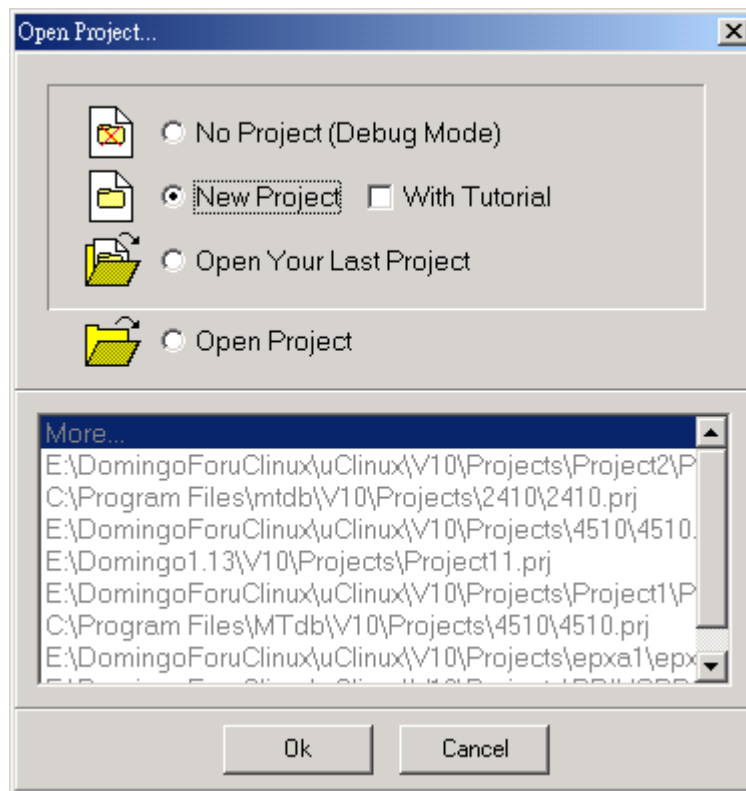
1. 請在 Linux Host 端桌面或”主選單”下執行”個人資料夾”。
2. 複製/usr/src/creator/s3c2410/linux/arch/boot 目錄下的 zImage 檔到 /usr/src/creator/nfs/s3c2410 路徑下，並修改檔名為 zImage_nand。

- ◆ Linux kernel image 編譯完成後，請回到 Windows Host 端，確認 Domingo for Linux 已安裝無誤。

- ◆ 確認 Windows Host 端與 ICE、Linux Target 端連結無誤，詳細連結方法請參考”Creator PreSOCes Development Kit User’s Guide”一書。

- ◆ 開啓 Domingo for Linux，此時出現”Open Project...”對話框，因為現在只是要做燒

錄 Linux kernel image 的動作，所以選擇”No Project(Debug Mode)”，之後按”OK”鈕。



- ◆ 再來會出現 Hardware Option...對話框，其中設定如下：

TargetMachine	選擇	WINEZ(WIN2K)ARM
Processor	選擇	S3C2410X
Memory Model	選擇	Little Endian
Non-cacheable Memory Address(Reserved 8 Bytes) 設定為 40000000		

Hardware Options...

TargetMachine:
WINaZ (WIN2K) ARM

Processor:
S3C2410X

Project Setup

Parallel interface (print port)
378

☐ PC adapted SMC super I/O

Option
TIME OUT : 256 (DEC)

Memory Model
☒ Little Endian
☐ Big Endian

Clock Select
☐ RTCK
☒ ICE Clock 20 MHz

Peripheral Configuration Manager(PCM)
☒ To Initialize PCM when Connect Config PCM...

☒ Coprocessor Option

Non-cacheable Memory Address(Reserved 8 bytes) : 40000000

Advanced Options
System Recovery Counts (6~255) : 15 (DEC)

Ok Cancel

再來設定 "Peripheral Configuration Manager(PCM)" group box 。
其中 "To Initialize PCM when Connect" CheckBox 選擇 Checked，之後按
"Config PCM" 鈕，開啓 Config Peripheral Configuration Manager 對話框，設定
PCM 檔。

Config Peripheral Configuration Manager

No	Register	Address	Value	Size
01	WTCON	53000000	0	4
02	BWSCON	48000000	22111120	4
03	BANKCON0	48000004	1700	4
04	BANKCON1	48000008	2700	4
05	BANKCON2	4800000C	700	4
06	BANKCON3	48000010	700	4
07	BANKCON4	48000014	700	4
08	BANKCON5	48000018	700	4
09	BANKCON6	4800001C	18005	4
10	BANKCON7	48000020	18005	4
11	REFRESH	48000024	8E0459	4
12	BANKSIZE	48000028	B2	4

