14 March Lec 12 Thursday

Multi-Tasking (CN / ICT Course)
- ↳ At a time, multiple tasks perform kna
- ↳ To do multiple things at a same time within a computer.

Is se milta julta concept hai
↳ Multi-Threading
  ↳ To do multiple ~~things~~ tasks at same time within a program.

Thread?
  ↳ It is light weight process / program / procedure.

Address space → OS mai parhe the concept.
↳ Eik progrm ko execution ke liye jo space, RAM mai jo area chahiye hota jahan code, data store kate.

Progrm memory mai load kate tou space require krta → address space.

Eik aur instance → uske apne address space.
eik aur instance → uske apne address space.
→ Agar pore instance load krne lag jaye, sharing krwde tou optimize krskte - -

It actually shares address space of parent process, apni share ni kr rahe hota... → Is liye ye light-weight process.

جو parent ہے اپنا address space [illegible] اور [illegible]
- share address space [illegible]

Benefit of using thread:
Efficient / optimized use of memory
Lekin sharing to si tou bht si problms hogi. Deadlock.

starvation, synchronization problems at thread ke sath

Threads in Computer (2)

OS/Kernel level threads
→ Sirf ye CPU pr execute hote

User level threads
→ They will not directly execute on CPU.
→ They map to your kernel level threads phir OS unko CPU pr execute krwata..

How

- How to create these user-level threads in Java?
- How they will be mapped on your system/kernel level thread taki CPU pr execute kr sake?
- Banaye ge kaise Aur execute kaise?

Banaye ge kaise threads?

First program we wrote → We were using threads at that time

Java mai simple se program bhi likhte, tou Java us program ko run krne ke liye thread lade. Aur jo pehle pehle kahe — usko $\boxed{\text{main thread}}$ kahte.

→ Is mai jakr ye statement likhe

System -- (Thread.currentThread());

Thread → built in Java class exists in java.lang.* pckg

currentThread() → static function of this class (It will show current running thread)

Jb isko print karwaye ge to is trd se info ayegi.

→ [main, 5, main]

name of thread ↲ (main)

5 → Thread jo create huc uski priority → Scheduling Algorithm

main → Parent thread jis ne isko create kiya

→ yahan main likha h ye pehla thread hai.

Scheduling Algorithm
How process will be scheduled on CPU?
(LIFO, FIFO, ---)

Q. Java ke threads CPU pr kaise execute krege?

→ Onko map krne Kernel level threads pr aur

→ [JVM] unko map krta based on priorities.
↳ transparently maps your user level threads to system level threads based on their priorities.

2nd lec
↳ Platform independence.

Java provides JVM for each OS?

Why not one JVM for all OS?
↳ One of reason is priority based threads.

Total priority levels = 10.
1 → min priority 5 → center ---
10 → highest

Ye CPU pr execute ni krega user level ---

Windows NT → 7 total level
$2^{32}-1$ → Solaris mai levels.
↳ Difference dekhe itna zyad.

آپ نے ان 10 کو 7 پر یا $2^{32}-1$ پر map کرنا

JVM ہر OS کے لئے الگ ہے۔
maps

→ Java mei jitne bhi threads create krege,
normal / default priority = 5 ke sath hoge.

→ Parent process should reside in memory until
child process terminates.
[illegible] process ہر ایک memory میں ہے۔
child اپنا [illegible] share
parent --

Agar parent process priority ↑ & it terminates
tou eik temporary parent process (init process)
assign kate child ko..
Kuch ~~process~~ programs allow ni krte ke parent terminate
kare child se pehle lekin Java allow krte
tou OS ki taraf se ek dummy
parent process assign.

⇒ Main thread is responsible to execute main function
control entry point

اگر further thread بنانے ہیں تو main سے
ہی بنتا ہے۔

Main thread baqi threads ko create krwande.
Tou hamare child thread jo hum brayege
unke parent main thread hoge.

[t1, priority, main]
Humare thread jo hum ne create kiye
Parent
Q k main ke bad bn age

User level threads ke parent → main thread.

Eik prgrm mei 2,3, n tasks ||(parallel chal rahe hoge)
→ Jitne CPU's, utne multiple threads ki execution chal rahi hogi.
→ 2001 → single CPU → tou vahega jo multi-tasking mei hote..

How to create User-level threads?

Try kr run karo game & music.

music ruk ruk kr chale ge

↙

Switching of CPU..

