"Components of O.S"

Cache Memory :-

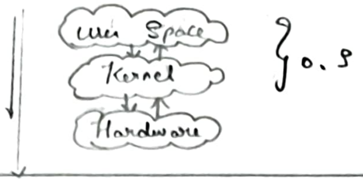
It is a Small, high Speed Storage area in a Computer the Cache is a Smaller & faster memory that Stores Copies of the data from frequently used main memory locations.

- * C.M is much more fast than main memory (RAM).
- * When CPU need data it 1st Check C.M if not main memory
- * reduce the average time to access data.

*) Components :-

i) Kernel

ii) User Space.



1) User Space :-

- * where application Software runs, app's don't have access to underlying hardware. It interacts with the kernel
- * Provide Convinent environment for user apps.
- * Eg-1) GUI → Graphical user Interface

  2) CLI → Command Line Interface (Terminal, Powershell)

  mac windows

* Mkdir (CLI).
* Newfile (GUI).

2) Kernel :-

- * Heart of O.S/Core Component.
- * It interacts with direct hardware and perform more Crucial tasks.
- * Very first Part of os to load on Start-up.

*) Shell → Known as Command interpreter, is that Part of o.S that receives Commands from user and get them Executed.

→ Functions of Kernel:

I) Process Management:

- * Scheduling Processes & threads on the CPUs
- * Creating & deleting both user and System Process.
- * Suspending & deleting (resuming) Process.
- * Provide mechanism for Process Synchronization (or) Process Communication.

II) Memory Management:

- * Allocating and deallocating memory Space as per need.
- * Keeping track of which Part of memory are Currently being used and by which Process.

{Stores directory in hierarchical way in a Tree}

___/___/_____

3) File Management :-

- * Creating & deleting files.
- * Creating & deleting directories to organize files.
- * Mapping files into 2° Storage
- * Backup Support onto a Stable Strong media.

4) I/O Management :-

* To Manage and Control I/O operations & I/O devices.

Print Spooling & Mail Spooling ← i) Spooling → 2 Jobs → differing Speeds (data copy b/w 2 devices)

Youtube Video buffering ← ii) Buffering → within 1 Job (loading of data).

← iii) Caching → Memory Caching, Web Caching

→ Types of kernels :-

1) Monolithic Kernel :

- * all funⁿ (4) are in kernel itself.

{demerits} ←
- * Bulky in Size.
- * Memory required to run is high.
- * Less reliable, one module Crashes → whole kernel is down.

{merits} ←
- * High Performance as Communication is fast. (No overhead).
- * Eg → Linux, Unix, MS-Dos.

[user mode (UM) and Kernel Mode (KM). Switch → Software interrupts]

2) Micro Kernel :

- * only major functions are in kernel;
  - i) Memory Mgmt
  - ii) Process Mgmt
- * File & I/O Mgmt are in user Space.
- * Smaller in Size * more reliable * Most Stable.
- * Performence is Slow * overhead Switching b/w K.M & U.M
- * Eg → L4 Linux, Symbian O.S, MINIX.

Symbian O.S ↓ {Nokia keypad}

Process
↓
i) Independent
ii) Cooperative

q) How Communication Happens b/w usermode & kernel mode :

⇒ Inter Process Communication (IPC), Done by "Shared memory + message Passing".

<2 Process Executing independently, having independent memory — / — / — Space (memory Protection) -> they need to Communicate for Some work

3) Hybrid Kernel :

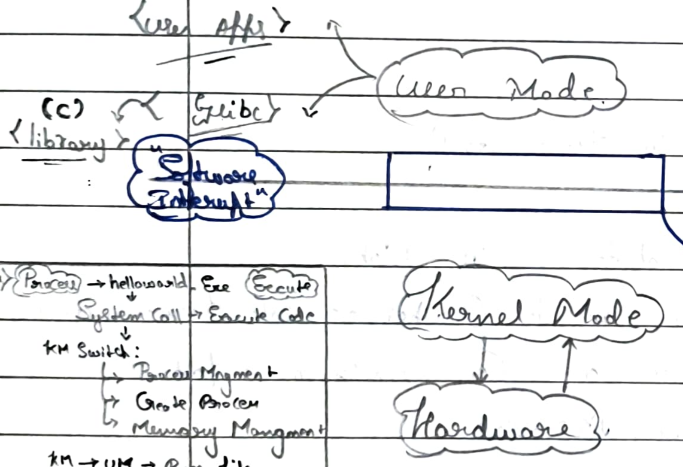
- * Combined approach
- * Advantages of both worlds (F.M in user Space & rest in kernel).
- * Speed and design of mono
- * Modularity and Stability of micro.
- * Eg → Macos, Window NT 7/10.
- * IPC also happens but lesser Overheads.

4) Nano / Exo kernels

System Calls

⊙ How do apps interact with kernel? → By System Calls.

⊙ S.C is a Mechanism using which user Program Can request a Service from kernel for which it does not have the Permission to Perform.



i) Process → helloworld.Exe Execute
↓
System Call → Execute Code
↓
KM Switch :
↳ Process Mngment
↳ Create Process
↳ Memory Mangment
KM → UM → Run file.

→ System Call Interface [SCI]
↓
{ Search for mkdir Corresponding code in C from Pool & give it to kernel }

Eg → i) Mkdir Command
↓
→ It indirectly Calls kernel and asked the file mgmt module to . Create a New directory
↓
→ Mkdir interacts with kernel using System Calls.

ii) Creating a Process : User Executes a Process (US). ← return to US
↓
Gets System Call (US).
↓
Execute System Call to Create a Process (KM).

"A Software Interrupt is a mechanism used by Software to Signal the Processor that it needs its attention".

"way for a Program to request a Service from o.s / to Signal event".

__/__/____

*) Transitions from US to KS done by Software interrupts.

*) Systems Calls are implemented in "c".

* user Programs → donot have Permission to Perform I/o operations.
→ donot have access to Communicate with other Program

∴ S.C are the only way through which a Process goes from user mode to kernel mode.

→ Types of System Calls:-

1) Process Control:
* end, abort * load, Execute * wait for time.
* wait event, Signal event * allocate & free memory
* Create & terminate Process * get & Set Process attribute

2) File Mgmt:
* Create & delete file * Read, write, Reposition.
* open & Close file * get & Set file attribute.
↳ (Security Process).

3) Device Mgmt:
* request device, release device * Read, write, Reposition
* get & Set device attribute * logically attach/detach.

4) Information Maintanence:-
* get time or data, Set time or data.
* get/set System data * get/set Processes, file etc

5) Communication Management:-
* Create, delete Communication Connection
* Send, receive Message * transfer Status information.
* attach / detach remote devices.