ELEC 377 – Operating Systems

Week 9 – Class 3

Last Week

- I/O Systems
  - Block and Character Devices

Today

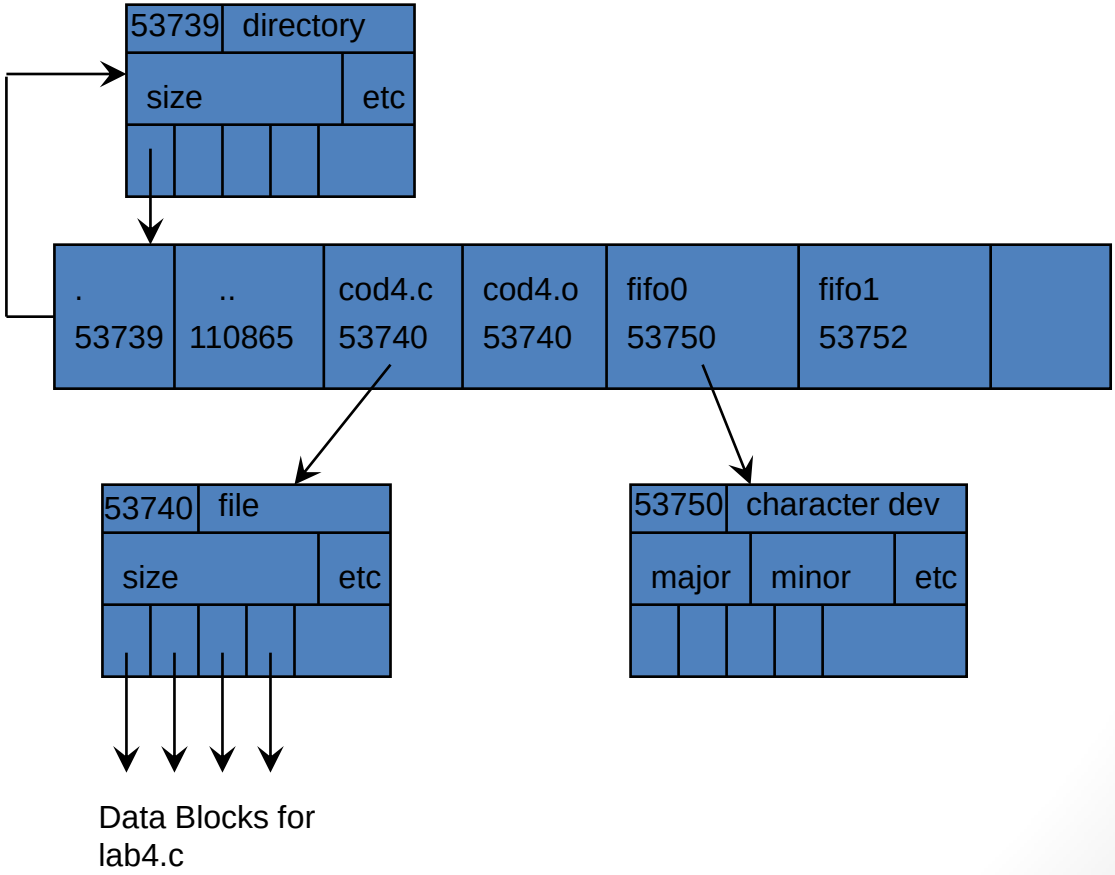
- I/O Systems
  - ◊ Block and Character Devices
  - ◊ Network Devices
  - ◊ Kernel Services
- Distributed Systems

