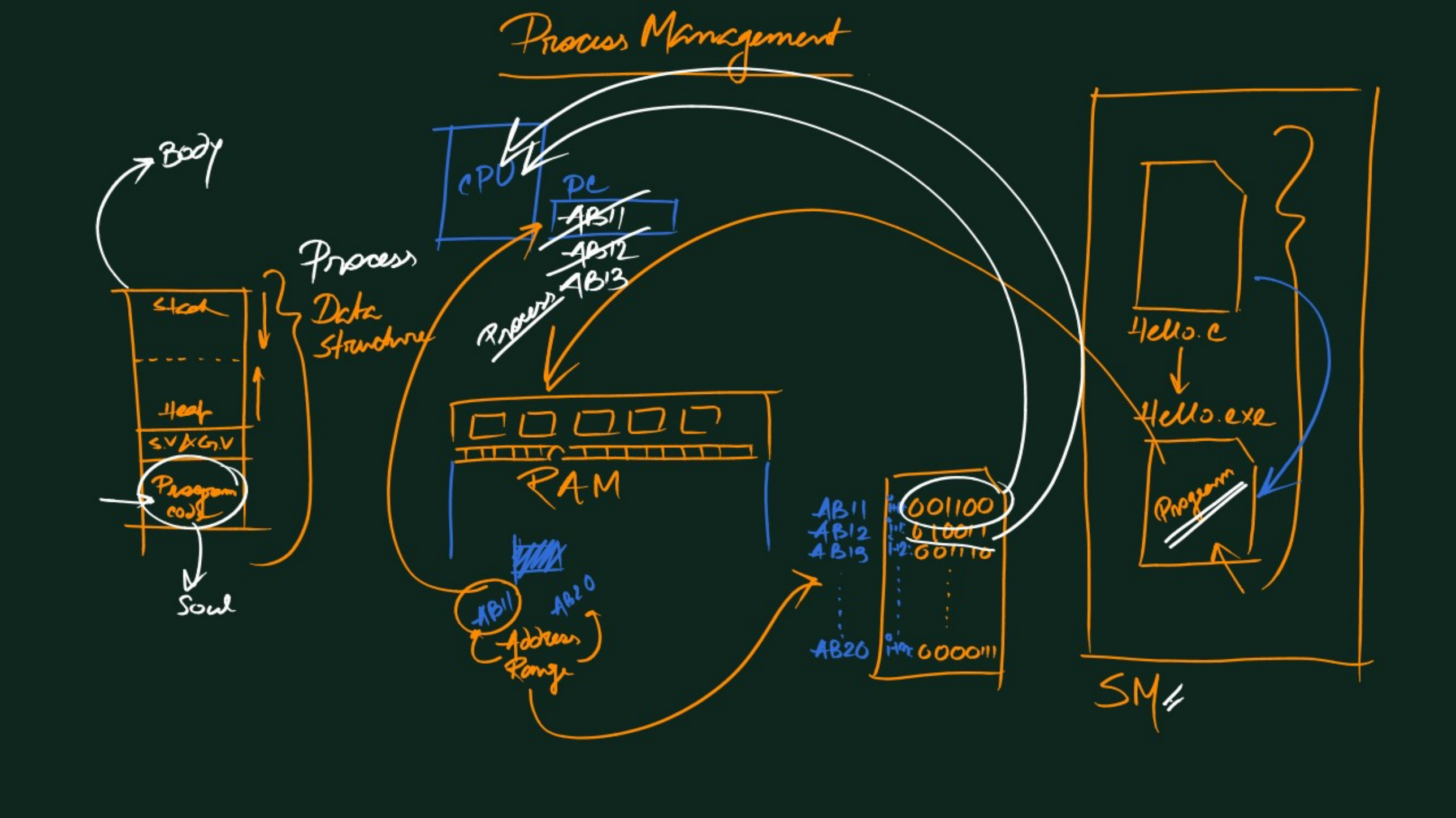
Process Management



Attributes of Process:

i) Process ID
ii) Program Counter
iii) Process state
iv) Priority
v) General Purpose Registers
vi) List of open files
vii) List of open devices
viii) Protection

```
fp = fopen(..);
.
.
.
fclose(fp);
```

2+3=?

