Operating System 16-01-2023

Bridge b/w user program and hardware.

operating system Objectives

→ Convinience
→ Efficiency
→ Ability to evolve

Layers of computer System

(→) interact

interrupt

Application program → End user
utilities ⇒ facilitates other programs } programmer
Operating system
Computer hardware → OS designer

System call

Services provided by OS 19/01/23

- Program development (editor & debuggers)
- Program execution
- Access to I/O devices
- Protected access to files
- Error detection & response (Internal & External error)
  - Memory error
  - device failure
- Software errors (arithmetic overflow, access forbidden memory locations)
- Error recovery
- Accounting
  - collect statistics
  - monitor performance
  - used for biling users
  - used to anticipate feature enhancement

Operating System:

from application : System Calls
from hardware : Interrupt.
Both calls have different responses.

Kernel: • resides in main memory (RAM)
• portion of OS.
• contains independent part (Code for System calls) and dependent (device drivers)
• Maintains OS state
• Executes in supervisor mode
• resides on the top of main memory

Evolution of OS

• Serial processing (One operation at a time)
• Simple batch system (monitors)
• Job control language (JCL)
  • Special type of progm language
  • provides instruction to monitor what compiler & data to use

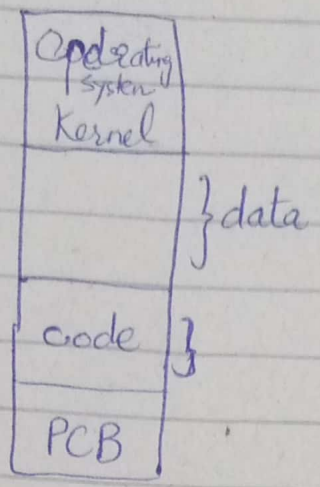
Uniprogramming (wait for Input/output)
Multiprogramming (switch to other program).
Time sharing (multiple users can use the system at a same time).

Main OS Concepts

• Processes
• Memory Management
• Info protection & security
• Scheduling & resource management (deadlock)
• System structure

Process: program under execution
Program; piece of Code; exe file
→ program can have multiple instances (ek program kai dafa run ho sakta he).

PCB ⇒ Program Control Box



context
data
program (code)

Process :

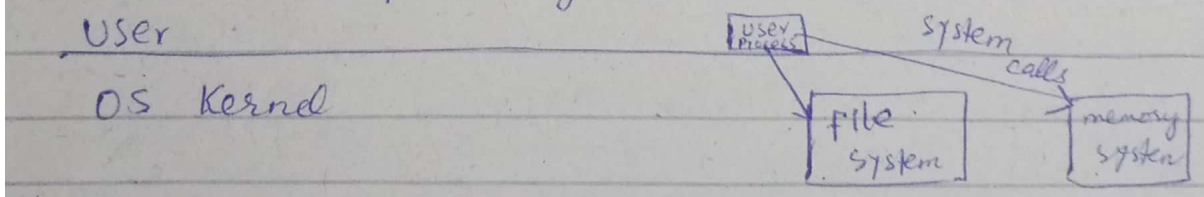
→ three components
- Executable program
- Associated data needed by program
- Execution context of program
  ↳ all info OS need to manage process

Memory Management

- Virtual memory
- File System (implements long-term store
- Paging
- Security
- Process isolation
- Long term Storage

Traditional OS

- monolithic layered System (heavy hota he her cheez add ho jati).
  → good performance but rigid
  → one / N layers all executed in Kernel-mode



Microkernal : (assigns only a few feature)

→ address space → basic scheduling → IPC (interprocess communication)

Flexible but not efficient

Characteristic of Modern OS:

→ Multithreading ⇒ process is divided into threads than can run simultaneously

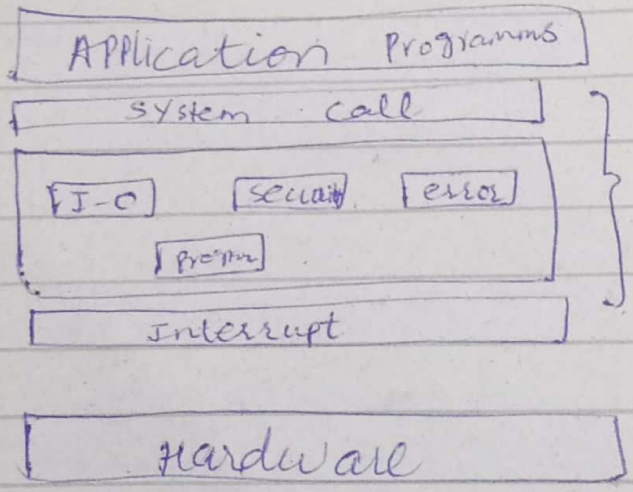
→ Threading ⇒ dispatchable unit of work
- executes sequentially and is interruptable.

Symmetric multiprocessing
- multiple processors
- all processors perform same functions
- Same main memory / IO

Chapter 02 Abraham

Operating System Structure

- Interface provide krta he.
  - ↳ can be Graphical
  - ↳ can be command line
- Program execution
- I/O operations perform krta
- Communication
- Security
- error detection



MKDIR → Make File
rm – remove
mv – move
cp ⇒ copy

Grap

GUI
CLI
Voice
Text

Shell scripting ⇒ Commands
- sh

different commands are used to execute

programs. like gcc for c, g++ for c++ & java for java.

System Calls:

programing interface that help programmer to interact with interface.

API ⇒ application program interface

Win-32 API (Window) linux, Mac ⇒ Posix

Way to access API :-

→ Key, Methods, parameters and their datatype.

In API, we have no link with internal program.

System calls are around the services of operation system.

26-01-2023

Resource allocation, Accounting, Protection and security.

API → SYSTEM CALLS

man read ⇒ to get the function of read

Types of system calls

create process, abort, terminate process

variation

free memory

Command & external

Emulation can allow an OS to run on non-native hardware.

Simple structure MS-DOS

More Comple

Layered

MicroKernal

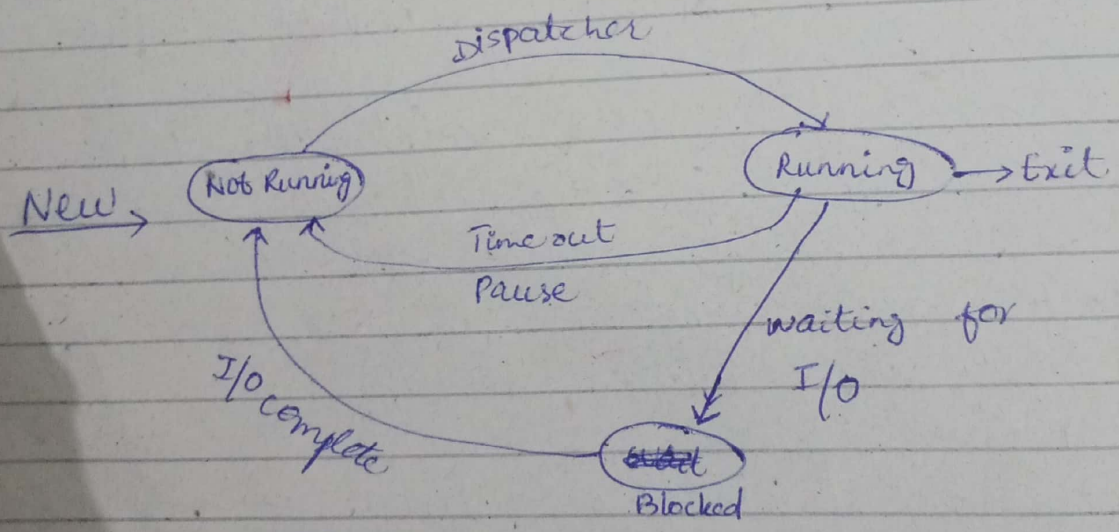
profiling is periodic sampling of instruction pointer to look for statistical trends

30-01-2023

Process description & Control
Chapter 03

Two state process:
Process may be in one of two states
- Running
- Not Running

Process creation*
- Submission of batch job
- Create to provide service like printing
- user logs on
- process creates another processes.
  - parallism achieve krni ho
    ↳ task divide kr k parallel execute k
    or task aik dosry per depend na h