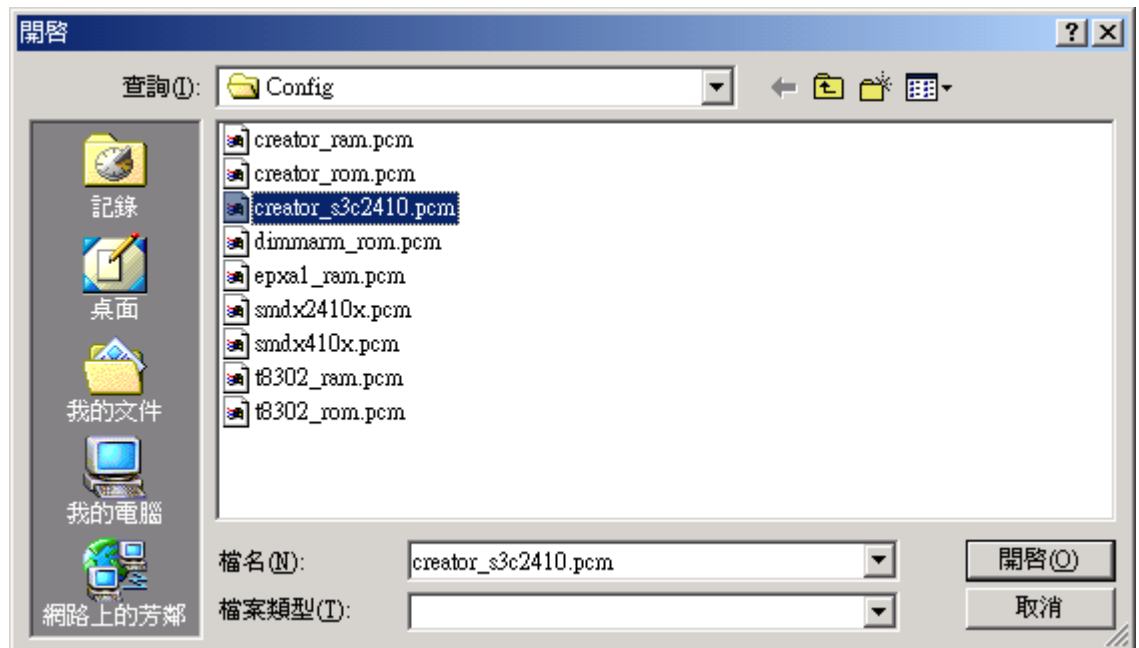
Move Up

Move Down

Append Insert Modify Delete Delete All

Import... Export... OK Cancel Apply

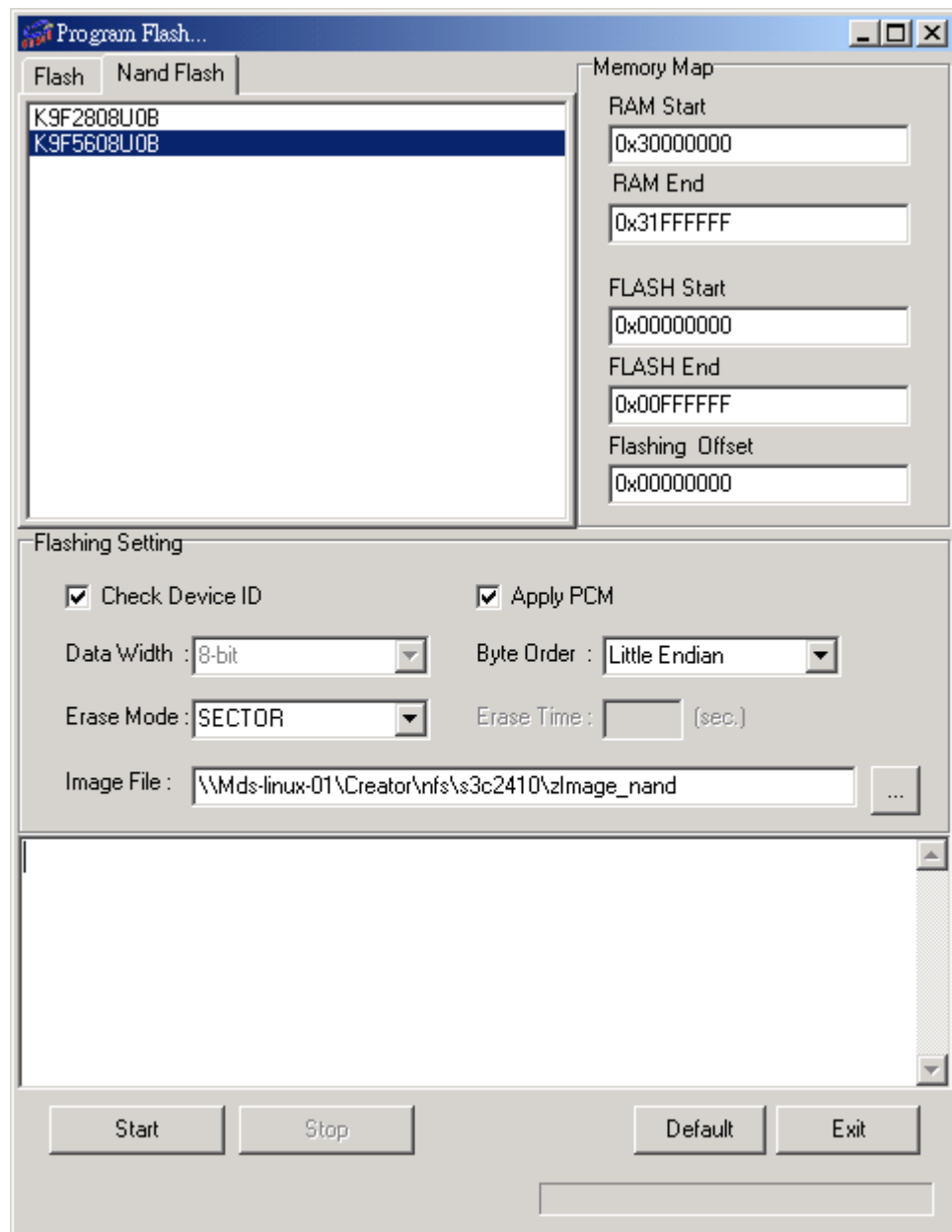
要設定 PCM 檔，請按”Import...”鈕，選擇 Domingo for Linux 安裝路徑下
\\v10\ARM\ARM7\Config\creator_s3c2410.PCM 檔即可，如下：



- ◆ 以上完成了 HardWareOptions...的設定，程式進入 Domingo for Linux 主畫面後，此時 Domingo for Linux 與 ICE 要設定為連結狀態，請選擇 Debug\Connect 使 Domingo 與 ICE 連結。
- ◆ 連結成功後將依序把 Linux kernel image 及 Root Filesystem 檔案燒錄到 Linux Target 端的 NandFlash 中，首先要把 Linux kernel image 燒錄到 NandFlash 0x0 的位置，請選擇 Domingo for Linux 中 menu 的”\ProgramFlash”開啓 ProgramFlash 對話框並設定如下：
 1. page : NandFlash
 2. Select Device : K9F2808U0B (NAND Flash 之型號依照 Creator-S3C2410 子板上之 U11 實際使用的 IC 為依據)
 3. Memory Map :

RAM Start	: 0x30000000	RAM End	: 0x31FFFFFF
FLASH Start	: 0x00000000	FALSH End	: 0x00FFFFFF
Flashing Offset	: 0x00000000		
 4. Flashing Setting :

Check Device ID	: checked
Apply PCM	: checked
 5. 燒錄檔案：請選擇網路上的芳鄰中的 Microtime 群組下的 \\Mds-linux-01\Ccreator\nfs\s3c2410\zImage_nand 檔。(假設 Linux Host 端因 smaba 伺服器設定在”Microtime 群組下，電腦名稱爲”Mds-linux-01”)