Acc 00000010 00000011

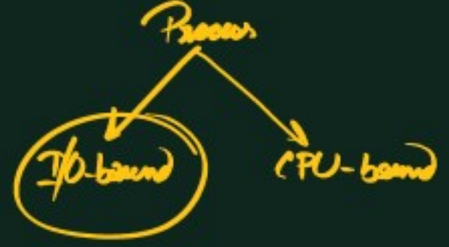
Intel 8085

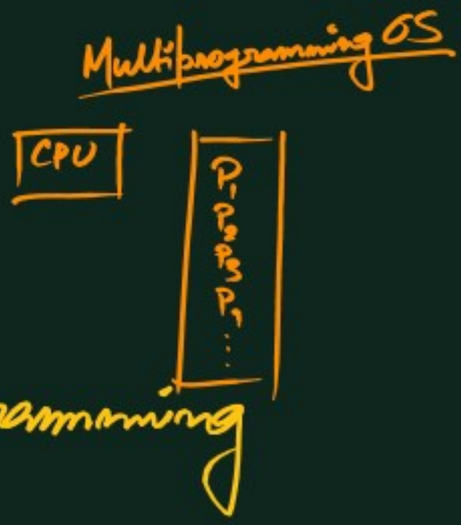
```
reg int x=2;
reg int y=3;
.
.
.
z=x+y;
```

highest priority

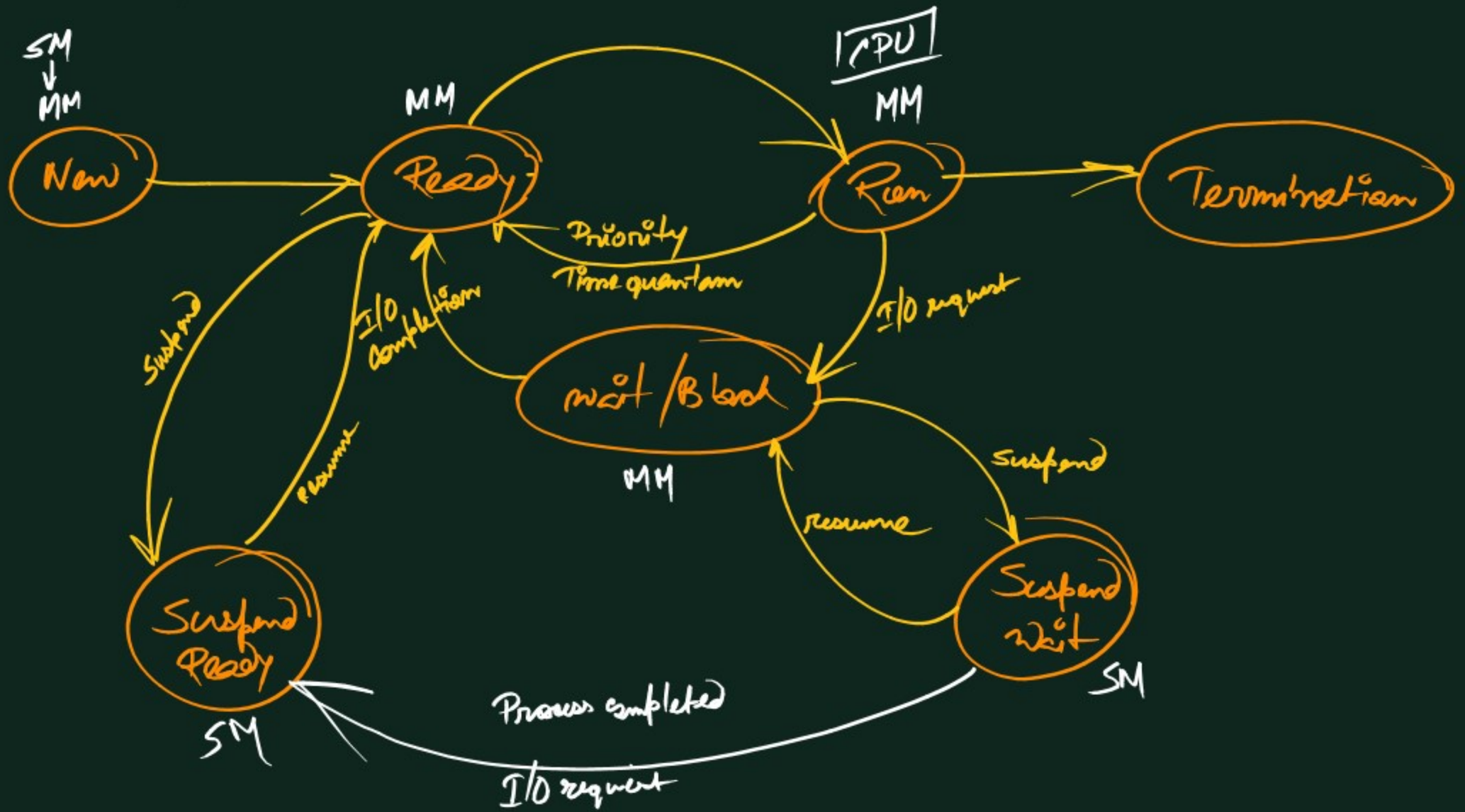
Process States :

