



嵌入式Linux内核裁减与移植

Copyright 2007-2008 Farsight.
All rights reserved.



Linux简介

- 姓名： Linux
- 生日： 1991 年10月
- 父亲： Linus Torvalds
- 堂兄： UNIX
- 国别： 芬兰
- 家庭住址： <http://www.linux.org/>
- 吉祥物：



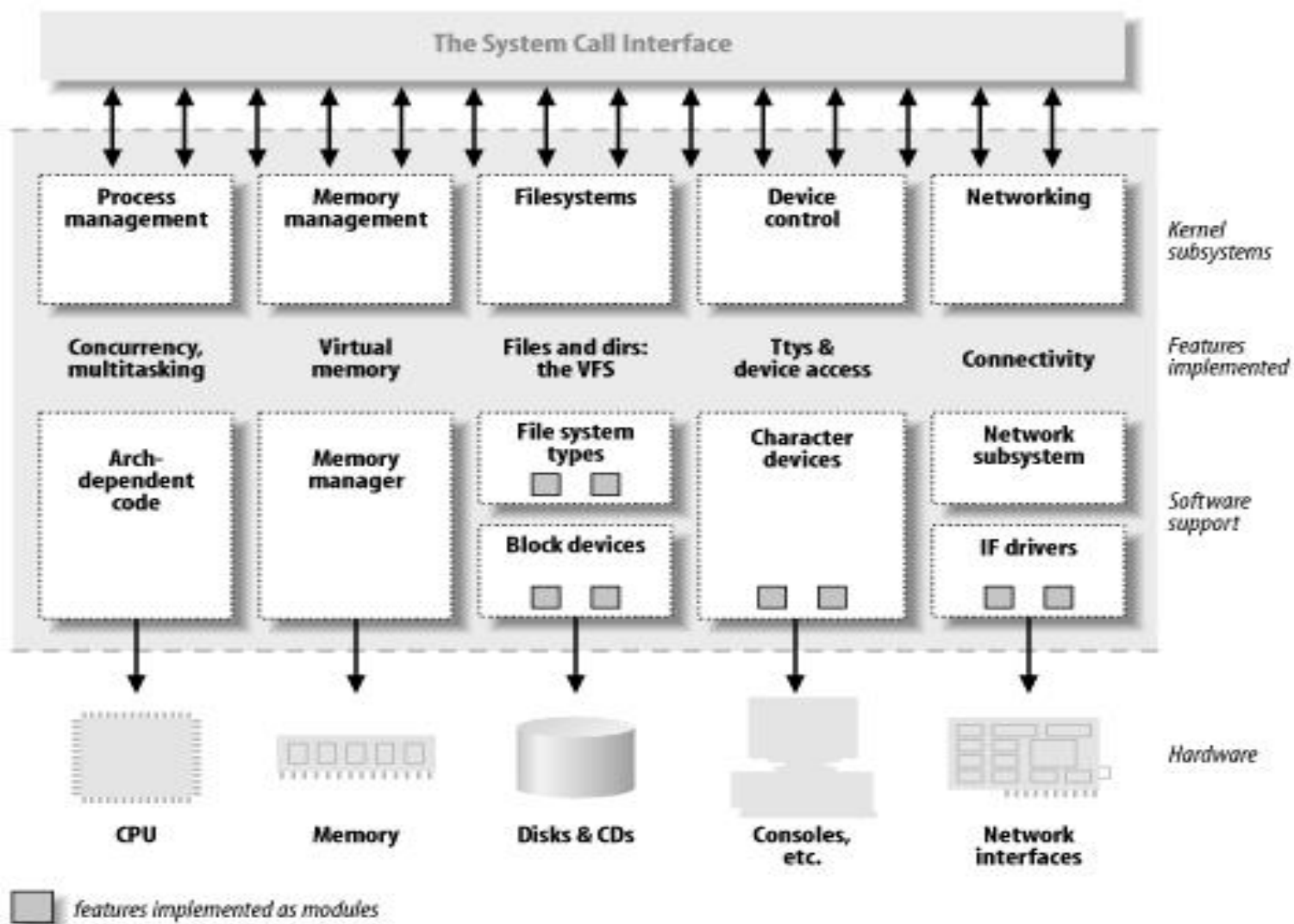


Linux优点

- 提供了先进的网络支持
- 多任务、多用户
- 符合IEEE POSIX标准
- 支持数十种文件系统格式
- 完全运行于保护模式
- 开放源代码
- 采用先进的内存管理机制，更加有效地利用物理内存