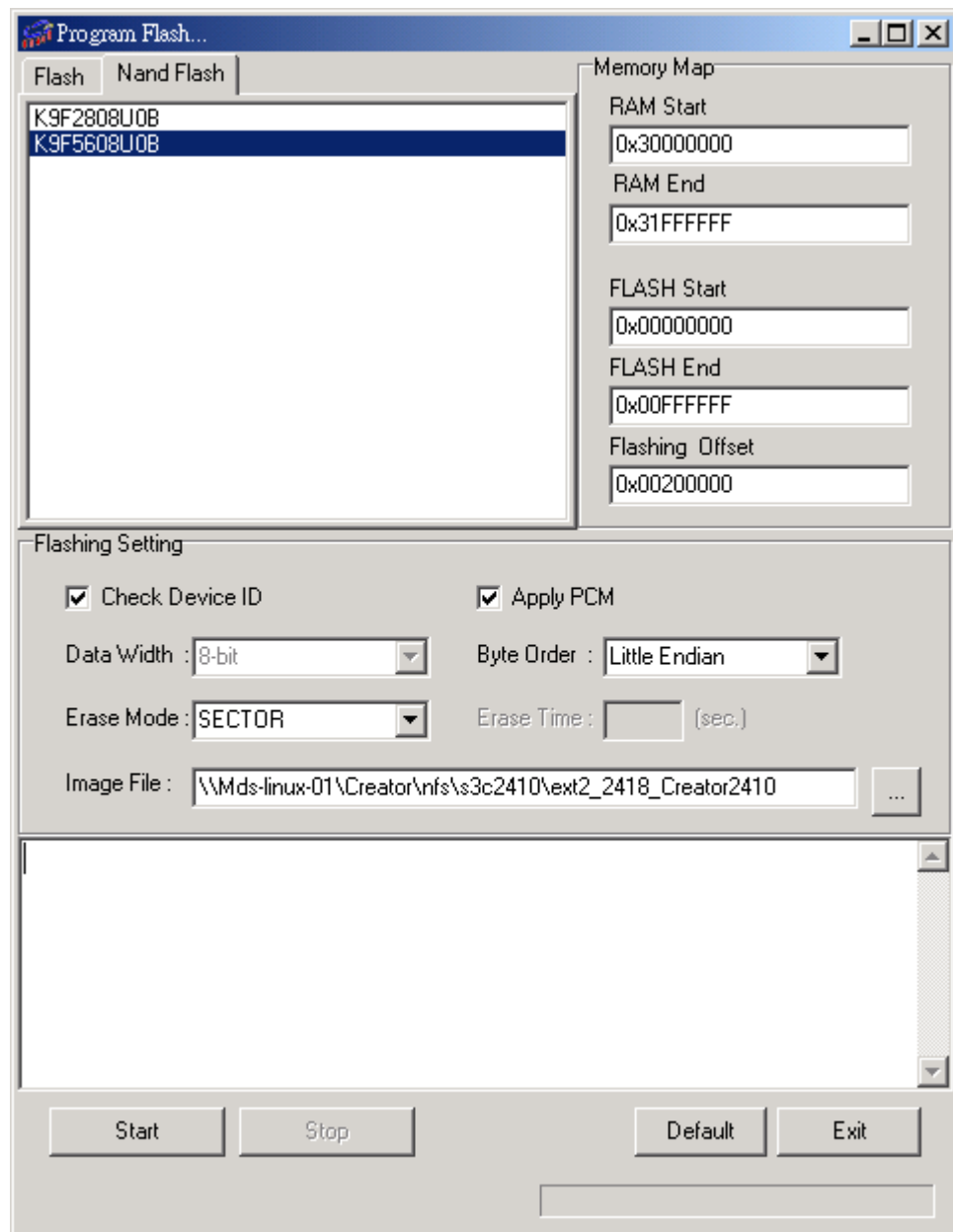
◆ 接下來要把 Root Filesystem 燒錄到 Linux Target 端的 NandFlash 0x00200000 的位置，請選擇 Domingo for Linux 中 menu 的”\ProgramFlash”開啓 ProgramFlash 對話框並設定如下：

1. page : NandFlash
2. Select Device : K9F2808U0B (NAND Flash 之型號依照 Creator-S3C2410 子板上之 U11 實際使用的 IC 為依據)
3. Memory Map :

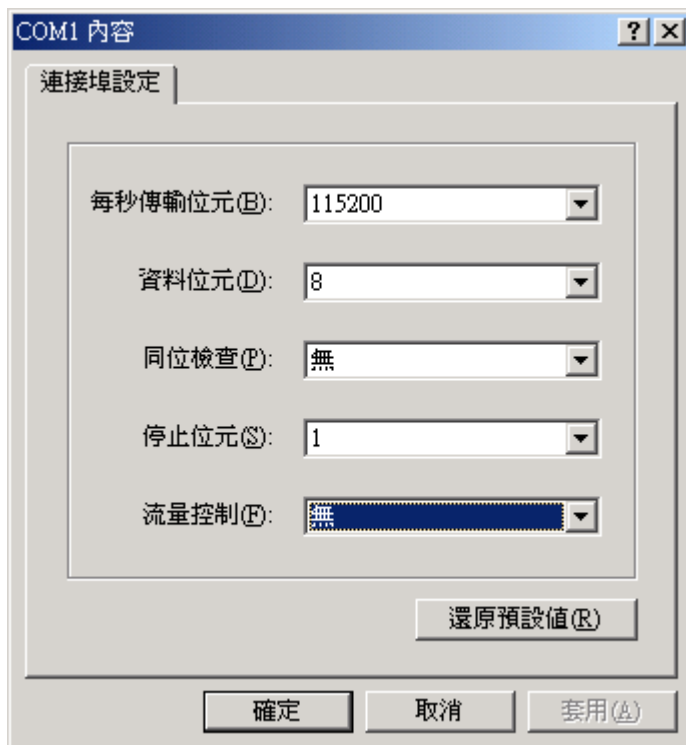
RAM Start	: 0x30000000	RAM End	: 0x31FFFFFF
FLASH Start	: 0x00000000	FLASH End	: 0x00FFFFFF
Flashing Offset	: 0x00200000		
4. Flashing Setting

Check Device ID	: checked
Apply PCM	: checked
5. 燒錄檔案：請選擇網路上的芳鄰中的 Microtime 群組下的

\\Mds-linux-01\Creator\nfs\s3c2410\ext2_2418_Creator2410 檔。(假設 Linux Host 端因 smaba 伺服器設定在”Microtime 群組下，電腦名稱爲”Mds-linux-01”)



- ◆ 設定 Windows Host 端的超級終端機，以顯示 Linux Target 端的 Embedded Linux 啓動資訊，超級終端機的設定如下：



- ◆ 現在已經把 Linux kernel image、Root Filesystem 燒錄到 Linux Target 端的 NandFlash 中，所以要改變 Linux Target 端 Creator-S3C2410 子板的 pin 腳，改成由 NandFlash 啟動 Embedded Linux。
請先關閉 Linux Target 端的電源，把 Creator-S3C2410 子板的 J19 pin 腳改接成連結 2、3 腳位，並把 J3 pin 腳短路，現在開啓 Linux Target 端的電源，便會自動啟動 Embedded Linux，其開啓的相關資訊也會在 Windows Host 端的超級終端機中顯示，如下：