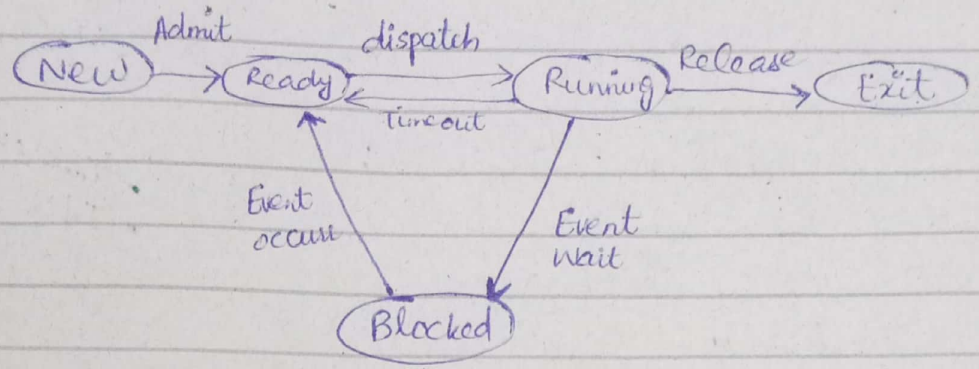
Three state process

Running Queue implement nahi hoti
Not Running ⇒ Ready (queue)
Blocked.

Five State

- Running
- Ready
- Blocked (Waiting for I/O)
- New You can
- Exit

Blocked state mein multiple process so sakty hain
Priority queue will be used.



Suspended Processes

Two states:

→ Blocked Suspended hard disk mein
→ Ready Suspended

Processor involve nahi hota input or output mein
I/O slow hoti he
Processor fast hota he

Reasons

Swapping
Time
Parent process request
Other OS reason
Interactive user request

Memory Tables

Allocation of main memory
secondary
Protection attributes for access to shared memory region.
Info needed to manage Virtual memory

I/O Tables

I/O device is available

File Tables

- Existence of files
- Location on secondary memory
- Current status
- Attributes
- Sometimes this info is managed by FMS. (File management system).

- time out
- I/O

Process Tables

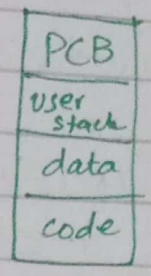
Where process is located
Attributes necessary for its management
- Process ID
- Process state
- Location in memory

Process Control Block

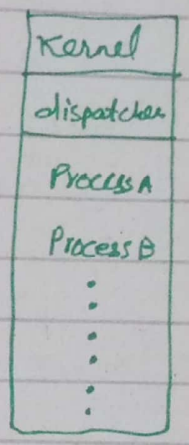
- Process Identification
- Processor State Information
  - User visible registers
  - Stack pointer
- Register Fast hota he
- Process control information

Operating System

Shared address space mein global or static variables aaty hain.



Content of processor registers
→ User visible register
→ Control and Status registers
→ Stack Pointer
Program status word (PSW)



Modes of execution

1- User- mode
- Less priviledged mode
- User programs typically execute in this mode.

2- System mode, kernel mode
- More priviledged mode

Process creation

→ Assign a unique process identifier
→ Allocate space for process
→ Initialize process control block
→ Set up appropriate linkages.
→ Create of expand other data structure
- maintain an accounting file

* When to switch a process

→ Clock switch → trap (error
→ I/O interrupt → Supervisor call
→ Memory fault (OS problem)

* memory address is in virtual memory so it must be brought into main memory.

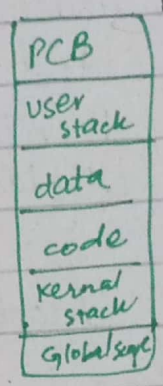
Execution of OS

- Non-process Kernel
- Execution within user processes

1-

- Execute ~~process~~ kernel outside of any process
-

3- Process-based OS: 01|02|03



Process Creation

Child process pid = 0
parent process pid = 1 / +ve
negative represent failure.

Process Scheduling

Short term
Long term → block to ready
Mid term → Swapping

Interprocess Communication

- Independent process
- Cooperating process (Can affect or can be affected by other process)

* Information sharing
* Computation Speedup
* Modularity
* Convenience

} Reasons for cooperating processes

Models of IPC

→ Shared Memory
→ Message passing

Cooperating processes need IPC.