/dev filesystem

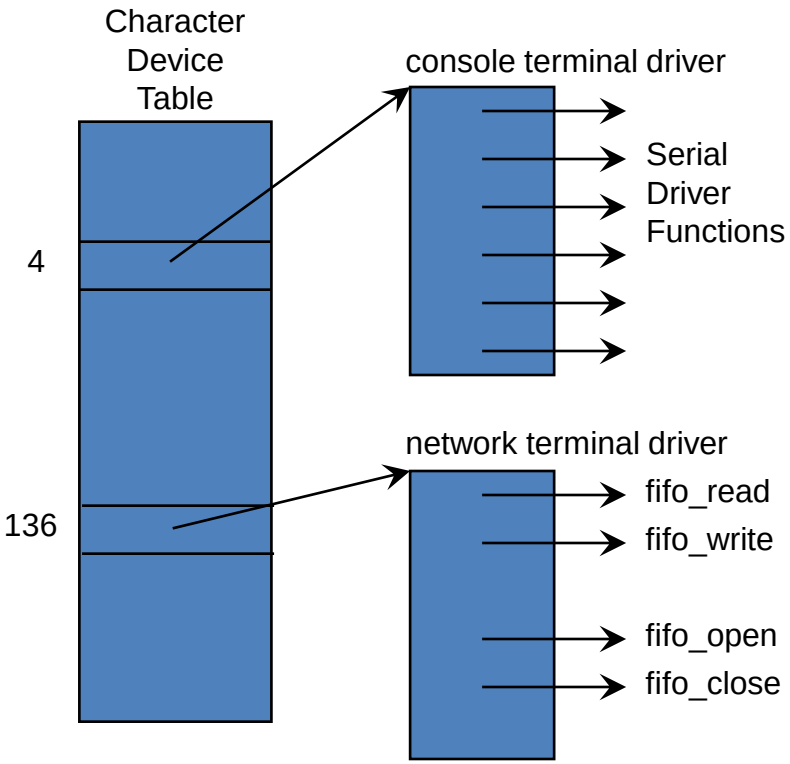
- Direct access to devices under unix.
  - ◊ device access looks like a file
- Everything looks like a file under unix
  - ◊ inode attributes specify device
  - ◊ alternate layout of inode
  - ◊ no data blocks
  - ◊ major and minor numbers
    - major number specifies device driver
    - minor number specifies device
- Traditionally, these files are in /dev, but can be anywhere on the disk. (device id in inode)
  - ◊ traditional names (hda - ide, sda- scsi, tty - console and serial, pty - network terminal) , but names can be anything, major number specifies driver

Directories and Files Structures

- inodes are the file control blocks in Unix
  - ◊ directories are files



Character Device Drivers



- The major number connects the device file to the device driver.

There can be multiple files with same major, same or different minors, all connect to the device driver

/dev filesystem examples

- Console tty

crw--w---- 1 shannon tty 4, 1 Nov 10 09:39 /dev/tty1
crw------- 1 root root 4, 2 Sep 9 10:57 /dev/tty2

  - ◊ character device
  - ◊ major = 4, minor = 1
  - ◊ shannon is logged in to the first console
  - ◊ minor specified which console (alt-F*n*)

- network pseduo tty

crw--w---- 1 dean tty 136, 2 Nov 10 09:40 /dev/pts/2

  - ◊ used to simulate terminal for network connection
  - ◊ console editors such as vi need terminals
  - ◊ major = 136, minor = 2

/dev filesystem examples

- IDE drives

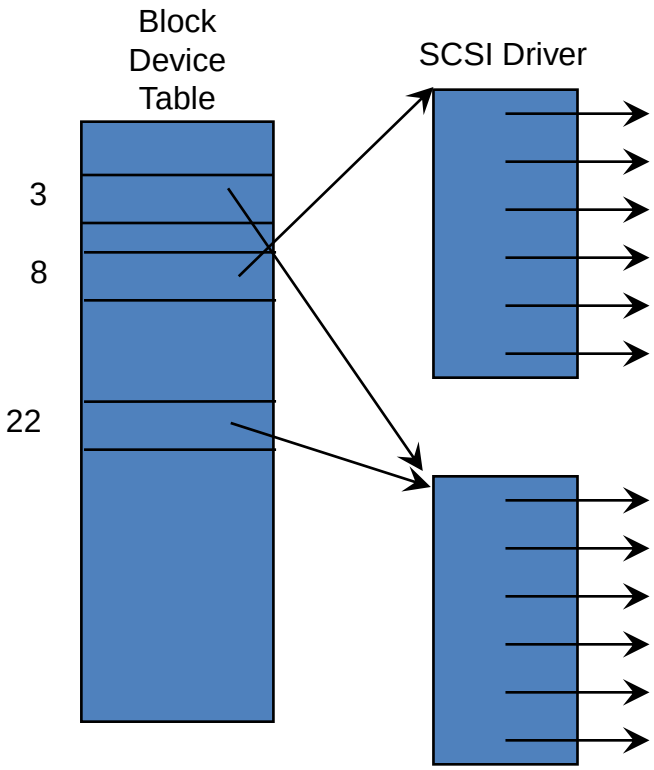
```
brw-rw----   1 root     disk      3,   0 Jan 30  2003 /dev/hda
brw-rw----   1 root     disk      3,   1 Jan 30  2003 /dev/hda1
brw-rw----   1 root     disk      3,  64 Jan 30  2003 /dev/hdb
brw-------   1 shannon  disk     22,  64 Jan 30  2003 /dev/hdd
```

