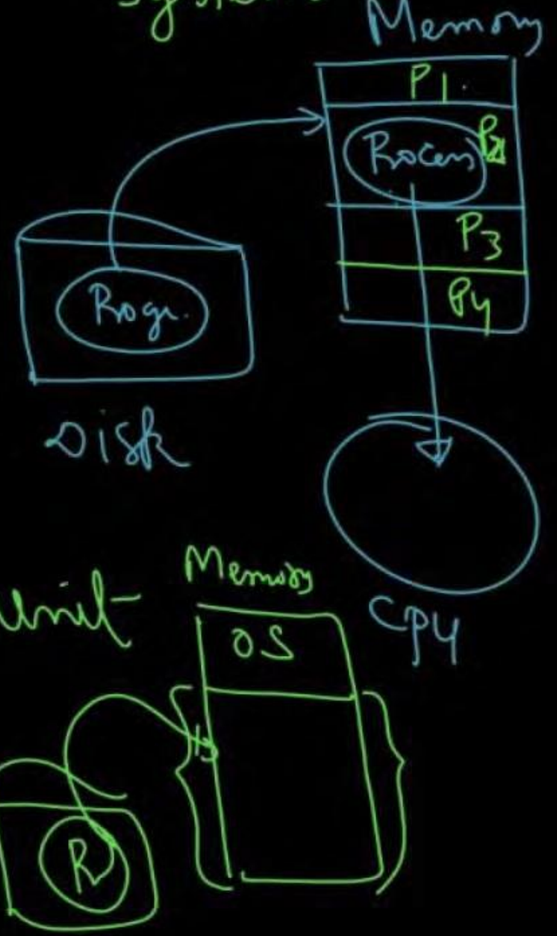
Process Concepts:

Program vs Process

Program in execution → utilizing Resources of a Computer System (CPU + Mem + IO)

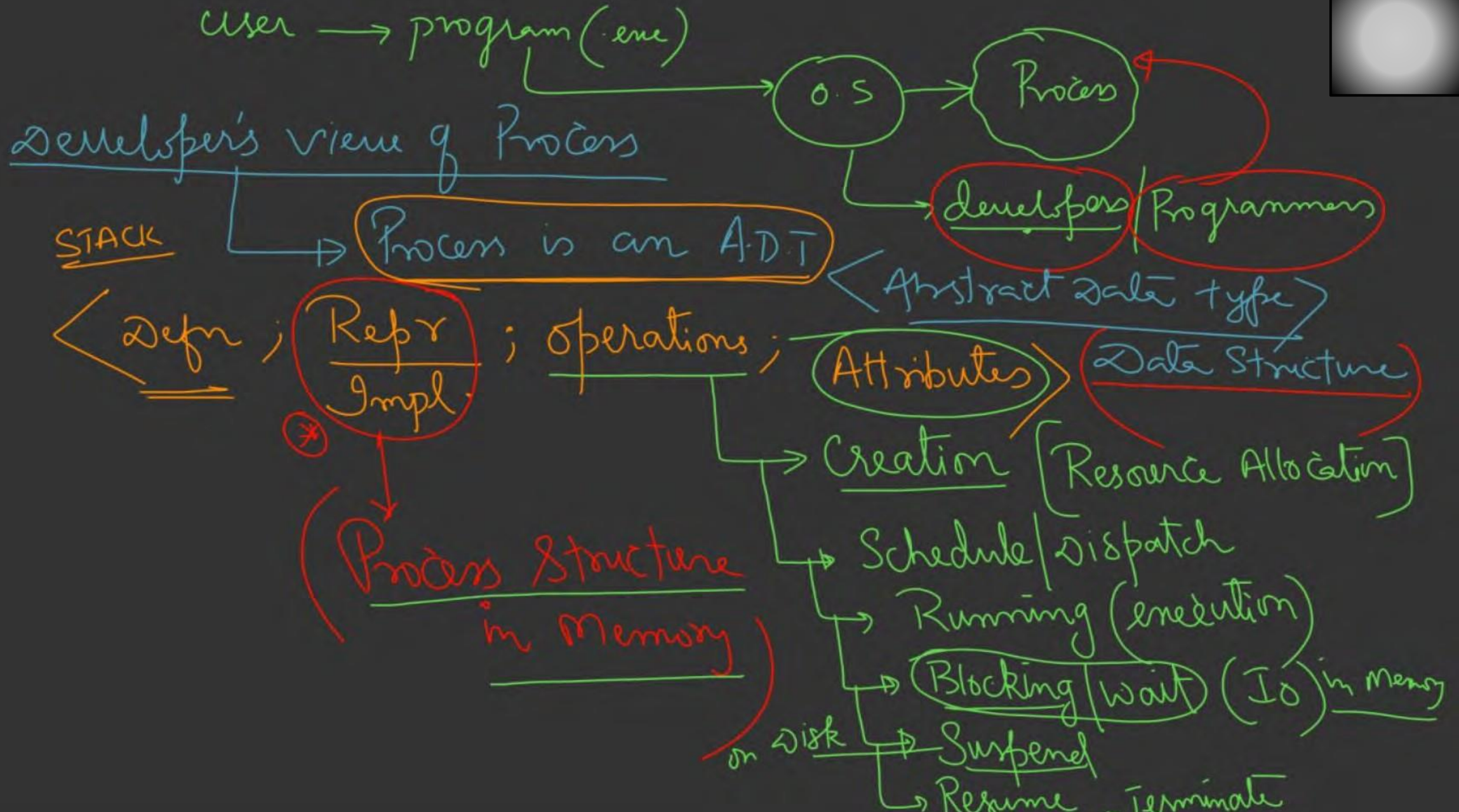
Defns of Process:

- Instance of a program;
- unit of cpu utilization
- Active entity
- Schedulable / Dispatchable unit
- Locus of control of OS
- Animated Spirit



created from program

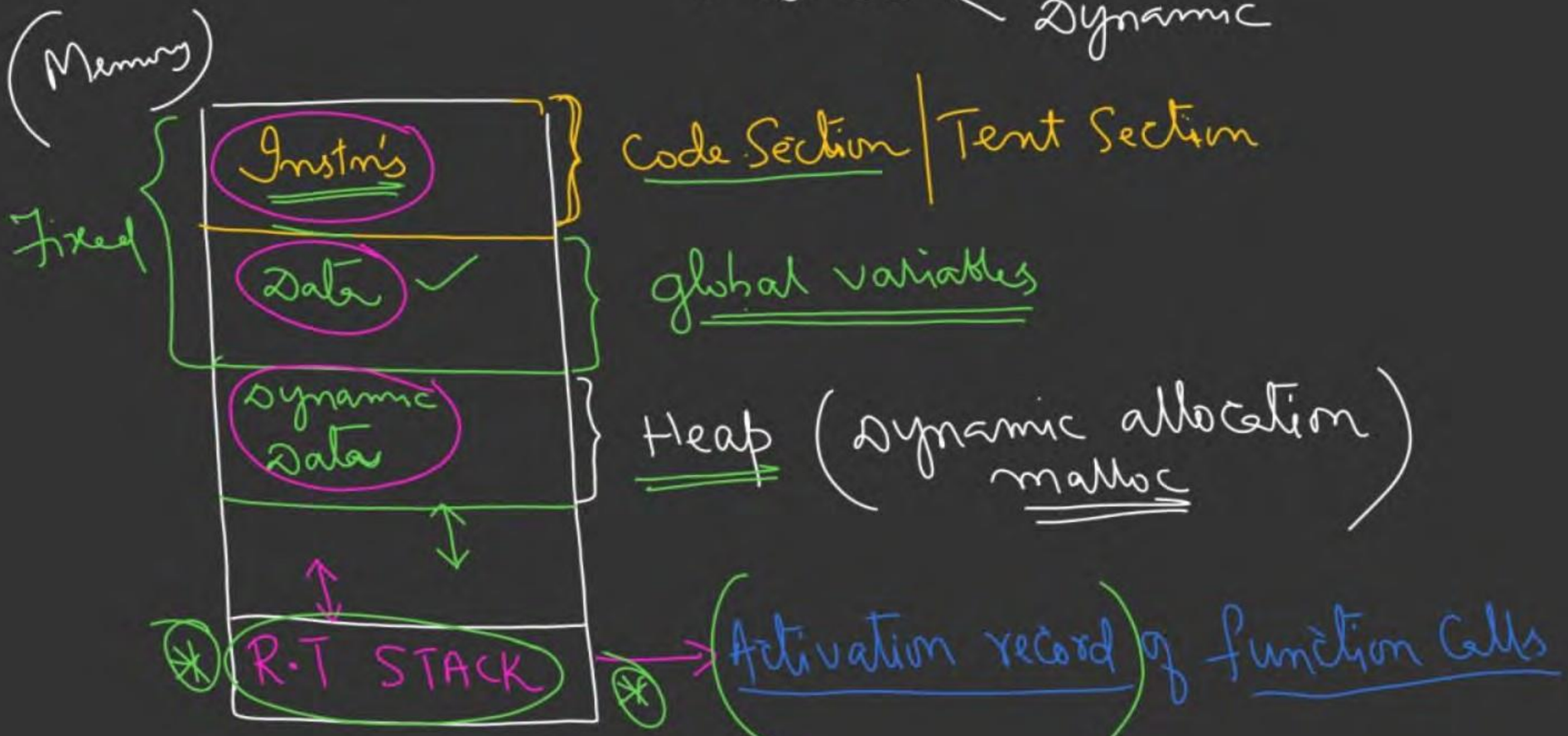
Program (.exe)	Process
1) on disk	1) In Memory
2) without resources	2) utilizing resources
3) Passive	3) Active (Alive)


