Building Linux Kernel

Outline

- Linux核心程式碼的目錄結構及各目錄的相關內容
- Linux核心各設定選項內容和作用
- Linux核心設定檔config.in的作用
- Linux核心的編譯過程
- 將新增核心程式加入到Linux核心結構中的方法

核心程式碼目錄介紹 (1)

- **arch**：arch子目錄包括所有與體系結構相關的核心程式。arch的每一個子目錄都代表一個Linux所支援的體系結構。例如：arm目錄下就是arm體系架構的處理器目錄，包含我們使用的PXA處理器。

- **include**：include子目錄包括編譯核心所需要的標頭檔。與ARM相關的標頭檔在include/asm-arm子目錄下。

- **init**：這個目錄包含核心的初始化程式，但不是系統的引導程式，其中所包含main.c和Version.c檔是研究Linux核心的起點。

- **mm**：該目錄包含所有獨立於CPU體系結構的記憶體管理程式，如頁式儲存管理記憶體的分配和釋放等。與ARM體系結構相關的程式在arch/arm/mm中。

- **Kernel**：這裡包括主要的核心程式，此目錄下的檔實現大多數Linux的核心函數，其中最重要的檔案是sched.c。與Xscale體系結構相關的程式在arch/arm-pxa/kernel目錄中。

核心程式碼目錄介紹 (2)

- **Drives**：此目錄存放系統所有的設備驅動程式，每種驅動程式各占一個子目錄。
  - /block：區塊設備驅動程式。區塊設備包括IDE和scsi設備。
  - /char：字元設備驅動程式。如串列埠、滑鼠等。
  - /cdrom：包含Linux所有的CD-ROM程式。
  - /pci：PCI卡驅動程式程式，包含PCI子系統映射和初始化程式等。
  - /scsi：包含所有的SCSI程式以及Linux所支援的所有的SCSI設備驅動程式程式。
  - /net：網路設備驅動程式。
  - /sound：音效卡設備驅動程式。
- **lib**目錄放置核心的函式庫程式；
- **net**目錄包含核心與網路的相關的程式；
- **ipc**目錄包含核心行程通訊的程式；
- **fs**目錄是所有的檔案系統程式和各種類型的檔案操作程式，它的每一個子目錄支援一個檔案系統，如JFFS2；
- **scripts**目錄包含用於設定核心的腳本檔案等。每個目錄下一般都有depend檔和一個makefile檔，他們是編譯時使用的輔助檔，仔細閱讀這兩個檔案對弄清各個檔案之間的相互依託關係很有幫助。

核心的設定的基本結構

- **Makefile**：分佈在Linux核心程式中的Makefile，定義Linux核心的編譯規則；頂層Makefile是整個核心設定、編譯的整體控制檔案；
- **設定檔(config.in)**：給使用者提供設定選擇的功能；.config：核心設定檔，包括由使用者選擇的設定選項，用來存放核心設定後的結果；
- **設定工具**：包括對設定腳本中使用的設定命令進行解釋的設定命令解釋器和設定使用者介面（基於字元介面：make config；基於Ncurses圖形介面：make menuconfig；基於xWindows圖形介面：make xconfig）
- **Rules.make**：規則檔，被所有的Makefile使用。

編譯規則Makefile

- 利用 make menuconfig（或make config、make xconfig）對linux核心進行設定後，系統將產生設定檔（.config）。在編譯時，頂層 Makefile 將讀取 .config 中的設定選擇。
- 頂層 Makefile完成產生核心檔（vmlinux ）和核心模組（module）兩個任務，爲了達到此目的，頂層 Makefile 遞迴進入到核心的各個子目錄中，分別調用位於這些子目錄中的 Makefile，然後進行編譯。至於到底進入哪些子目錄，取決於核心的設定。
- 頂層Makefile中的include arch/$(ARCH)/Makefile指定特定 CPU 體系結構下的 Makefile，這個Makefile包含了特定平台相關的資訊。