  - ◊ block device
  - ◊ major = 3, minor = 1
  - ◊ different major for second ide bus, but same driver
  - ◊ more than one major may point to the same driver

- SCSI:

```
brwxrwxrwx   1 root     disk      8,   1 Jan 30  2003 /dev/sda1
```

Device Drivers with multiple Majors



- One of the parameters to the device driver is the major number. Thus the same device driver can be used for multiple majors

In the case of the IDE driver, Major 3 = IDE bus 1 and Major 22 = IDE BUS 2.

Today

- I/O Systems
  - ◊ Block and Character Devices
  - ◊ Network Devices
  - ◊ Kernel Services
- Distributed Systems <<<<<<

Distributed Systems

- What is a distributed system?
  - ◊ loosely coupled processors connected by a communication network
  - ◊ Local Resources vs. Remote Resources

- Reasons for Distributed Systems
  - ◊ Resource Sharing
  - ◊ Computation Speedup
  - ◊ Reliability
  - ◊ Communication

Types of Distributed Systems

- Network Operating System
  - ◊ Unix, Windows, Macintosh
  - ◊ Users are aware of network
  - ◊ Users are aware of local vs remote resources
  - ◊ remote login, file transfer

- Distributed Operating System
  - ◊ Users are not aware of network
  - ◊ Data Migration
  - ◊ Computation Migration (RPC)
  - ◊ Process Migration

Distributed Operating System

- Data Migration
  - ◊ Network File Systems
  - ◊ Andrew File System (AFS)
  - ◊ NFS, SMB, AFSP
- Computation Migration
  - ◊ Remote Procedure Call (RPC, RMI)
  - ◊ Stored Procedures in Databases
  - ◊ CORBA
  - ◊ Data vs Commands

Distributed Operating System

- Process Migration
  - ◊ Load Balancing(LoadLeveler, Condor)
  - ◊ Computation Speedup (RenderFarms, RSA Factoring, Seti@Home)
  - ◊ Clusters (Beowulf)
  - ◊ Hardware Preference (heterogenous systems)
  - ◊ Software Preference (heterogenous systems)

Network Topology

- Networks connected in a variety of ways
  - ◊ Installation Cost
  - ◊ Communication Cost
  - ◊ Availability (Reliability)
- Fully Connected
  - ◊ Expensive, but every node direct to other nodes
- Partially Connected
  - ◊ Variety of possible connections
  - ◊ Combinations possible

Network Types

- Local Area Networks (LAN)
  - ◊ High Speed, High Cost
  - ◊ Ethernet (1 Mbit - 1 Gigbit)
  - ◊ Token Ring, Optical Fibre
  - ◊ Short Distance (100's Meters)
- Wide Area Network (WAN)
  - ◊ Long Distance (100's -1000's Km)
  - ◊ Internet (Arpanet)
  - ◊ Private Networks (IBM Global Services, UUNet)
  - ◊ Routers
  - ◊ Slow (T1 = 1.544 Mbits, T4= 28 T1 = 45 Mbits, ISDN = 128Kbits, 56K, 33k, DSL)