115200 - 超級終端機

檔案(F) 編輯(E) 檢視(V) 呼叫(C) 轉送(T) 說明(H)

S3C2410 UDA1341 / IIS initialized
usb.c: registered new driver hub
usb-ohci.c: USB OHCI at membase 0xd9000000, IRQ 26
usb.c: new USB bus registered, assigned bus number 1
hub.c: USB hub found
hub.c: 2 ports detected
NET4: Linux TCP/IP 1.0 for NET4.0

IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 4096 bind 4096)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NetWinder Floating Point Emulator V0.95 (c) 1998-1999 Red Hat
RAMDISK: ext2 filesystem found at block 0
RAMDISK: Loading 8192 blocks [1 disk] into ram disk... done
Freeing initrd memory: 10240K
EXT2-fs warning: mounting unchecked fs, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 48K
INIT: version 2.78 booting
INIT: Entering runlevel: 2
EXT2-fs: Unrecognized mount option
EXT2-fs: Unrecognized mount option
mount: / not mounted already, or bad option
Starting Network
bash-2.04#

已中斷連線 自動偵測 115200 8-N-1 SCROLL CAPS NUM 顯示 列印

第三章 如何修改 Creator-S3C2410 的 Kernel Address 和 Clock Parameter

這個章節將介紹 Linux kernel address 及 Clock Parameter 各個相關設定值在哪些檔案中定義。

修改 Linux kernel address，如下：

Name	FILE NAME	Value (default)
ZTEXTADDR	arch/arm/boot/Makefile	0x30F00000
ZRELADDR	arch/arm/boot/Makefile	0x30008000
RAMDISK_DN_ADDR	include/asm-arm/arch-s3c2410/s3c2410.h arch/arm/arch-s3c2410/arch.c	0x30800000
ZIP_RAMDISK_SIZE	include/asm-arm/arch-s3c2410/s3c2410.h arch/arm/arch-s3c2410/arch.c	4*1024*1024
MEM_SIZE	include/asm-arm/arch-s3c2410/s3c2410.h	64*1024*1024
__Distance_PA_VA	include/asm-arm/arch-s3c2410/memory.h	0x90000000 used by __virt_to_bus and __bus_to_virt and if you want to change the address mapping from physical to virtual, CHANGE this value.

DESCRIPTION	FILE NAME	Value (default)
Mapping from Physical address to Virtual Address	arch/arm/mach-s3c2410/mm.c	include/asm-arm/arch- s3c2410/s3c2410.h
ZRELADDR	arch/arm/boot/Makefile	0x30008000
RAMDISK_DN_ADDR	include/asm-arm/arch-s3c2410/s3c2410.h arch/arm/arch-s3c2410/arch.c	0x30800000
ZIP_RAMDISK_SIZE	include/asm-arm/arch-s3c2410/s3c2410.h arch/arm/arch-s3c2410/arch.c	4*1024*1024

修改 Clock Parameter，如下：

DESCRIPTION	FILE NAME	Value (default)
CLKDIVN MPLLCON UPLLCON	arch/arm/boot/compressed/head- s3c2410.S	
FCLK HCLK PCLK UCLK BUSWIDTH	include/asm-arm/arch- s3c2410/s3c2410.h	FCLK= 202.8Mhz HCLK = FCLK/2 PCLK = HCLK/2 UCLK=48Mhz BUSWIDTH=32
CURRENT_BAUD_RATE	include/asm-arm/arch- s3c2410/s3c2410.h	115200 UART 0 Fixed.

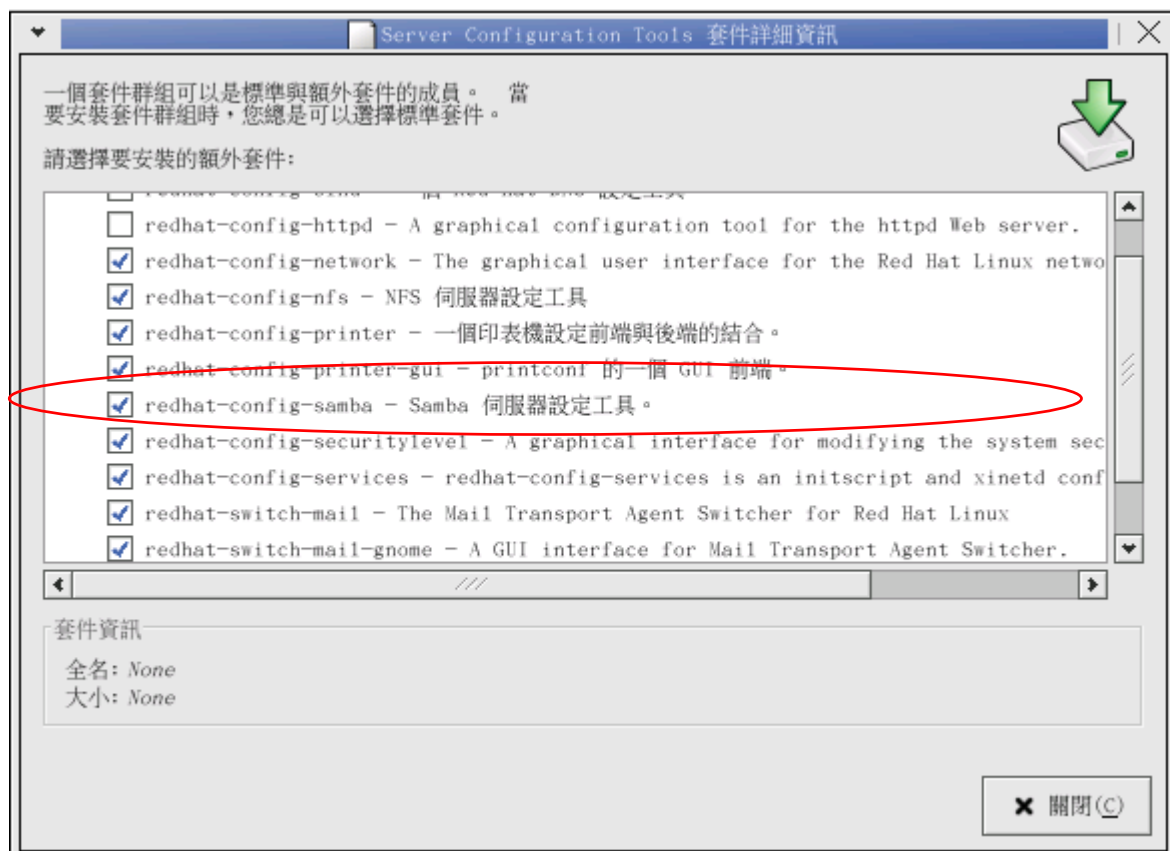
附錄 A Samba 伺服器的安裝與設定

Samba 伺服器可以讓我們在 Linux Host 端的檔案，在不同的作業平台分享，所以在我們的驗證過程中也將利用 samba 伺服器，讓我們可以在 Windows Host 端讀取 Linux Host 端的檔案。

