1

September 9, 2025

[ ]:

l1 to l3 cache

Binary code

fetch line by line

decode

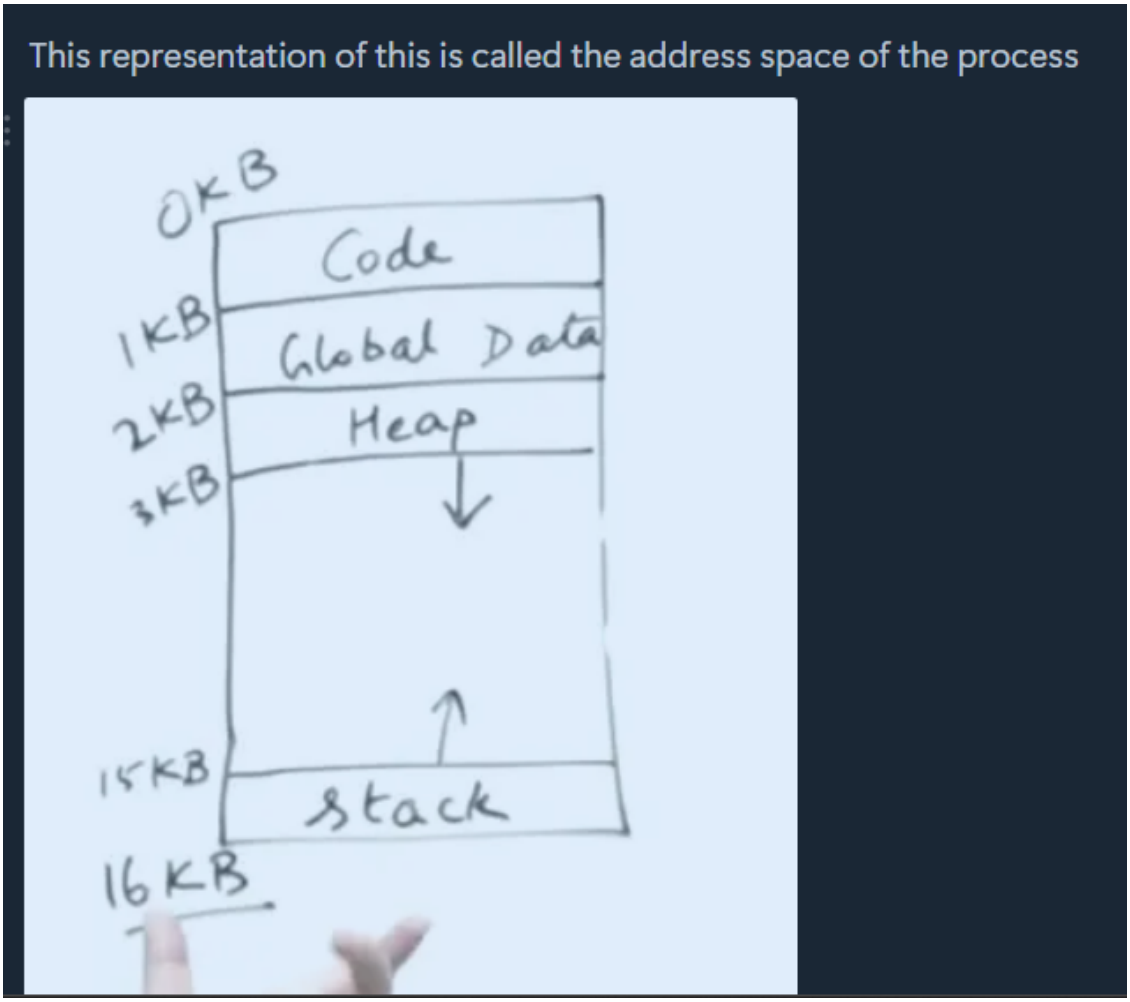
execute

In detail by naveen notes, where?

Text
compiled code, loaded from memory

Data
Global and static data

Stack
local variables, function args and return values



stack pointer
current line that is being executed,

frame pointer
current function ends, next function start

#Components of OS 1. kernel

kernel
1. A kernel is that part of the operating system which interacts directly with
the hardware 2. Heart of OS/Core component 3. Very first part of OS to load on start-up Apps intract with kernal : using system calls (process is moved to kernal stack from user stack)

0.0.1 Process management

1. Scheduling
2. Creating & deleting both user and system process
3. Suspending and resuming processes
4. process synchronization

5. process communication

0.0.2 Memory management

1. Allocating and deallocating memory space as per need
2. Keeping track of which part of memory are currently being used and by which process.

0.0.3 File management

1. Creating and deleting files
2. Creating and deleting directories to organize files
3. Mapping files into secondary storage

1 IO management

1. Buffering (data copy between two devices)
    1. Within one job.
    2. Youtube video buffering
2. caching
    1. Memory caching
    2. Web caching etc
3. spooling
    1. Within differing speed two jobs
    2. Print spooling and mail spooling

(printer store the data in some area and then print it slowly)

2 User space

Where application software runs, apps don't have privileged access to the underlying hardware. It interacts with kernel

Monolithic kernel
Linux, Unix, MS-DOS

All functions are in kernel itself
Bulky in size
Memory required to run is high
Less reliable, one module crashes -> whole kernel is down
High performance as communication is fast
Less user mode, kernel mode switching overheads

Micro kernel
Symbian OS

Only major functions are in kernel 1. Process mgmt 2. Memory mgmt

Rest in User space File mgmt. IO mgmt.

smaller in size
More Reliable and More stable

Performance is slow (Overhead switching b/w user mode and kernel mode)

Hybrid kernal
MacOS, Windows

same as above except
IO in kernal

system calls
calls to os to get resources
mmap for available memory
scanf is wrapper over read()

Transitions from US to KS done by

1. hardware interrupt

urgent interrupt or non urgent

keyboard and mouse, creates an interrupt

2. software interrupts

traps/exception/faults
special kind of software interrupt

division by 0

Ilegal memory access

1. Degree of multi-programming: The number of processes in the memory
2. LTS controls degree of multi-programming