Linux Kernel 组件





进程

- 进程的定义
- 进程调度器
- 进程 VS 线程



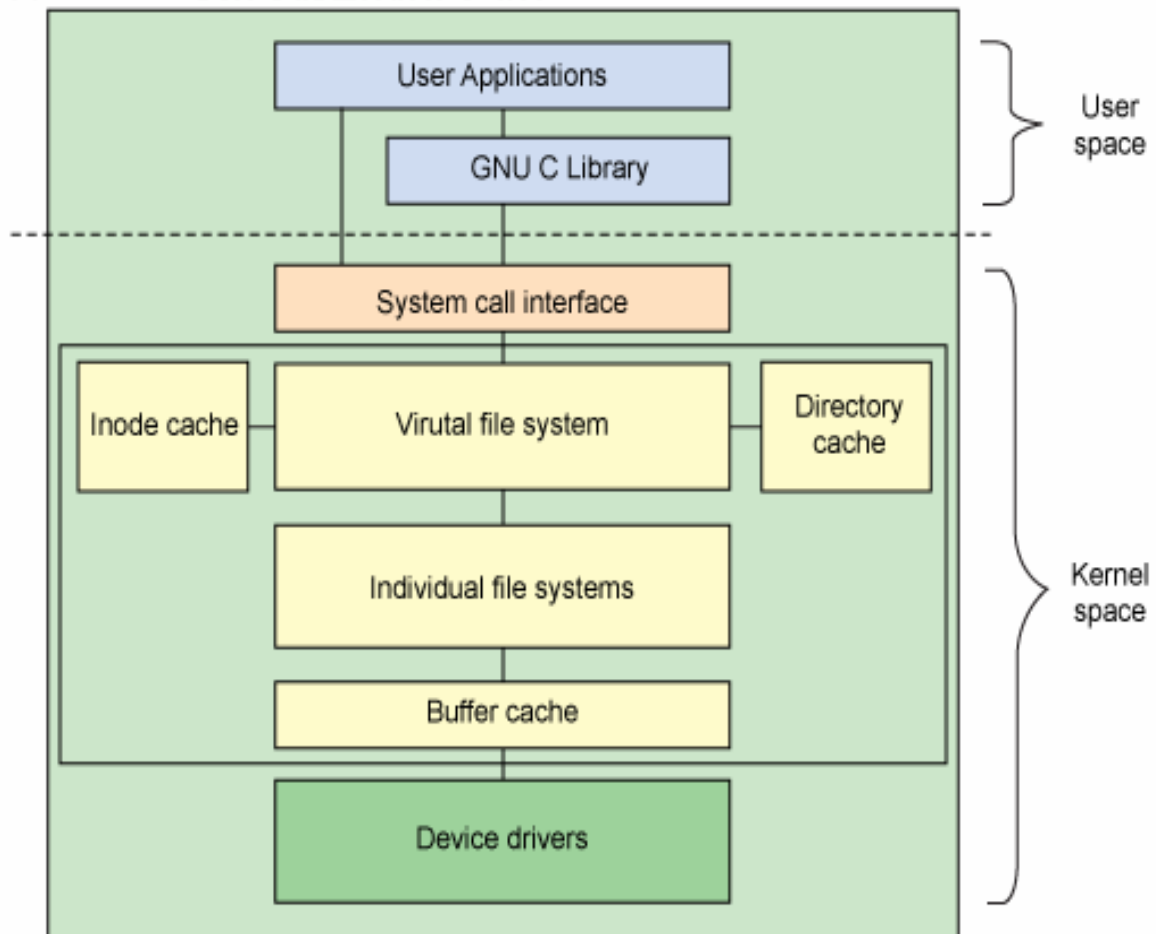
内存系统

- 地址转换
- 内存管理
 - 基于页式管理
- 内存分配
 - 基于slab算法



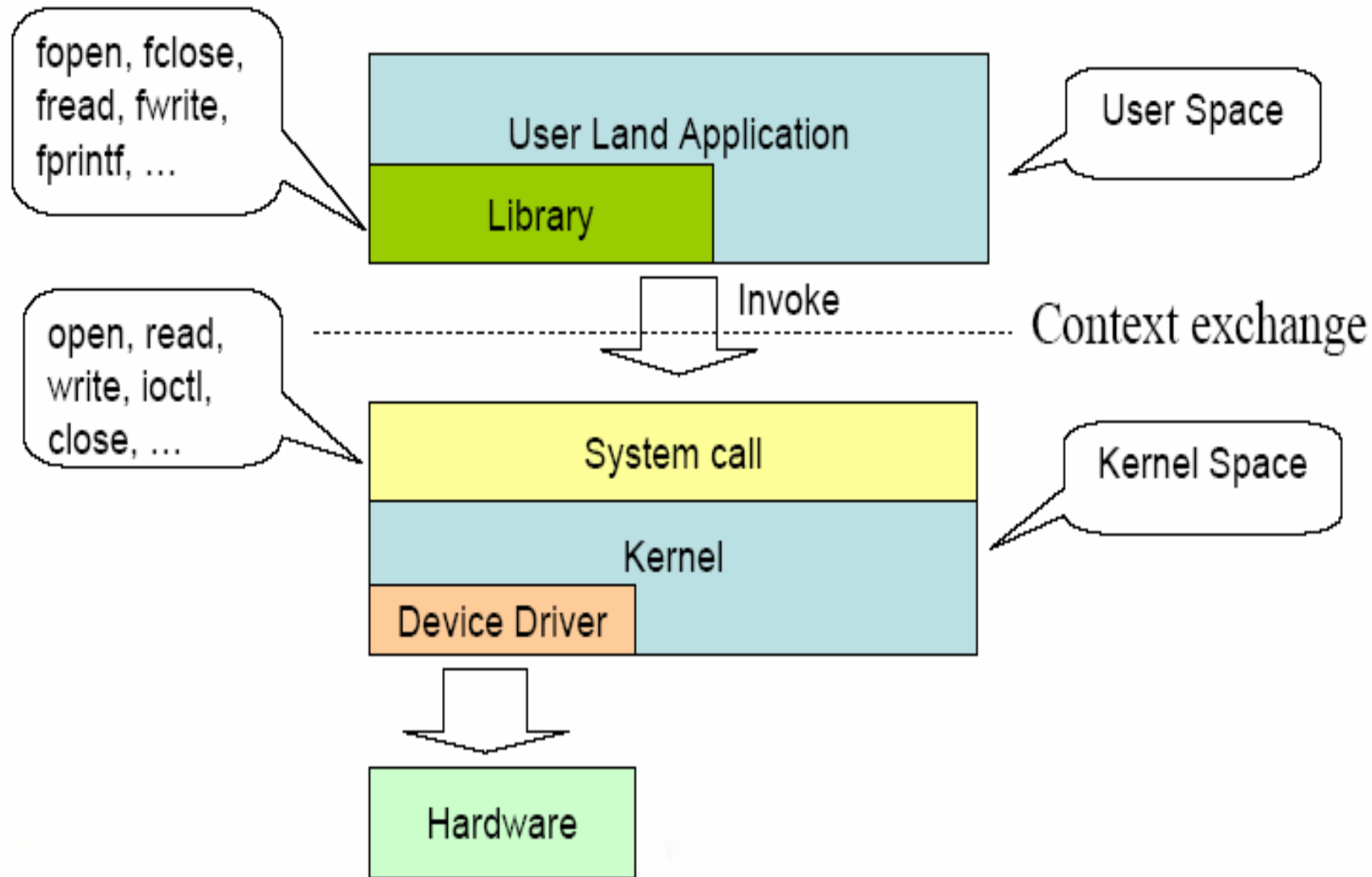
文件系统

图 1. Linux 文件系统组件的体系结构





Bridge to AP and Kernel





Kernel configuration

make config

- Asks you the questions 1 by 1. Extremely long!

make menuconfig

- Same old text interface as in Linux 2.4. Useful when no graphics are available.

make gconfig

- New GTK based graphical configuration interface for Linux 2.6. Functionality similar to that of make xconfig.



Make menuconfig

```
Linux Kernel v2.6.16.26 Configuration

Networking options

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, </>
for Search. Legend: [*] built-in [ ] excluded <M> module < >

[ ] Network packet debugging
<M> Packet socket
[*] Packet socket: mmaped IO
<*> Unix domain sockets
< > PF_KEY sockets
[*] TCP/IP networking
[ ] IP: multicasting
[ ] IP: advanced router
[*] IP: kernel level autoconfiguration
[ ] IP: DHCP support
[ ] IP: BOOTP support
■(+)
```

<Select> < Exit > < Help >



Compiling statically or as a module

- Compiled as a separate module
 - CONFIG_ISO9660_FS=m
- Driver options
 - CONFIG_USB=y