以下我們將介紹如何在 Linux Host 端建立 Samba 伺服器。操作步驟如下：

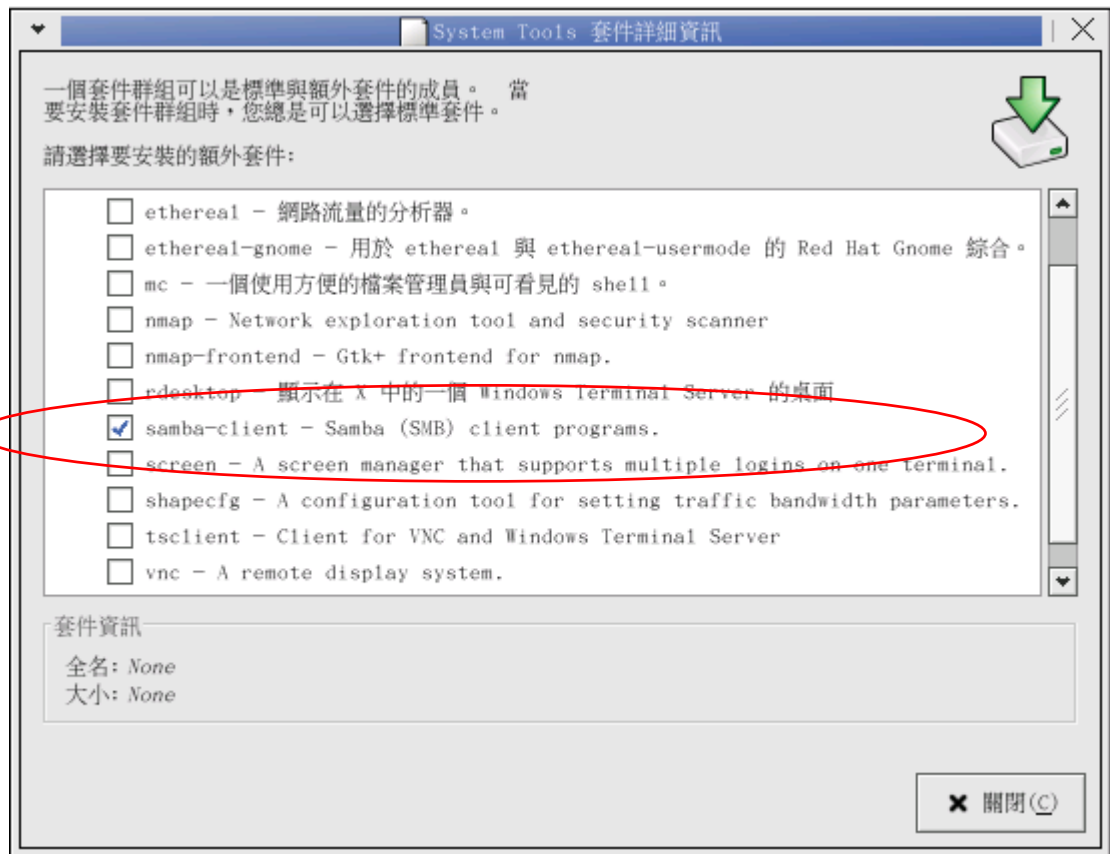
- ◆ 檢查 Linux Host 端 samba 伺服器套件是否安裝，請在 Linux Host 端的”主選單”下選擇”系統設定”的”新增移除應用程式”，在新增移除應用程式對話框中做以下設定：