Distributed File Systems

- Integral Part of Distributed Operating Systems
  - ◊ Many implementations
  - ◊ Data Migration
- Concepts
  - ◊ **service** – software entity running on one or more machines providing a particular function (file access)
  - ◊ **server –** a machine running the service software
  - ◊ **client** - process that can invoke a service
  - ◊ **client interface** - operations on the service available to clients
  - ◊ Machine may be both a server and a client - Peer-to-peer

Distributed File Systems

- Ideally, a distributed file system looks the same where ever you log in
  - ◊ Suns in CASLAB - home file system is on zeus
  - ◊ Single server - relatively easy
  - ◊ Research System
    - multiple unix systems each with disk space
    - /home/stephan is located on cetus.ee.queensu.ca
    - /home/stephan is a remote mount on all other machines
    - /home/li is located on orion.
    - /home/li is a remote mount on all other machines
  - ◊ Machines are all both server and client

Distributed File Systems

- Transparency - remote and local disks look the same
  - ◊ Virtual File System abstracts interface to multiple file systems.
  - ◊ User does not know where the files are located
  - ◊ There may be more than one copy of a file - (**replication**)
  - ◊ Two components
    - Location Transparency (static)
      - name does not reveal location
    - Location Independence (stronger, dynamic)
      - name does not change if location changes
      - file migration

Distributed File Systems

- Location transparency vs independence
  - ◊ separate data from location
  - ◊ static location transparency - share files
    - location independence - share space
  - ◊ separate naming hierarchy from storage hierarchy
    - remove restrictions on system architecture

Distributed File Systems

- Diskless workstations
  - ◊ ROM loads kernel from server
  - ◊ popular in Late 80’s
  - ◊ resurgence now

Distributed File Systems

- File Naming - three approaches
  - ◊ host + location
    - not location transparent not location independent
  - ◊ Attach remote directories to local directories
    - remote mount
    - permissions??
  - ◊ total integration
    - global name structure spans all files
    - problem with special files
  - Examples:
    - ◊ most current DFS tie location to mount point. Drive Z: is on files.engineering.queensu.ca
    - ◊ difficult to move a single file on Z: to be on some other server. All the files on the other server are on H:

Distributed File Systems - Caching

- Performance
  - ◊ network overhead in addition to other I/O overhead
  - ◊ Similar to cache for disk I/O
- Consistency
  - ◊ more than one client may be accessing same file
    - client initiated (check for consistency before using cached value)
    - server initiated (track clients and notify)
- Location
  - ◊ main memory
  - ◊ local disk

State

- Stateful Connections
  - ◊ Connection between client and server is persistent
  - ◊ Server keeps track of all clients
  - ◊ Amortize overhead of connection, I/O
  - ◊ AFS, AFPS
- Stateless
  - ◊ Each operation is a separate request
  - ◊ NFS
- Tradeoffs
  - ◊ Client/Server Crash?
  - ◊ Performance

Security - Three meanings

1. Protection and Authentication
   - ◊ Identify Users
   - ◊ Users only access information they have privileges for
   - ◊ secrecy
2. System Integrity
   - ◊ Only authorized users
   - ◊ prevent execution of code by outsiders
3. Information Security
   - ◊ Statistical Attacks
   - ◊ Medical/Financial

Security

- Security
  - ◊ impossible in practice
  - ◊ accidental violations (easy to protect)
  - ◊ malicious (harder)
    - – Reading of data (info theft)
    - – Modification of data
    - – Destruction of data
    - – Denial of service
  - ◊ Cost tradeoffs

Security Levels

- Physical
  - ◊ bios on PC
- Human
  - ◊ social engineering
- Network
  - ◊ packet interception, denial of service
- OS
  - ◊ only level OS has control over

  - first two are outside of OS control but necessary
  - hardware protection for OS
  - harder to add security than design for it

Security - Informational Security

- Statistical Attacks
  - ◊ Individual pieces of information reveal nothing
  - ◊ collectively they reveal private information
  - ◊ statistics databases
- Statistics Canada
  - ◊ only release information in predefined categories
- Traffic Analysis
- User Generated Queries
  - ◊ carefully crafted queries
  - ◊ refuse queries whose results are small counts
  - ◊ You and Joe are the only mid level managers - “what is the average of mid levels managers salaries” tells you Joes salary.