Date : / /
Page No :

Operating System

(i) Computer System Structure

(a) Computer System Structure divided into 4 parts

(i) Hardware (ii) operating system (iii) Application programs (iv) users

(b) Abstract view of computer system -

user1 user2 user3 ..... user N

Application program
OS
computer hardware

(ii) operating system

(a) what is OS
(i) It is a program
(ii) Act as - Intermediary B/w user and computer

(b) Goal of OS
(i) Primary goals - convenience, user friendly
(ii) Secondary goals - Efficiently use hardware

(c) Function / components of OS

(i) Process manager
(ii) main memory manager

(iii) Secoundary memory manager
(iv) File management
(v) I/o device management
(vi) Network management
(vii) Security and protection

(d) Services of OS

(i) Program execution ①
(ii) I/o operations ④
(iii) File system manipulation ③
(iv) communication ⑤
(v) Error detection ⑦
(vi) Resource allocation ②
(vii) Protection and security ⑥

(e) Main Parts of OS

(i) Kernel
(ii) Shell

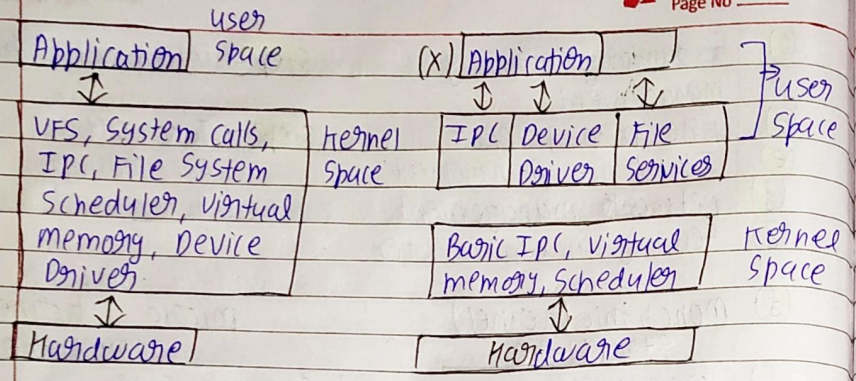
Kernel	Shell
(i) Interacts with H/w and Shell	(i) Interacts with Application and Kernel
(ii) Kernel acts as a Resource manager such as	(ii) Shell acts as a command interpreter such as
(a) Process manager	Step1 - Take command from user
(b) main memory management	Step2 - interprets them
(c) Secoundary memory management	Step3 - Take action accordingly
(d) File management	
(e) I/o device management	
(f) Network management	
(g) Security and protection	

(f)

Monolithic (Kernel) Structure	micro Kernel Structure
(i) large in size	(i) Small in size
(ii) Fast in execution	(ii) Slow in execution
(iii) Adding new services is hard	(iii) Adding new services is easy
(iv) Debugging is hard	(iv) Debugging is easy
(v) complex to design	(v) Designing is easy
(vi) For example - linux, DOS, Unix	(vi) For example - mac OS, window NT, QNX
(vii) Defination - All OS services are stuffed in Kernel space	(vii) Defination - All OS essential services are stuffed in Kernel space, All OS non essential services are stuffed in user space
(viii) Problem - Security Issues	(viii) Problem - NO security Issue
(ix) When Kernel performing I/O	(ix) When Kernel performing I/O
(a) NO Need of context swith	(a) Need of context switching
(b) NO Need of message passing	(b) Need of message passing

Date : / /
Page No :

(x)



(g) Re-entrant Kernel

(i) contain " executable code"

(ii) Many process or Thread can execute at same time with effecting each other in Re-entrant kernel

(iii) Problem occur in Re-entrant kernel (Because of Data sharing)
(a) Data corruption
(b) Data inconsistency
(c) multi-different copy of global data

(iv) To slove these problem, Re-entrant kernel use
(a) Re-entrant Function
(b) locking mechanism

Re entrant Function	Non-Rentrant Function
Function in which process or Thread modify "local data"	Function in which process or Thread modify "global data"

Date : / /
Page No :

(#) SPOOLING

locking mechanism

Re entrant Function	Non Rentrant Function
many process or Thread can execute at same time in Re-entrant Function	One process or Thread can execute at same time in non-Rentrant Function.