i) New (SM)
ii) Ready (MM)
iii) Run / Running (MM)
→ iv) Wait / Block (MM)
v) Suspend Wait (SM)
vi) Suspend Ready (SM)
vii) Termination





Process state Diagram



Process Management

- **What is Process?** (Process is body & Program is soul!)

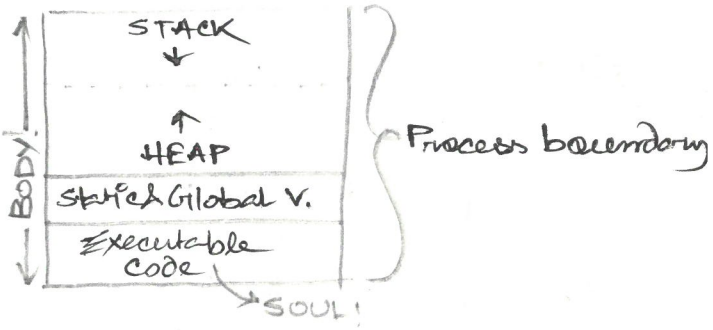
Whenever we write a HLL code & execute compilation on it (using specific compiler), an executable code is produced. Both the HLL code & executable code (which resides in SM) are called Program.

Now, the OS brings the executable code (Program) to MM and places it in a particular Data structure called Process so that the Processor can execute it.

- Process is created by the OS in order to execute a program.

Structure of a process (in MM):

- During execution of a process, the OS is not supposed to move further the process boundary, otherwise, SEGMENTATION FAULT may occur.

