Writing Device Drivers Under Linux

Character Device Drivers

Character Devices

- Simplest kind of device
- Character oriented
- Often not random access

Example of character devices

- Mouse device
- Data acquisition devices
- /dev/random
- /dev/null
- /dev/zero

Other character devices

- TTY devices
- SCSI tape devices
- There are special kernel interfaces for these, which we will consider later

Kernel interface for character devices

- Uses Linux's filesystem abstractions directly
- Hence, simplest to program

Elements of a character device driver

- Initialization function
- Open function
- Close function
- Read function
- Write function
- Lseek function (if random access allowed)

mem.c example

- Page 4 in the example sources
- Implementation of standard, old and familiar devices
  - /dev/null
  - /dev/full
  - /dev/zero
  - /dev/mem

Initialization function

- Look at chr_dev_init() (mem.c:409)
- Master initialization function
  - Called at boot time
  - Each char device driver compiled into the kernel is initialized from here.
- register_chrdev() function registers a file_operations table for a major device number

Device open function

- Look at memory_open() (mem.c:358)
- Called to open a device with major #1
- Normally the device open function initializes device hardware, etc.
- In this case, this function:
  - Uses the minor device number as a dispatch
  - Replaces the file operations table
  - Calls the minor device open function (if present)

Device read function

- Look at read_mem() (mem.c:48)
- Takes four arguments: inode, file, memory pointer, size
- Reads up to <size> bytes and places them in memory_pointer
- This function may block

Device write function

- Look at write_mem() (mem.c:79)
- Takes four arguments: inode, file, memory pointer, size
- Writes up to <size> bytes and places them in memory_pointer
- This function may block

Device lseek function

- Look at memory_lseek() (mem.c:247)
- Takes four arguments: inode, file, offset, lseek_flag
- Sets file->f_pos
- Filesystem layer provides a default lseek() function

Memory mapping

- Beyond the scope of this course
- Needed for mmap() on your device to work
- Automatically handled by the block and SCSI device abstraction

Bus Mouse driver

- A more complex character device driver
- Uses
  - The miscellaneous major layer
  - Interrupt driver

Miscellaneous major driver

- Look at page 12 (misc.c)
- Major numbers are a limited resource
- Many devices don't need their own major number
- The misc_major driver dispatches to individual drivers based on minor device number

January 7, 1997 Character Device Drivers 15

Interrupt routine

- Registered in open_mouse() (busmouse.c:141)
- Interrupt routine is mouse_interrupt() (busmouse.c:63)
- Routine reads data from the mouse and wakes up the wait queue `mouse.wait`

January 7, 1997 Character Device Drivers 16

Select routine

- Look at mouse_select() (busmouse.c:222)
- Takes four arguments: inode, file, select type, select table
- Called by the select system call

Select routine II

- If the device is ready, return 1
- If the device is not ready
  - Call select_wait() to register a wait_queue with the select table
  - Return 0