 - CONFIG_SCSI=y
- Compiled statically in the kernel
 - CONFIG_PROC_FS=y



Cross-compiling setup

Example

- If you have an ARM crosscompiling toolchain in `/usr/local/arm/3.3.2/bin`
- You just have to add it to your Unix search path:
`export PATH= /usr/local/arm/3.3.2/bin :$PATH`
- Double check your CrossCompile again
`arm-linux-gcc -v`
Thread model: posix
gcc version 3.3.2



Cross-compiling the kernel

- Update the version as usual
- You should change the default target platform.
 - Example: ARM platform, crosscompiler command: arm-linux-gcc
 - ARCH = arm
 - CROSS_COMPILE = arm-linux-
 - The Makefile defines later CC = \$(CROSS_COMPILE)gcc
- or run (arm example):
 - make ARCH=arm CROSS_COMPILE=arm-linux-
 - Useful when you compile for several platforms

See comments in Makefile for details



Building the kernel

- Run make (if you have modified your Makefile) or otherwise (ARM example) make ARCH=arm CROSS_COMPILE=arm-linux-
- Copy arch/<platform>/boot/zImage to the target storage
- You can customize arch/<arch>/boot/install.sh so that make install does this automatically for you.
- make INSTALL_MOD_PATH=<dir>/ modules_install and copy <dir>/ to /lib/modules/ on the target storage