User → program (.exe)
O.S
Process
developers / Programmers
Developer's view of Process
STACK
Process is an A.D.T
Abstract data type
Defn ; Repr / Impl ; operations ;
Attributes
Data Structure
Creation [Resource Allocation]
Schedule / dispatch
Running (execution)
Blocking / wait (IO) in Memory
Suspend
Resume → Terminate
Process Structure in Memory
on disk

Abstract view of Process (in Memory) / Process Structure

Program < Instn's
Program < Data < Static
Program < Data < Dynamic

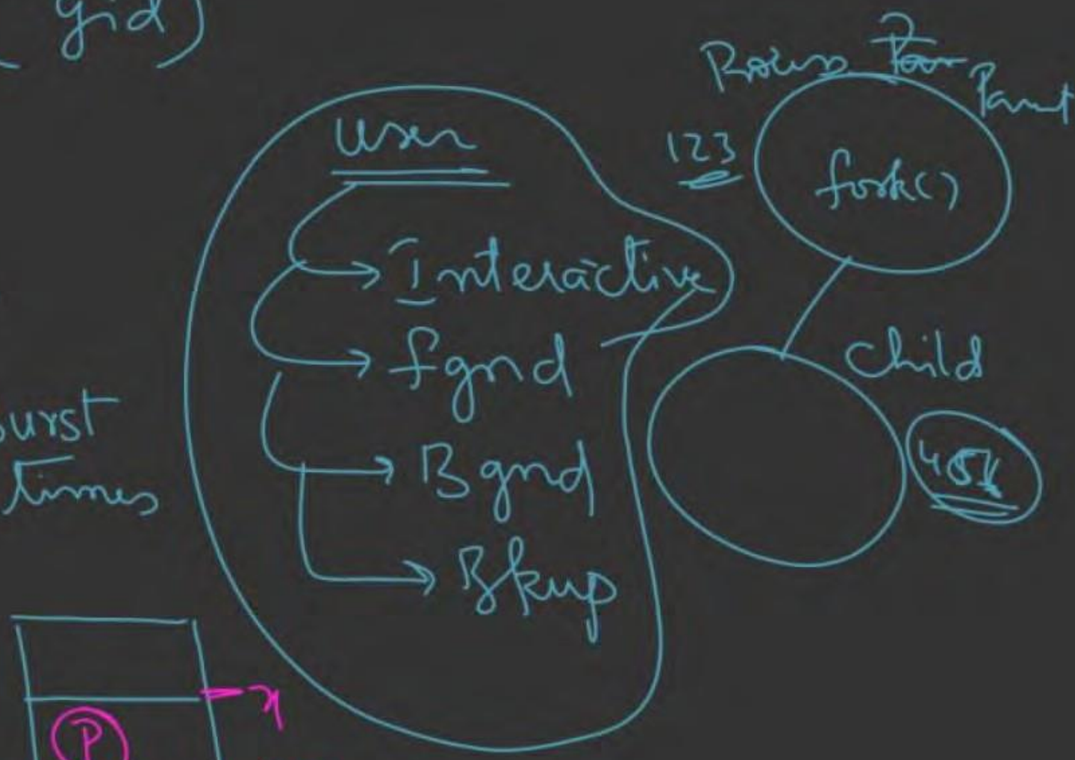
R.T : Run-time



Process Attributes:

PC: Address of next Instn in Code Section

1) Identification: Process_id (Pid); Parent Pid (PPid)
   group-id (gid)
2) cpu-related: Type; PC;
   Register_set;
   Priority; Burst times
   (State)
3) Memory related: Size, Mem limits
4) File related: list of open files



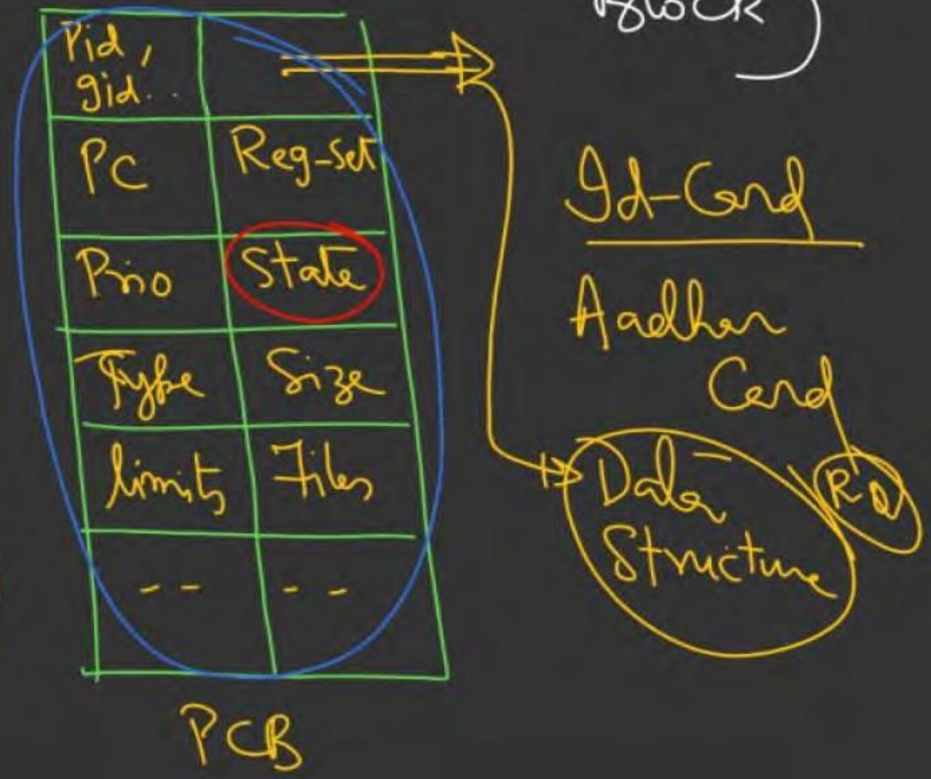
→ Attributes of a Process are Stored in a table (O.S) known as P.C.B (1M)

(Process Control Block)