1. 選擇伺服器→伺服器設定工具→redhat-config-samba - Samba 伺服器設定工具



選擇完成後，請按更新鈕。

2. 選擇系統→系統工具→samba-client - Samba (SMB) client program

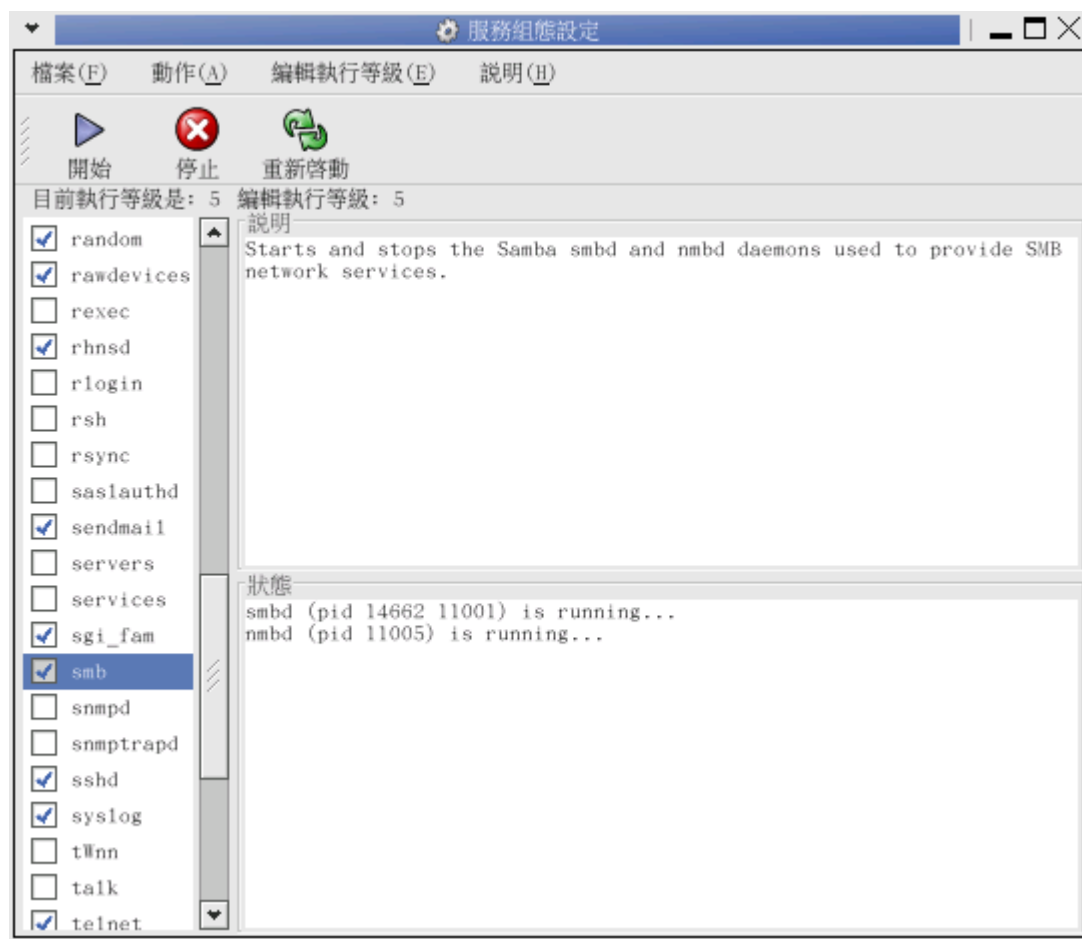


選擇完成後，請按更新鈕。

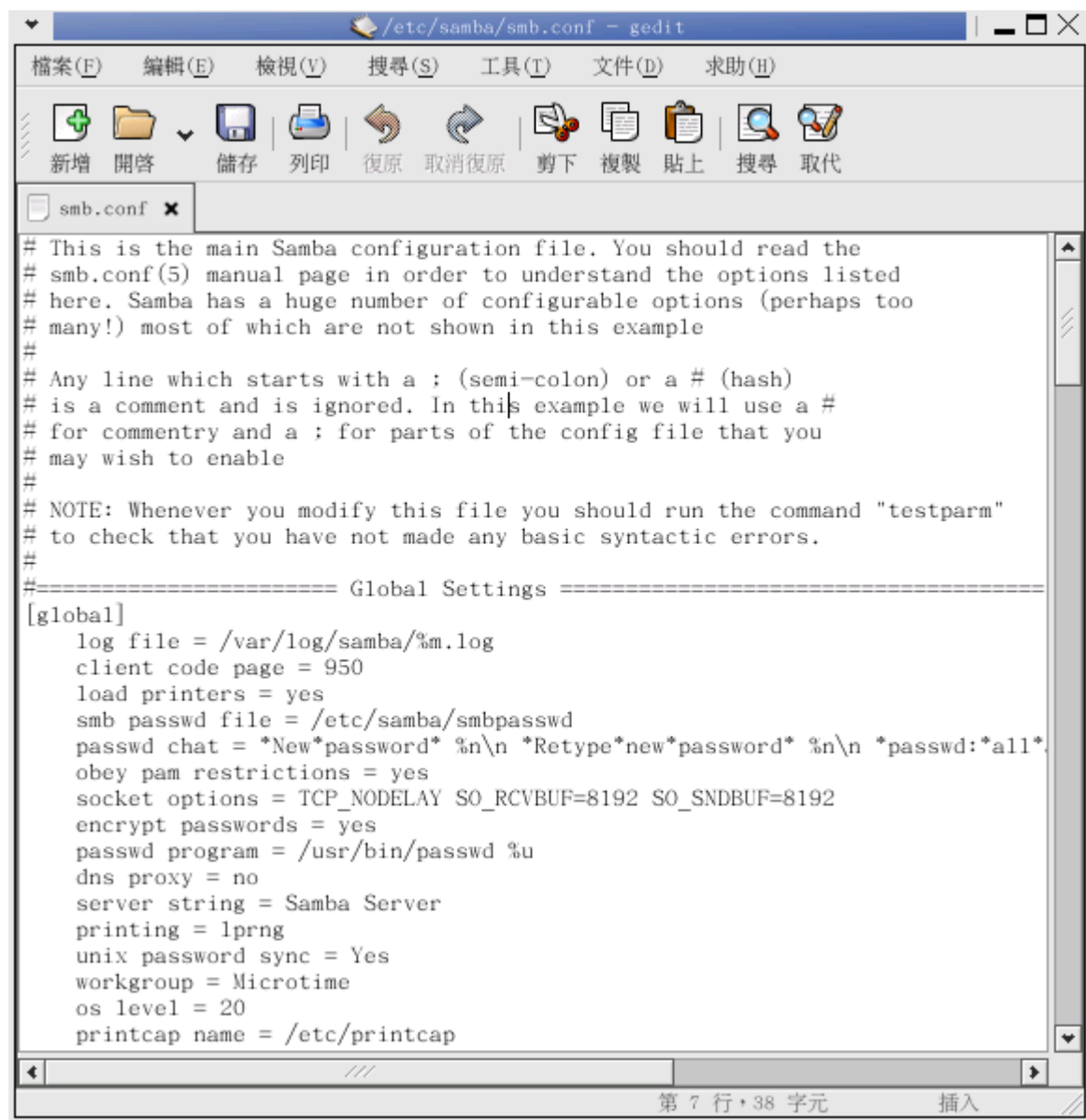
- ◆ 建立 Linux Host 端共享的資源目錄路徑，請在 Linux Host 端的終端機執行如下：

```
mkdir /usr/src/creator/nfs/s3c2410 -p
```

- ◆ 要啟動 Samba 伺服器，請在 Linux Host 端執行”主選單/系統設定/服務”此時開啓”服務組態設定”對話框，請選擇”smb”為 checked，並在 toolbar 選按”開始”，則開啓 Samba 伺服器，如下：



- ◆ 再來要編輯 Samba 伺服器的設定，我們將利用 Linux Host 端提供的文字編輯器編輯 /etc/samba/smb.conf 檔，首先在 Linux Host 端執行”主選單/輔助程式/文字編輯器”此時開啓”文字編輯器”，請開啓目錄/etc/samba 下的 smb.conf 檔，如下：



```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps too
# many!) most of which are not shown in this example
#
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentry and a ; for parts of the config file that you
# may wish to enable
#
# NOTE: Whenever you modify this file you should run the command "testparm"
# to check that you have not made any basic syntactic errors.
#
===== Global Settings =====
[global]
    log file = /var/log/samba/%m.log
    client code page = 950
    load printers = yes
    smb passwd file = /etc/samba/smbpasswd
    passwd chat = *New*password* %n\n *Retye*new*password* %n\n *passwd:*all*
    obey pam restrictions = yes
    socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
    encrypt passwords = yes
    passwd program = /usr/bin/passwd %u
    dns proxy = no
    server string = Samba Server
    printing = lprng
    unix password sync = Yes
    workgroup = Microtime
    os level = 20
    printcap name = /etc/printcap
```