↑

Game mei interruption

Thread → complete process

4 CPU → 4 threads at a time.

Har thread eik CPU pr

↳ multiple, powerful CPU | Ja T ←

[illegible] blem U/o switching

4 threads diff CPU pr run kr rahe..

Thread 1, 2, 3, 4 alag O/P aaí..

lekin kisi aur ko system pr maybe CPU 2 pehle free

ho tou 2 ki O/P pehle..

→ Eik progrm ko multiple CPU/machine pr run tou slight variation in sequence depending ke kaise assign hue...

→ Sare threads ko main create. By default main ki priority hi child ki hogi. Aur hum change krskte priority (1-10).

↳ out of range jayege tou exception.

→ 2 ways.

Name of interface

↑

1) Using Interface (Runnable):

Name of class

↙

2) Using class (Thread):

① You need to create a class

class Test implements Runnable

implements → contract ka note

Test → Responsible to provide function implementation of this interface

Runnable → Only one function. run().

② - Override this function.

Override void run()

This run function is similar to main.

run function → Control entry point of your thread.

main → execution program ki start jahan se hoti.

- Wherever your thread will get CPU, it will run its run function.

→ t1 ke through file read karo chahte ho ye file read run function mei krege.. Aur jab CPU milega t1 ko, tab vo file-read krega.

CPU mile → start execution of run process

③ U need to create object of Thread class

thread t = new Thread(________).

Thread → java.lang. pckg

Here we need to pass an instance of runnable type. Vo object jis ka pass thread ke control entry point hai vo pass karo.. Jis ke pass

→ Main thread

ts ke control entry point hai.

<u>Kya pass kia yahan?</u>

Is-a relationship.

→ Test is a runnable type

Test type ka object pass.

Thread t = new Thread (new Test ());

→ Thread create kiya.

Eik instance / object create hoga RAM mei.

(4) ⇒ You need to push this thread in <u>Ready Queue</u>.

CPU process wil [illegible]

[illegible] [illegible] wo [illegible]

For this purpose, you need to make a call.

t.start( );

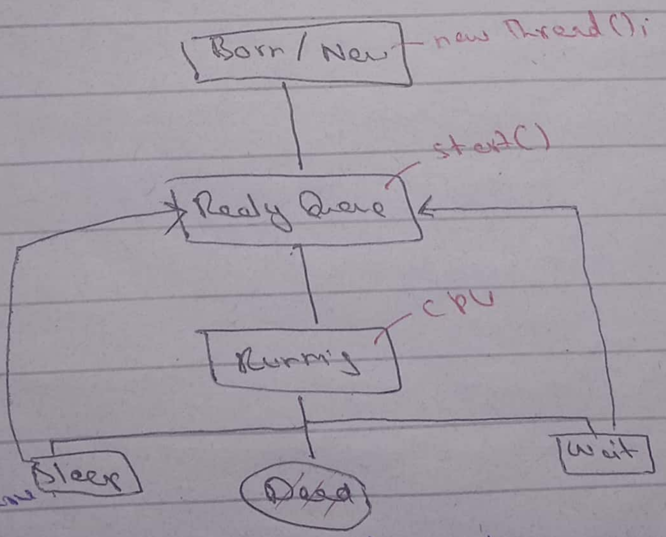
Wherever you will call start, it will push it in

<u>Ready Queue</u> mei eik queue mei lage hai.

Jb CPU milega tb execute start hoga.

warna ni.

<u>Life Cycle of Thread:</u>



Ready Queue mei push kele phir CPU pass jake jo state hogi running.

sleep ↑

Jitne thread ko time mile, usne kam khtm kiya, execution khtm, dead hoga.

→ Suppose running mei tha, kisi high priority thread ne usko nikal kr waiting mei bhej diya, ye kisi aur

thread ne usko sleep state mei bhej diya.
Jb vo is state se bahir niklega, tou vo CPU/running ke pass nhi jayega... vo ready Queue mei jayega & again it will wait for CPU.

new Thread(); → born state.

Start(); → ready Queue mei jayega

CPU mile → Running Condition

Given quantum, → Dead
time splice mei kar lete

Agar interrupt → sleep/wait

پھر جب ان سے باہر آئے گا
تو ready Queue کے پاس جائے گا
اور CPU کا ویٹ کرے گا
CPU ملے گا تو execution start