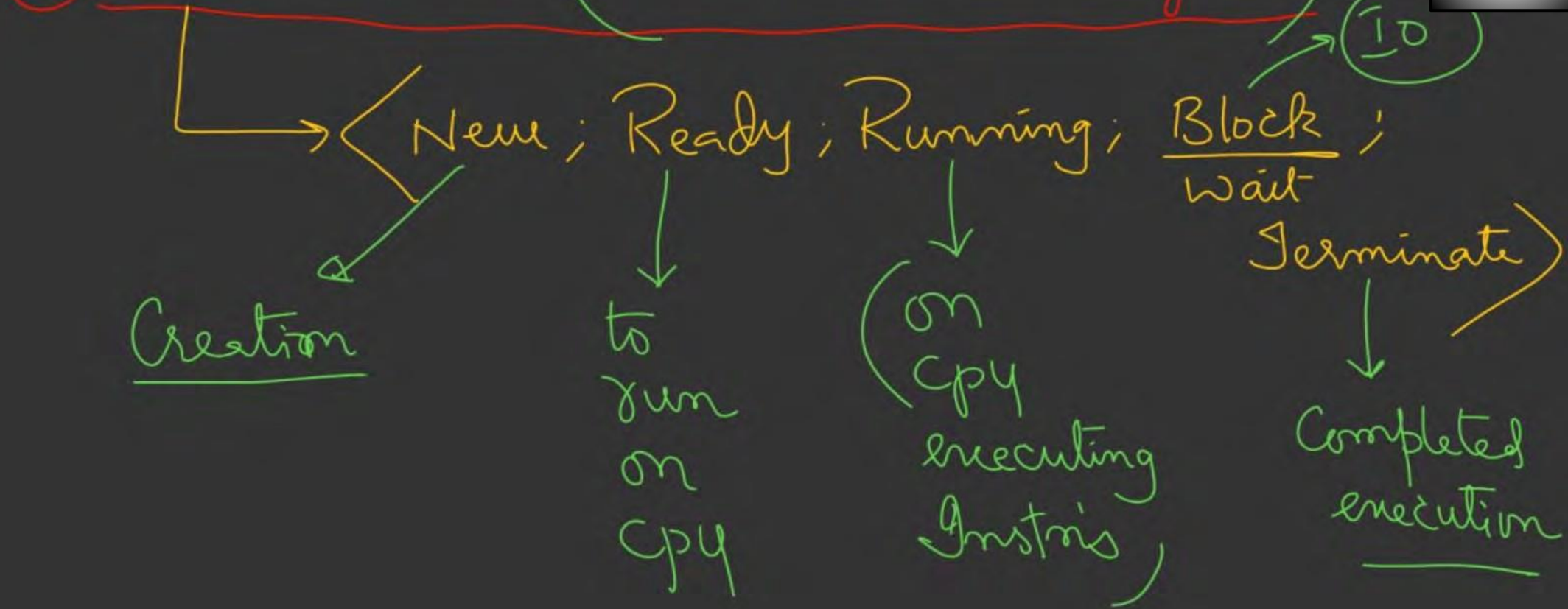
→ Each Process has its own PCB;

→ PCB is created @ the time of Process Creation & destroyed @ Process termination

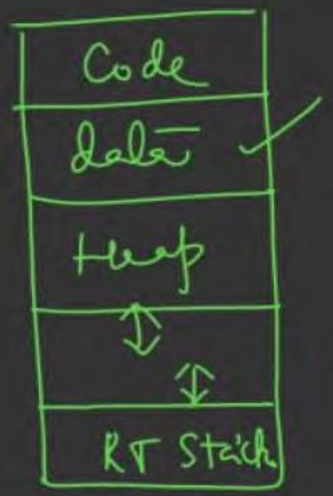
→ Volume of Information kept in PCB is known as Context / Environment of Process (Process Context)



④ Process States & State Transition Diagram (S/IO)



Activation Record contains : (Mem for formal Parameters, Local variables, + Return Address)



Global Data

2B

```
int x = 0;
```

```
main()
{
    int a, b, *c;
    a = 1;
    b = 2;
    c = (int *) malloc(sizeof(int));
    f(a, b);
}
```

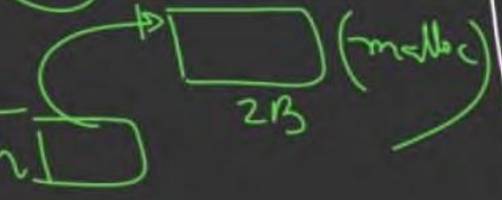
Local variable (2B 2B 2B)

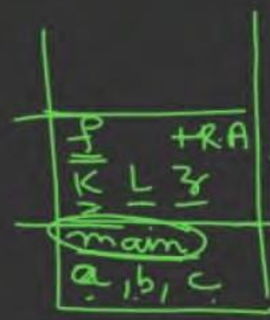
Local Code

Heap (2B)

Code

Actual Par.





R.T Stack

```
f(int k, int l)
{
    int z;
    z = k + l;
    printf("%d", k);
}
```

(2B 2B) formal Parameters

2B