OS Concepts & Architecture

BHUPENDRA PRATAP SINGH

CDAC, ACTS, PUNE

What is Operating System (OS)?

- Interface/bridge/glue between applications & Hardware
- Basis for executing applications
- Resource manager (physical, logical resources)

Is it possible to design apps without the need of OS? if so consequences –

- No healthy environment
- May not be efficient
- Not portable
- Degree of multi tasking will be limited.

TODO:

Available OS in today's market -- do some survey Customized OS for constrained portable devices more sensitive services -- power consumption thermal issues

Typical OS Services

- **Process Management:** process creation, terminate, scheduling, life cycle signaling, IPC(data exchange, synchronization).
- **Memory Management:** available, free, allocate, deallocate, non contiguous allocation expansion of memory(swap area, virtual mem), Paging.

Typical OS Services

- **File system management** (logical) file ops, perms, ownership, linking and many more
- **Disk/storage management** (physical) I/O Management (storage is kind of subset)
- **Network management** (physical, logical)
- Authentication, Authorization, Security etc.

Typical system architecture

- CPU
- Memory (primary)
- I/O Devices including storage (peripherals)
- System Bus

Central Processing Unit (CPU)

- Execution core
  - (Arithmetic Logic Unit, Control Unit),
- Registers, clock, caching.

Special Purpose Registers

Program counter(PC)/instruction pointer(IP) address of next instruction to be execute program Status word(PSW)/FLAGS register status & control of CPU ops

Stack pointer, frame/base pointer locate stack frame of current execution(application, function)

General Purpose: accumulator, implicit operand for some operations