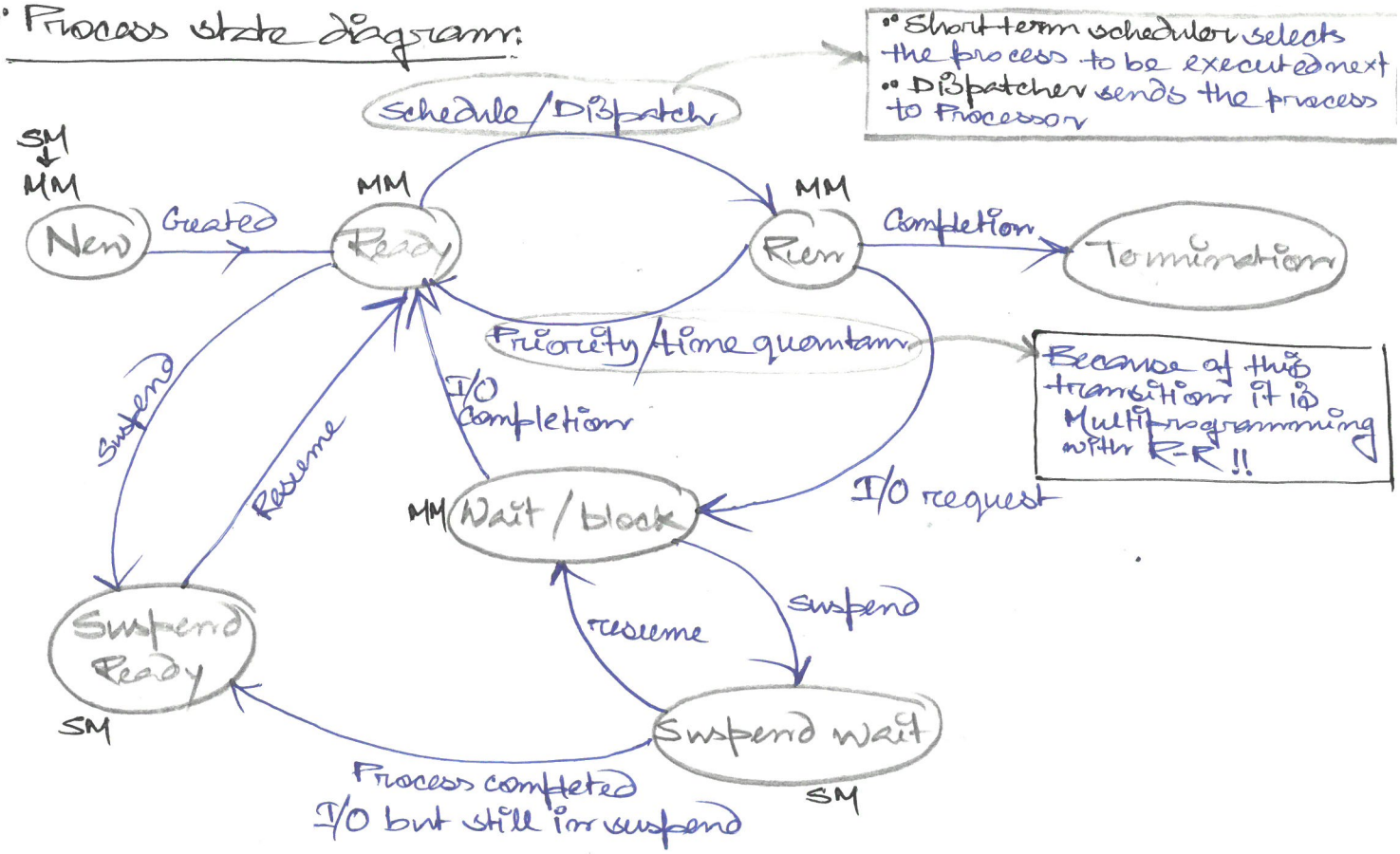


- **Attributes of Process:**

1) Process id: For every process created in MM, OS assigns a unique number to it. (like SSN)
2) Program counter: Keeps track of the next instruction to be executed in a process.
3) Process state
4) Priority
5) General Purpose Registers: During preemption the status of the GPRs (values in the GPRs) should be saved somewhere so that the previous process is executed properly when it's execution is resumed.
6) List of open files
7) List of open devices
8) Protection: OS protects the processes to be accessed from outside (from other processes)

All these attributes are stored in Process Control Block. The PCBs of different Processes are stored in a Linked List. → A.K.A. Context

Process state diagram:



- Minimum no. of states required for execution of a process is 4 (i.e. New, Ready, Run & Termination)

3 decisions are made here,

1) How many processes are to be created? (Based on Resources)
   → Taken by Long time scheduler.
2) Which process to send to the CPU (run state)?
   → Taken by Short time scheduler/Dispatcher.
3) Which process to suspend?
   → Taken by Medium time/term scheduler.

- Degree of multiprogramming: # processes can reside in Ready state.

- Short term scheduler is responsible for context switching.
- Mid-term scheduler is responsible for Swapping.