Embedded Operating System

Agenda

- System Call
- System Call Execution
- Dual Mode Protection
- Security and Protection
- Networking
- File Management
- User Interfacing

System Calls

- Software interrupt is used to implement OS/Kernel services.

- Functions exposed by the kernel so that user programs can access kernel functionalities, are called as "System calls".
  - e.g. Process Mgmt: create process, exit process, communication, synchronization, etc.
  - e.g. File Mgmt: create file, write file, read file, close file, etc.
  - e.g. Memory Mgmt: alloc memory, release memory, etc.
  - e.g. CPU Scheduling: Change process priroty, change process CPU affinity, etc.

- System calls are specific to the OS:
  - UNIX: 64 syscalls e.g. fork(), ..
  - Linux: 300+ syscalls e.g. fork(), clone(), ...
  - Windows: 3000+ syscalls e.g. CreateProcess(), ...

- Read -- Dual Mode Protection & IO Protection (Galvin)

System Call Execution

- Linux System call on ARM Cortex-A
  - Linux maintains addresses of all system call implementations in a syscall table.
  - The swi_handler get address of syscall implementation and invoke it.
  - System call API generate software interrupt using swi instruction.
  - Arguments are passed in r0, r1, r2, r3 and value is returned in r0.
  - The syscall number is passed in r7.

Security and Protection

- Security is securing system from "external" threats e.g. virus, trojans, worms, hacking, etc.
  - Security is optional feature and is not implemented in many OS.
  - Security is usually provided by "Anti-virus" application.
  - Windows 10+ comes with Windows Defender, which is handling security aspects.

- Protection is protecting system (programs & files) from internal threats/elements.
  - Dual mode protection: CPU can differentiate whether code belongs to OS or user application.
  - IO protection: Only OS should be able to perform IO operations.
  - User programs should use system calls to perform IO.
  - This is feasible when IO instructions are privileged instructions (they can only be executed in kernel mode).
  - Memory protection: One process should not access memory of another process directly, so that one process cannot disturb execution of another process.
  - This is implemented using MMU.
  - CPU protection: If a process goes in infinite loop, the whole system should not hang. This is done using Timer hardware.

Networking

- Networking feature enable computers (processes) to communicate with each other.
- Even though important (nowadays), networking is optional feature of OS.
- For networking, computers are connected to each other in LAN, MAN or WAN.
- Computers are connected with different topologies e.g. bus, star, ring, mesh, ...
- Networking feature internally use "sockets" IPC mechanism.
- Socket is communication end-point.

File Management

- File is collection of data/information on storage device.
  - File = Contents (Data) + Information (Metadata)
  - The data is stored in zero or more Data blocks (in FS), while metadata is stored in the FCB (in filesystem).

- FCB is called as "inode" in UNIX/Linux It contains
  - type: UNIX/Linux has 7 types of files
    - -: regular, d: directory, l: symbolic link, p: pipe, s: socket, c: char device, b: block device
  - size: number of bytes
  - links: number of hard links
  - mode (permissions): (u) rwx, (g) rwx, (o) rwx user & group
  - time-stamps: modification, creation, access.
  - info about data blocks

  - terminal> ls -l type,
    - mode, links, user, group, size, timestamp, name.
  - terminal> stat filepath

File System

- Files are stored on storage device. Arrangement of files in storage device is called as **"File System".**

- e.g. FAT, NTFS, EXT2/3/4, ReiserFS, XFS, HFS, etc.

- File System logically divide partition into 4 sections.
  - **Boot block/Boot sector**
    - Contains programs/info required for booting of OS.
    - Typically contains bootstrap program and boot loader program .
  - **Super block/Volume control block**
    - Contains information of whole partition.
    - Capacity, Label.
    - terminal> df -h Total number of data blocks / inodes.
    - Number of used/free data blocks/ inodes.
    - Information of free data blocks/ inodes.
  - **Inode List/Master file table**
    - Inodes (FCB) for each file
  - **Data blocks**
    - Stores data of the file.
    - Each file have zero or more data blocks.
    - Size of data blocks can be configured while creating file system.

- File system is created by the format utility while formatting the partition.
  - Windows: format.exe
  - Linux: mkfs
    - terminal> sudo mkfs -t ext3 /dev/sdb1
    - terminal> sudo mkfs -t vfat /dev/sdb1
      - -t fs_type e.g. ext3, ext4, vfat, ntfs, ...
    - partition e.g. /dev/sdb1

User interfacing

- **UI of OS is a program (Shell) that interface between End user and Kernel.**
  - Shell -- Commmand interpreter
    - End user --> Command --> Shell --> Kernel
  - User interfacing (Shell)
    - Graphical User Interface (GUI)
    - Command Line Interface (CLI)

- **Example shells**
  - Windows
    - GUI shell: explorer.exe
    - CLI shell: cmd.exe, powershell.exe
  - DOS
    - CLI shell: command.com
  - Unix/Linux
    - CLI shell: bsh, "bash", ksh, csh, zsh, ...
    - ls /bin/*sh echo $SHELL
    - shell of current user can be changed using "chsh" command.

- **GUI shell/standards**
  - GNOME: GNU Network Object Model Environment (e.g. Ubuntu, Redhat, CentOS, ...)
  - KDE: Kommon Desktop Environment (e.g. Kubuntu, SuSE, ...)
  - XFCE: XForms Common Environment (e.g. Raspberry Pi, ...)

- **Shell Scripts**
  - Shell scripts is collection of program along with programing constructs .
  - Shell scripts syntax is different from shell to shell .

Thank you!

Kiran Jaybhave

email – kiran.jaybhave@sunbeaminfo.com