FreeBSD Internals
Day 3

Joel Isaacson
Ascender Technologies Ltd

Device Drivers

- A device in this context is a term used mostly for hardware-related stuff that belongs to the system, like disks, printers, or a graphics display with its keyboard.
- A device driver is the software component of the operating system that controls a specific device.

Device Drivers

- There are also so-called pseudo-devices where a device driver emulates the behavior of a device in software without any particular underlying hardware.
- Device drivers can be compiled into the system statically or loaded on demand through the dynamic kernel linker facility `kld'.

Dynamic Kernel Linker Facility

- The kld interface allows system administrators to dynamically add and remove functionality from a running system.
- This allows device driver writers to load their new changes into a running kernel without constantly rebooting to test changes.

Dynamic Kernel Linker Facility

- The kld interface is used through the following privileged commands:
  - kldload - loads a new kernel module
  - kldunload - unloads a kernel module
  - kldstat - lists the currently loaded modules

Dynamic device nodes

- The device filesystem, or devfs, provides access to the kernel's device namespace in the global filesystem namespace.
- This eliminates the problems of potentially having a device driver without a static device node, or a device node without an installed device driver.

Character Devices

- A character device driver is one that transfers data directly to and from a user process.
- This is the most common type of device driver and there are plenty of simple examples in the source tree.

Block Devices (Are Gone)

- Block devices are disk devices for which the kernel provides caching.
- This caching makes block-devices almost unusable, or at least dangerously unreliable.
- FreeBsd has done away with block devices and kept only the character devices.

Network Drivers

- Drivers for network devices do not use device nodes in order to be accessed.
- Their selection is based on other decisions made inside the kernel and instead of calling open(), use of a network device is generally introduced by using the system call socket(2).

Basic Character Drivers

- There are two sets of operations that are done by drivers:
  - Drivers initialization and termination
  - Driver data operations

Basic Character Drivers Initialization and Termination

- Sometimes called “bus methods”. The calls are
  - probe – find if the device is present
  - attach – attach the device, allocates resources
  - detach – detach the device
  - shutdown

device_t pointer

- Device_t is the pointer type for the device structure.
  - device_get_parent(dev) Get the parent bus of a device.
  - device_get_driver(dev) Get pointer to its driver structure
  - device_get_name(dev) Get the driver name.
  - device_get_name(dev) Get the driver name

device_t pointer

- Device_t is the pointer type for the device structure.
  - device_get_desc(dev) Get the device description.
  - device_set_desc(dev, desc) Set the description.
  - device_get_softc(dev) Get pointer to the device descriptor

Resources

- The information that a user enters into the kernel configuration file is processed and passed to the kernel as configuration resources.
- This information is parsed by the bus configuration code and transformed into a value of structure device_t and the bus resources associated with it.

Resources

- The bus resources are associated with each device. They are identified by type and number within the type.
  - SYS_RES_IRQ - interrupt number
  - SYS_RES_DRQ - ISA DMA channel number
  - SYS_RES_MEMORY - device memory mapped into the system memory space
  - SYS_RES_IOPORT - range of device I/O registers

Resources

- Three types of activities can be performed on resources:
  - set/get
  - allocate/release
  - activate/deactivate