◆ 在/etc/samba/smb.conf 檔案中，我們要修改幾個項目，請依下列步驟執行如下：

1. 設定群組名稱：

將[global]下的 workgroup 改成使用者網路芳鄰中的工作群組，例如筆者網路芳鄰中的工作群組為”Microtime”，所以改成如下：

```
workgroup = Microtime
```

2. 設定伺服器安全等級：

修改[global]下的 security，將可設定 samba 伺服器的安全等級，我們設定為”share”，表示每一個人都可以分享，且不需要帳號和密碼，如下：

```
security = share
```

3. 使用者資源共享設定：

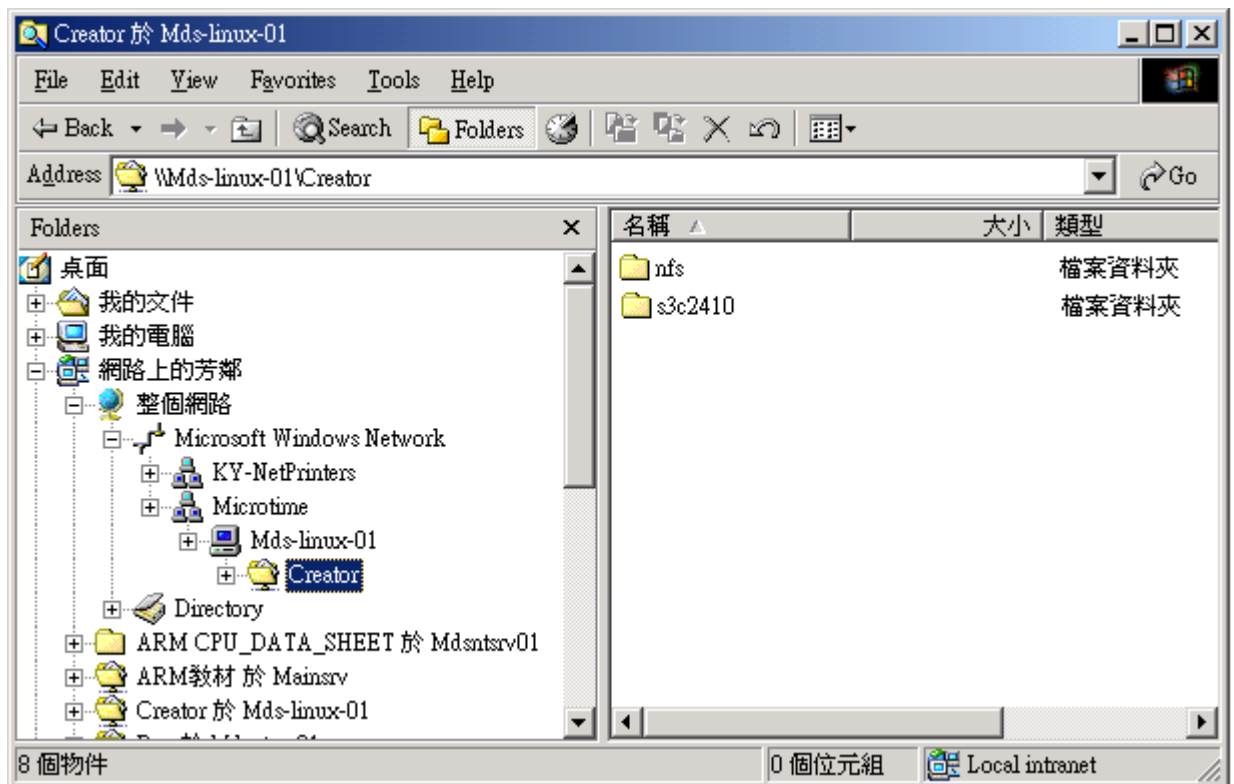
我們將設定在 linux 上共享的資源目錄為 Creator，完整路徑為/usr/src/creator/nfs，所以請在 smb.conf 檔案的最後加入如下：

```
[Creator]
    copy = tmp
```

```
guest ok = yes
path = /usr/src/creator/nfs
```

以上表示，[Creator]目錄的屬性是複製於[tmp]，而其完整的路徑是 /usr/src/creator/nfs。

- ◆ 最後請在 Linux Host 端執行”主選單/系統設定/服務”再度開啓”服務組態設定”對話框，一樣選擇”smb”為 checked，並在 toolbar 選按”停止”，在按”重新啓動”以確保剛剛設定被正確執行。
- ◆ 現在觀察 Windows Host 端檢查剛剛的設定是否無誤，從上列的設定可以知道 Linux Host 端是設定在”Microtime”的網路群組之下，假設 Linux Host 端的電腦名稱是”Mds-linux-01”，所以現在開啓 Windows Host 端的檔案總管，觀察網路上的芳鄰的 Microtime 群組下可以看到代表 Linux Host 端的電腦 Mds-linux-01，如下：



附錄 B Root Filesystem 的建立與修改

Linux 的最重要的特徵之一是它的為許多不同的檔案系統的支援。這使其非常靈活從而與許多另外的作業系統可以很好的共存。在本書寫作的時候，Linux 已支援 15 種檔案系統: ext , ext2 , xia , minix , umsdos , msdos , vfat , proc , smb , ncp , iso9660 , sysv , hpfs , affs 及 ufs , 並且沒有疑問，將來支援的檔案類型將被增加的更多。

它們被構建成為一個單一的層次樹狀結構以作為代表檔案系統的實體。Linux 通過安裝一個檔案系統將該新檔案系統加入它的檔案系統樹中。所有的檔案系統，不管是什麼類型，都安裝在檔案系統樹的一個目錄上並且該檔案系統之上的檔案將掩蓋掉這個安裝目錄中原來存在的內容。這個目錄稱為安裝目錄或安裝點。當檔案系統被卸掉之後，安裝目錄中原來的檔案才再次可見。

檔案系統是建置於儲存設備的分割區之上，用以儲存檔案資料。一個檔案系統在使用之前必須先「掛上 (mount)」，將它掛在系統樹狀目錄結構的某個點上。同理，當某個分割區的檔案系統不使用時，可以將它「卸下 (umount)」，如此系統會將尚未寫入的資料寫回去，並將它從系統的樹狀目錄中除去。