設定檔config.in

- Linux所有選項設定都需要在 config.in 檔中用設定語言來編寫設定腳本，然後頂層 Makefile 調用 scripts/Configure，按照 arch/arm/config.in 來進行設定。
- 命令執行完後產生儲存有設定資訊的設定檔（.config）。下一次再做 make config 時將產生新的 .config 檔案，原.config 被改名爲.config.old。

編譯核心的常用命令

- **Make config**：核心設定，調用./scripts/Configure按照arch/i386/config.in來進行設定。命令執行後產生檔.config，其中儲存著設定資訊。下次在做make config時將產生新的.config檔案，原檔案config更名爲config.old
- **make dep**：尋找依存關係。產生兩個檔. depend和.hdepend，其中.hdepend表示每個.h檔都包含其他哪些嵌入檔。而.depend檔有多個，在每個會產生目標檔(.o)檔的目錄下均有，它表示每個目標檔都依賴於哪些嵌入檔(.h)
- **make clean**：清除以前建構核心所產生的所有的目標檔，模組檔，核心以及一些暫存檔案等，不產生任何檔
- **make rmproper**：刪除所有以前在建構核心過程所產生的所有檔，及除了做make clean 外，還要刪除.config，.depend等檔，把核心程式恢復到最原始的狀態。下次建構核心時必須進行重新設定；
- **make, make zImage, make bzImage**：
  - **make**：建構核心。通過各目錄的Makefile檔進行，會在各個目錄下產生一大堆目標檔，如核心程式沒有錯誤，將產生檔vmlinux，這就是所建構的核心。並產生映射檔system.map通過各目錄的makefile檔進行。.version檔中的數加1，表示版本號的變化。
  - **make zImage**：在make的基礎上產生壓縮的核心映射檔./arch/$(ARCH)/boot/zImage以及./arch/$(ARCH)/boot/compressed目錄下產生一些暫存檔案。
  - **make bzImage**：在make的基礎上產生壓縮比例更大的的核心映射檔./arch/$(ARCH)/boot/bzImage以及./arch/$(ARCH)/boot/compressed目錄下產生一些暫存檔案。在核心太大時進行

核心編譯過程

- make mrproper：刪除所有以前在構核過程所產生的所有檔
- make menuconfig：核心設定
- make dep：尋找依存關係
- make zImage：產生壓縮的核心映射檔
  - 核心編譯完畢之後，產生zImage核心映像檔儲存在程式碼的arch/arm/boot/目錄下

Main Menu


root@esl24:~/XScale/Linux/Kernel/linux-2.4.21-51Board_EDR
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
Linux Kernel v2.4.21-rmk1-pxa1-xsbase270 Configuration
Main Menu
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
Code maturity level options --->
Loadable module support --->
System Type --->
General setup --->
Parallel port support --->
Memory Technology Devices (MTD) --->
Plug and Play configuration --->
Block devices --->
Multi-device support (RAID and LVM) --->
Networking options --->
<Select> < Exit > < Help >
ModuleTest.tar [唯讀]
檔案(F) 編輯(E) 顯示(V) 求助(H)
[root@esl24 source]#
[root@51Board ~]# mkdir make_menuconfig
[root@51Board ~]#
ModuleTest.pro
18 個檔案 (124.1 KB)
3 個檔案
完成
File ···
root···
root···
root···
GoAh···
hello.···
Modu···
Modu···

Code maturity level options

System Type


root@esl24:~/XScale/Linux/Kernel/linux-2.4.21-51Board_EDR
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
Linux Kernel v2.4.21-rmk1-pxa1-xsbase270 Configuration
System Type
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
(PXA270/250/210-based) ARM system type
Archimedes/A5000 Implementations --->
Footbridge Implementations --->
SA11x0 Implementations --->
AT91RM9200 Implementations --->
Intel PXA270/250/210 Implementations --->
CLPS711X/EP721X Implementations --->
--- Processor Type
[*] Workaround for XScale cache errata (see help)
--- Processor Features
<Select> < Exit > < Help >
ModuleTest.tar [唯讀]
[root@51Board ~]# mkdir make_menuconfig
[root@51Board ~]#

General setup

Memory Technology Devices (MTD)