(h) Dual mode of processor in OS

(a) Process, Thread or instruction execute in 2 modes
(a) USER mode
(b) Kernel mode

USER mode	Kernel mode
(i) USER mode Process, Thread, instruction can not directly interact with H/w	(i) Kernel mode Process, Thread, instruction can directly interact with H/w
(iii) Mode Bit = 1	(ii) Mode Bit = 0
(ii) Private virtual address Space	(iii) Single virtual address Space
(iv) In user mode Non-Privileg instruction executes	(ii) In Kernel mode Non-Privileg and Privileged instruction can execute.

(v) Exception or error can crash only particular Application

(v) Exception or error can crash entire OS

(vi) Also know as - Non-Privileged mode, Restricted mode

(vi) Also know as - Privileged mode, Monitor mode, System mode, Supervisor mode.

(6) Mode Switch - Process, Thread and instruction switch in 2 modes (Kernel and user mode)

Mode Switch Done By 3 ways
(i) System call / Supervisor calls
(ii) Trap → software interrupt
(iii) Interrupt → Hw interrupt

Mode switch use -
Provide Protection and security to -
(i) OS (ii) user program

Mode switch is a decision of OS

Mode Bit - Indicate in which mode, instruction is executing

Booting time - Hw start in Kernel mode
Then OS load
Then computer CPU / OS switch to user mode

Context Switching	Mode Switching
(i) Take more time	(i) Take less time

(7) Spooling

(i) Simultaneous Peripheral operations Online (store)
(ii) Principle - Refer to Putting Job in Buffer
Process - (a) Input Stored in disk (Buffer) First
(b) Input transfer Blw Disk → memory
(c) Input transfer Blw memory → CPU
(d) CPU Process Data and generate output
(e) Output transfer Blw CPU → memory → Disk
(f) Output transfer to Output device

(iii) Advantage
(a) Increase CPU utilization
(b) (Buffer / Disk) Provide "waiting station"
(c) Resolve Problem - Speed mismatch Blw different Device
(d) Processor is able to overlap (CPU) (I/O operation and Processor operation For Different Job)
(e) For example - Print Spooling.

CPU

memory

Input Output

Disk

Date : / /
Page No :

(J) component of OS

(i) Process management

(a) Creating and deleting → user and Sytem Processes
(b) Suspending and Resuming Processes
(c) Provide mechanism For
  (i) Process synchronization
  (ii) Process communication
  (iii) Deadlock handling

(ii) File management

(a) Creating and deleting - Files, ~~deleting~~ directories
(b) Manipulating - Files, directories
(c) mapping Files into secoundary storage
(d) Backing up Files on stable storage media

(iii) Secoundary memory management

(a) Free space management
(b) Storage Allocation
(c) Disk Scheduling

(iv) main memory management

(a) Keep track of Primary memory
(b) Allocating and deallocating memory space
(c) Deciding which processes are to loaded in main memory

1/2/19

Date : / /
Page No :

(v) I/o management

(a) Hide detail of I/o device from user
(b) Interfacing with device drivers
(c) caching, Buffering and Spooling of I/o device

(K) Evolution of OS

(i) Serial Processing System
(ii) Batch Processing System
(iii) Multiprogramming System
(iv) Time Sharing system
(v) Real Time System
(vi) Distributed operating system
(vii) Multi Threading operating system

(i) Batch Processing system

Principal - Job with similar needs were Batch together and run as a group

Advantage - File and I/o management is simple
No manual instruction required

Disadvantage - No interaction B/w user and computer
Execute one program at a time
No context switching
No way to prioritise process
CPU is mostly Idle.

(ii) multiprogramming system

Date : / /
Page No :

Execute one program at a time.

