How to Get Started in Kernel Programming

Jeffrey Hsu

hsu@dragonflybsd.org

Prerequisites

- 1 letter
- 2 machines

Configure and Build Kernel

- edit config file
- make

Sources of information

- BSD man page
- standards
  - POSIX
  - RFC
- Online resources
  - handbook

    http://www.dragonflybsd.org/docs/handbook/
  - http://www.freebsd.org/doc/en/books/developers-handbook/

Read the source

- Header files
- Source tree overview

queue.h

- Man page
- Usage
- mutable

Do a project

- Write a system call
- Write a sysctl
- Write a pseudo-device driver, /dev/niptuck
- Write a loadable kernel module

Ways into the Kernel

- system call
- sysctl
- ioctl
- /proc

debugging

- crash inevitable
- ddb
- kgdb
- KTR
- http://www.dragonflybsd.org/docs/user/DebugKernelCrashDumps/

Anatomy of a Syscall

- libc/i386/SYS.h
- kern/syscalls.master
- trap.c
- copyin/copyout
- descriptors

Locking

- spinlock
- mutex locks
- ssleep example

VM

- red-black trees
- Dillon’s VM writeup
  http://www.freebsd.org/doc/en/articles/vm-design

Networking

- peruse source in net/ and netinet/
- Stevens, TCP/IP Illustrated Vol 2

VFS

- struct vnode
- kern vfs
- individual filesystems
  - nullfs

Memory allocation

- object cache
- Source
  - sys/malloc.h
  - kern_malloc.c
- blocking and non-blocking memory allocations

Scheduler

- null scheduler
- random scheduler project

Bibliography

- Vahalia, Unix Internals: The New Frontiers
- McKusick, The Design and Implementation of the FreeBSD Operating System
- Curt Schimmel, Unix Systems for Modern Architectures, Addison-Wesley