我們之前說過 Linux 啟動時包含 Boot Loader、Root Filesystem、Linux kernel image 這三個部分，其中 Root Filesystem 就是 Linux 啟動時第一個掛載的檔案系統，掛載的目錄是在根目錄下。在 Embedded Linux 系統的開發過程中，如何建立及修改 Root Filesystem 是非常重要的，然而要重新編譯一個 Root Filesystem 卻是一件巨大的工程，所以在此建議使用者如果要開發、修改 Root Filesystem，最好找一個已經編譯好且適用的檔案系統，然後再依照自己的系統架作修改，這樣可以大大減少開發的時間及難度。

ext2 及 jffs2 的檔案系統是在 Embedded 系統中較常使用的檔案系統，所以在這個章節中我們將依 ext2 及都 jffs2 為範例，介紹如何建立一個檔案系統大小、包含的應用程式及資料適合自己 Embedded 系統的 Root Filesystem。

B-1 建立 ext2 file system

Ext2 是 GNU/Linux 系統中標準的檔案系統，其特點為存取檔案的效能極好，對於中小型的檔案尤佳，這主要得利於其資料區塊快取層的優良設計。其單一檔案大小與檔案系統本身的容量上限與檔案系統本身的資料區塊大小有關，在一般常見的 x86 電腦系統中，資料區塊最大為 4KB，則單一檔案大小上限為 2048GB，而檔案系統的容量上限為 16384GB。但由於目前核心 2.4 所能使用的單一分割區最大只有 2048GB，因此實際上能使用的檔案系統容量最多也只有 2048GB。

以下介紹如何建立一個 ext2 的 Root Filesystem，請 Linux Host 端開啓終端機，執行下列步驟：

- ◆ 建立一個空的 ext2 檔案系統：我們將建立一個檔名為 ext2new 的檔案，且其內容共有 8604KB 左右大小的檔案，如下：

```
dd if=/dev/zero of=ext2new bs=1k count=8604
```

其中 dd 這個指令是用來轉換檔案並且 copy 用的；

if 指的是要被轉換的輸入檔案格式 /dev/zero 可以由 man zero 來查看內容；

of 指的是輸出的檔案，我們將之輸出到 ext2new 這個檔案；

bs 指的是一個磁區佔用幾個 kb ；

count 指的是要使用多少個 bs ，所以最後的容量為 $bs * count = 1k * 8604$

```
mke2fs -F -m0 -i 2000 ext2new
```

這是用來將磁碟格式化成 Linux 系統檔的指令。所以將 ext2new 檔案格式化成 ext2 格式，並設定其 node 數為 2000。

- ◆ 上一個步驟已經建立一個空的 ext2 檔案系統，現在掛載此檔案系統，如下：

```
mount -w -o loop ext2new /mnt/loop
```

將 ext2new 檔案系統掛載到/mnt/loop 目錄下，其中“-w”參數是設定其為 read/write 模式，若設為“-r”則為 read only 模式。

- ◆ 上一個步驟已經將 ext2new 檔案系統掛載起來，現在將需要檔案加入/mnt/loop 中，之前我提及可以利用以個編譯好且合適的檔案系統再加以修改成適合自己的 Embedded Linuc 系統，所以假設有一個 ext2org 檔案系統，我們將把此檔案系統下所有檔案資料複製到 ext2new 中，如下：

```
mount -w -o loop ext2org /mnt/looporg
```

將 ext2org 檔案系統掛載到/mnt/looporg 目錄下。

```
cp -dpR /mnt/looporg/. /mnt/loop/.
```

將/mnt/looporg 目錄下所有檔案複製到/mnt/loop 目錄下。

- ◆ 假設檔案系統已經修改完畢後，就使用 umount 將檔案系統卸載，如下：

```
umount -l /mnt/loop
```

umount 時會把 mount 時 file system 修改的資料寫回 image 中。

- ◆ 現在 ext2new 檔案內容已經和 ext2org 一模一樣了，使用者只要再依其所需修改即可，修改的方式跟上述步驟是一樣的，先把要修改的 Root Filesystem mount 起來，根據所需增加或刪除 Root Filesystem 的檔案，最後再將修改過的 Root Filesystem umount 回原來的 image 檔就可以了，其中要注意的是加入的檔案其編譯的版本跟 Root Filesystem 編譯的版本是要一致的。

B-2 建立 jffs2 file system

此為日誌式快閃 (Flash) 檔案系統，用於嵌入式裝置。

jffs2 有以下優點：

- a. JFFS2 在磁區級別上執行閃存擦除／寫／讀操作要比 Ext2 文件系統好。
- b. JFFS2 提供了比 Ext2fs 更好的崩潰／掉電安全保護。當需要更改少量

數據時，Ext2 文件系統將整個磁區複製到記憶體（DRAM）中，在記憶體中合併新數據，並寫回整個磁區。這意味著為了更改單個字，必須對整個磁區（64 KB）執行讀／擦除／寫例程 — 這樣做的效率非常低。要是運氣差，當正在 DRAM 中合併數據時，發生了電源故障或其他事故，那麼將丟失整個數據集合，因為在將數據讀入 DRAM 後就擦除了閃存磁區。JFFS2 附加文件而不是重寫整個磁區，並且具有崩潰／掉電安全保護這一功能。

- c. 這可能是最重要的一點：JFFS2 是專門為象閃存晶片那樣的嵌入式設備創建的，所以它的整個設計提供了更好的閃存管理。

jffs2 在嵌入式系統中有以下缺點：

1. 當文件系統已滿或接近滿時，JFFS2 會大大放慢運行速度。

安裝 mkfs.jff2：

- ◆ 首先檢查有沒有製作 jff2 的工具，如果沒有要先安裝，請檢查/sbin 下有無 mkfs.jffs2 執行檔，若無需先安裝，且依下列步驟執行。
- ◆ 使用 cvs 下載 source code，請開啓 Linux Host 端終端機執行如下：

```
cvcs -d :pserver:anoncvcs@cvs.infradead.org:/home/cvs login (password: anoncvcs)
cvcs -d :pserver:anoncvcs@cvs.infradead.org:/home/cvs co mtd
```

- ◆ 此時 cvs 會主動在目前目錄下建立 mtd 的目錄，且在目錄下會產生 mkfs.jffs2，請將其複製到/sbin 目錄下，如下：

```
cp /mtd/mkfs.jffs2 /sbin
```

建立 mkfs.jff2 檔案系統：

- ◆ 首先把” C-1 建立 ext2 file system”建立的 ext2 file system 掛載，如下：

```
mount -w -o loop ext2new /mnt/loop
```

- ◆ 之後可以執行 ”mkfs.jffs2 -d 輸入目錄名稱 -o 輸出目錄名稱”來建立 jffs2 檔，如下：

```
mkfs.jffs2 -r loop -o root.jffs2
```

其中 loop 為輸入目錄，root.jffs2 為輸出檔案名稱，這個步驟就是將 loop 目錄下檔案系統製作成 jffs2 檔案系統的 image 檔，檔案名稱是 root.jffs2。