

Kernel Programming

Table of contents

1 Linux Native Boot Process.....	2
2 Keyboard Driver.....	2
3 System calls.....	2
4 Processes (part I).....	2
5 Serial Driver.....	2
6 TTY layer.....	2
7 Floppy Driver.....	2

1. Linux Native Boot Process

Questo articolo introduce il boot process di Linux e cerca di illustrare al lettore, step by step, cosa realmente accade durante questa fase. L'articolo inizia mostrando come si crea un floppy di boot capace di stampare la stringa "Hello World" e, partendo da questo semplice esempio, ad ogni step aggiunge sempre nuove funzionalità, fino a completare il processo.

Clicca [qui](#) (boot.html) per leggere l'articolo.

2. Keyboard Driver

Questo articolo vuole essere una guida step by step che aiuterà il lettore nella scrittura di un driver per tastiera e console basato sul codice sorgente di Linux.

Clicca [qui](#) (keyboard.html) per leggere l'articolo.

3. System calls

Questo articolo spiega come aggiungere al nostro kernel l'infrastruttura per il supporto delle system calls.

Clicca [qui](#) (syscalls.html) per leggere l'articolo.

4. Processes (part I)

Al più presto verrà pubblicato un articolo che spiega, step by step, come aggiungere al nostro kernel il supporto dei processi. In questa prima parte verrà spiegato come aggiungere: uno scheduler, un memory management semplificato e le system calls fork ed exec. Per ora sono disponibili solo i sorgenti.

Clicca [qui](#) (processes.html) per leggere l'articolo.

5. Serial Driver

Al più presto verrà pubblicato un articolo che spiega, passo dopo passo, come aggiungere al nostro kernel un driver seriale. In esso, verranno introdotti i concetti hardware e software per la programmazione di un chip UART. Verrà mostrato, inoltre, come ottenere una connessione seriale a una vera linux box ed eseguire in essa numerosi comandi bash.

Clicca [qui](#) (serial.html) per leggere l'articolo.

6. TTY layer

Al più presto verrà pubblicato un articolo che spiega, passo dopo passo, come aggiungere al nostro kernel il layer TTY. In esso illustreremo come aggiungere 6 processi capaci di gestire una console locale o seriale. I primi 4 processi gestiranno una console locale in cui tutto ciò che viene digitato da tastiera viene stampato a video, invece, i restanti due processi gestiranno due console seriali collegate alle porte COM1 e COM2 grazie alle quali saremo in grado di gestire una vera linux box remota collegata al computer locale tramite cavo seriale null modem.

Clicca [qui](#) (tty.html) per leggere l'articolo.

7. Floppy Driver

In questo articolo viene presentato un floppy driver read-only per il nostro kernel didattico.

Clicca [qui](#) (floppy.html) per leggere l'articolo.

Open Community (C) 2004