Pipe ⇒ process A and B are at same machine.

Socket ⇒ process A and B are not running at same machine.

→ Shared Memory mein kernel involve nahi hota.

Unbounded buffer ⇒ No limit

Bounded buffer ⇒ Limited bytes

Buffer is the part of process.

Shared memory outside the process memory.

Shared Memory

Write

Read

Memory sharing

Message passing

Send (Message)

Recieve (Message)

Implementation of Communication link

physical → Shared Memory → Hardware bus

→ Network.

Logical → Direct or indirect → Synchronous or asynchronous →

Mailbox ⇒ Indirect communication

Blocking is considered Synchronous.

Non-blocking is considered asynchronous.

Blocking send or recieve level pr hoti he.

Communication

Shared Memory
→ Same machine
→ Faster

Message passing
→ Pipes • Named • Ordinary
→ sockets
→ LPC → RPC

* Consumner reads from the one end.
* producer writes to one end.
→ Producer ⇒ parent process
→ Consumer ⇒ Child process
1 for writing
0 for reading

Ordinary (Parent child relationship involve hota he.
→ Kernel
→ pipe kisi be process k address space pr lie nahi krta
→ Kernel level a object hota he.

Named pipes
• More powerful than ordinary pipe:-

Threads Libraries

→ Don't use static and dynamic variables.
→ Void pointers used for data whose data type is not known.

Two types of process
• Cooperative
→ Variable/Memory sharing
→ critical Section
• Independent

Sol. to critical section problem
• Mutual Exclusion (more than one).
• Prograss

- Bounded Waiting

Preemptive process ⇒ allows one process's execution pause and starts another process.

Peterson's Solution

- Applys only on two processes cooperative
- Syncronization ki problem solve krny ka solution he.
- Two variables are used for entry & exit. int turn, Boolean flag[2]
- Humble Algorithm

02-03-2023

OPERATING SYSTEM

* BUSY WAITING

Hardware
- Test & Set
- Swap

Software
- Lock
- Semaphore → Binary → Counting

Set and Test Instruction:
→ Return the same value that is passed in parameter.
→ Executed p atomically.
→ Set the new value of passed parameter to 'TRUE'.

Mutex Locks (Imp)

Semaphores (Imp).

→ Multiple process can enter CS in counting semaphore.

→ Binary Semaphore has two values 0 & 1.

- Acquire or release k liye functions hain.
- Wait ()
  - P
  - down
  - decrement
- Signal ()
  - V
  - up
  - increment

Bounded buffer

mutex ⇒ binary Semaphore ⇒ mutual exclusion

Full = 0 ; Empty = n ⇒ counting Semaphore.
↳ control overflow & underflow.

06-03-2023

Process Synchronization

Problems

→ Deadlock:
All processes are waiting for an event and no one is stopping to wait.

→ Starvation: Due to semaphore → blocked for a long time
longer period k liye wait
↳ indefinite time.

Readers & Writers problem
→ Two (Semaphores) functions

Mutual Exclusion
how many process read data at a particular time.

5- You can combine
6- It's good for business
7.

Dining-philosophers problem

- thinking
- Eating
- Acquire (Fi) left fork
- Acquire (Fi+1)%N
- Release (Fi) left
- Release (F(i+1)%N) Right

Semaphore No of forks ki array ho gi.
→ every semaphore (waits) represents fork.

Monitors

→ Abstract data type
→ Solution

Chapter 07
Deadlock

Process ⇒ circle
⇒ Rectangle
→ Resource ⇒ Square.
CPU ⇒ dot

Resource allocation graph

- Directed graph
- Two edges → request edge → Assignment edge
  ↳ from process ↳ toward process

agar graph mein cycle ho tu deadlock ka indication hota he. (100% surity

no cycle ⇒ no deadlock

Banker / Safe algorithm

Allocation Max need Available Remaining

→ conditions of deadlock

Deadlock characterization

Mutual Exclusion, hold & wait,
No preemption, circular wait.

Process ⇒ circle ○
Resource type ⇒ □

preEmption ⇒ execution disturb hoti he.

⇒ Banker algorithm detection or avoidance k liye use hota