Principle - ~~concurrent execution of multiple program~~

Process - Job scheduling, CPU scheduling, context switching

Advantage
increase CPU utilization → principal
Decrease waiting time
CPU Never idle (until no process exist)

Disadvantage
~~Difficult scheduling~~ (i) CPU scheduling is required
~~memory fragmentation~~ (ii) memory management " "
~~Paging (Non contiguous memory allocation)~~
(iii) highly complex

(iii) Real Time Operating System

Principle - Job Perform in well defined and fixed time constraints otherwise System fails

Type -

Characteristic	Hard Real Time	Soft Real Time
Response Time	Hard required	Soft desired
control of Pace	Environment	Computer
Safty	often critical	Non-critical
Size of Data	small / medium	large
Redundancy type	Active	check point - Recovery
Data integrity	short term	long term
Error detection	Autonomous	user assisted
Peak load performance	Predictable	Degraded

Date : / /
Page No :

(iv) Time sharing or multitasking OS

Principle Execute multi process together

Process - CPU switch B/w Process so quickly given an illusion that All Process executing at same time

Time Sharing = ~~Multi programming~~ + ~~context switching~~
OS OS OS

Multi programming Time sharing

Advantage
quick response
Reduce CPU idle time → principal

Disadvantage
Problem of reliability and data communication.

(v) Multi processor OS

Principle - OS with more then one processor

Process - multi process run Parallely at same time

Type - Symmetric Asymmetric

(i) All Processor have same Architecture
(i) All Processor may have same or diffrent architecture

(ii) All Processor communicate with Shared memory
(ii) No need of communication. All Process handle By master Processor

(iii) If processor Fail. the

Date : / /
Page No :

capability of the system reduce

(iii) If master Processor Fails slave is become master Processor
If slave Processor Fails its task is switch to other Processor.

Multiprogramming	Multitasking
(i) Single CPU	(i) Single CPU
(ii) increase CPU utilization	(ii) increase CPU utilization + Increase responsiveness
(iii) context switching is used	(iii) context switching + Time sharing is used

Multiprocessing	Multiprogramming
(i) Multi CPU	(i) Single CPU
(ii) Parallel Processing is used	(ii) context switching is used
(iii) more effective	(iii) less effective
(iv) more expensive	(iv) less expensive
(v) less time to Perform Job	(v) more time to Perform Job

(c) System call
(i) Provide interface to the services available in OS
(ii) written in higher language (C, C++)
(iii) Programmatic way in which a computer Program requests a service from the kernel of OS.

Date : / /
Page No :

Application program interface -

The API specifies a set of function that are available to an application programmer, including the parameter that are passed to each function and the return values the programmers can expect

(m) Structure of OS
(i) simple structure (ii) layered structure
(iii) micro kernels (iv) modules
(iv) monolithic structure

(i) Simple structure -
(a) Do not have well defined structure
(b) For example - MS-DOS
(c) Advantage
(i) Better Application performance (Reason - less interface B/w H/w and application Program)
(ii) Small, Simple and limited system
(iii) Easy for kernel developer to develop such an OS
(d) Disadvantage
(i) If one user program fail, crash entire system
(ii) No data hiding and No boundaries (B/w modules)
(iii) very complicated structure. → ☹ ??
(iv) Direct access to hardware.

(ii) layered Structure
(a) OS Divided into number of layers
(b) Example - Windows NT
(c) Advantage
(i) very easy to perform Debugging and system verification
(ii) very easy to make change in OS
(iii) Promote modularity, abstraction and no direct access to hardware

(d) Disadvantage
(i) Performance degraded as compared to simple str.
(ii) Designing required "careful planning"

(e) Rules
(i) Outermost layer → user Interface layer (must)
layer 4 → I/O Buffers
layer 3 → Process management
layer 2 → memory management
layer 1 → CPU scheduling
innermost layer → Hardware (must)

(ii) Each layer must have a specific function
(iii) A layer can access all the below layers
A layer cannot access all the above layer

(iii) Modules
(i) Best method to design OS
(ii) Involve using oops concets to design OS,
For example - Solaris OS
(iii) Rules -
(a) Core Kernel - has only set of core components
(b) Other services - Modules dynamically loadable to Kernel at run time or boot time