RAM/ROM/Flash chip drivers

Mapping drivers for chip access

Block devices

Networking options

Network device support

Character devices

File systems

Frame-buffer support

新增驅動程式到**linux**核心

- 假設將xsbase 驅動儲存到linux程式碼的 drivers/xsbase/ 目錄下：

  $ cd drivers/xsbase

  $ tree .

  |-- Config.in

  |-- Makefile

  |-- test.c

  `-- test_client.c

編輯設定檔

```
#
# XSBASE driver configuration
#
mainmenu_option next_comment
comment ‘XSBASE Driver'
bool 'XSBASE support' CONFIG_XSBASE
if [ "$CONFIG_XSBASE" = "y" ]; then
  tristate 'TEST user-space interface'
    CONFIG_TEST_USER
 bool 'TEST CPU ' CONFIG_TEST_CPU
fi
endmenu
```

Sound


應用程式 位置 桌面
10月31日(三) 11:21
ModuleTest.tar [唯讀]
檔案(F) 編輯(E) 顯示(V) 求助(H)
root@esl24:~/XScale/Linux/Kernel/linux-2.4.21-51Board_EDR
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
Linux Kernel v2.4.21-rmk1-pxa1-xsbase270 Configuration
Sound
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
<*> Sound support
< > Support for Turtle Beach MultiSound Classic, Tahiti, Monterey
< > Support for Turtle Beach MultiSound Pinnacle, Fiji
< > OSS sound modules
<*> Intel PXA27x AC97 audio
<Select> < Exit > < Help >
[root@esl24 source]#
[root@51Board ~]# mkdir make_menuconfig
[root@51Board ~]#
ModuleTest.pro
18 個檔案 (124.1 KB)
3 個檔案
完成
File ···
root···
root···
root···
GoAh···
hello.···
Modu···
Modu···

New Driver


root@esl24:~/XScale/Linux/Kernel/linux-2.4.21-51Board_EDR
Linux Kernel v2.4.21-rmk1-pxa1-xsbase270 Configuration
XSBASE Driver
Arrow keys navigate the menu. <Enter> selects submenus --->.
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
<M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
Legend: [*] built-in [ ] excluded <M> module < > module capable
[*] XSBASE support
<M> TEST user-space interface (NEW)
[*] TEST CPU (NEW)
<Select> < Exit > < Help >
ModuleTest.tar [唯讀]
[root@esl24 source]#
[root@51Board ~]# mkdir make_menuconfig
[root@51Board ~]#

編輯修改CPU體系目錄下的設定檔

- 在arch/arm/config.in檔進行修改，在檔案最後加入：source drivers/xsbase/Config.in

Makefile的修改與編輯

- 編輯xsbase目錄下的設定檔Makefile檔

```
#       drivers/xsbase/Makefile
#
#       Makefile for the XSBASE TEST.
#
L_TARGET := test.o
export-objs := test.o
obj-$(CONFIG_TEST)              += test.o
include $(TOPDIR)/Rules.make
clean:
    rm -f *.[oa] .*.flags
```

Makefile的修改與編輯

- 編輯 drivers/Makefile

```
subdir-$(CONFIG_MMC)            += mmc
subdir-$(CONFIG_XSBASE)          += xsbase
……
include $(TOPDIR)/Rules.make
```

Makefile的修改與編輯

- 編輯 Makefile

……

```
DRIVERS-$(CONFIG_PLD) += drivers/pld/pld.o
DRIVERS-$(CONFIG_MMC) += drivers/mmc/mmcdrivers.o
DRIVERS-$(CONFIG_XSBASE) += drivers/xsbase/test.o
DRIVERS := $(DRIVERS-y)
```

……

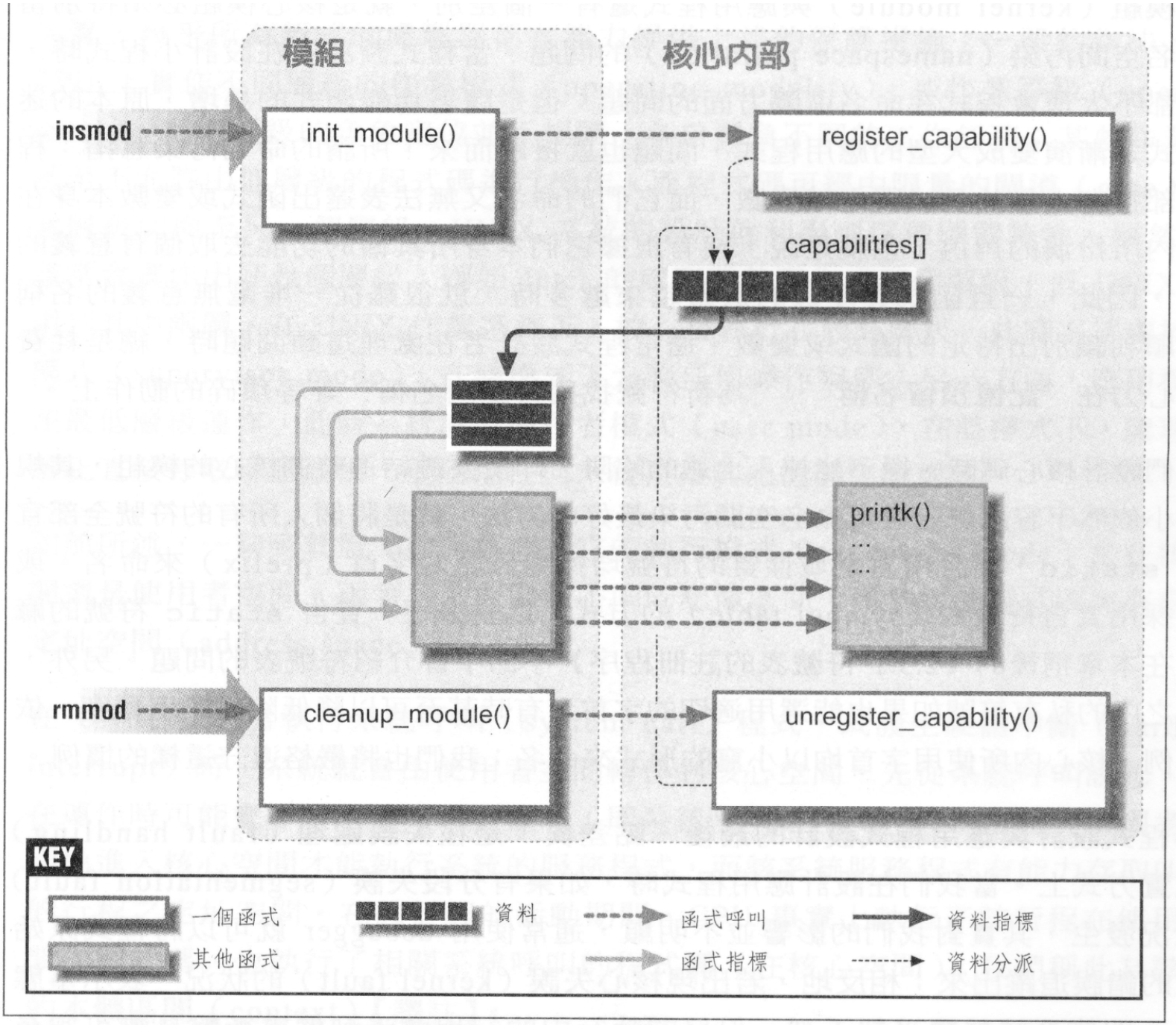
Driver Testing

- Use “arm-linux-gcc –c –o test_client test_client.c” to compile test program.
- Copy test_client and test.o to PXA270 by nfs.
- Create device node

  mknode “/dev/xsb_edr_8led” c 60 1
- Insert module

  insmod test.o
- Run application

  ./test_client


模組
核心內部
insmod
init_module()
register_capability()
capabilities[]
printk()
...
...
rmmod
cleanup_module()
unregister_capability()
KEY
一個函式
其他函式
資料
函式呼叫
函式指標
資料指標
資料分派