Operating Systems (Lecture-2)

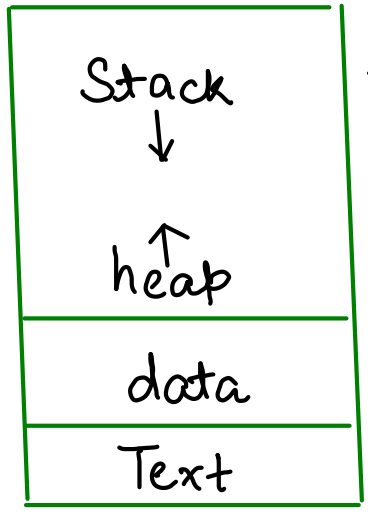
Process management

1) What is a process?

Program	Process
→ Set of instructions	→ Program under execution
→ static	→ dynamic

1) how much time? → OS
2) In which order? → OS

Sections of a process



→ local variable

→ Dynamic allocation

⇒ →Static and global variables

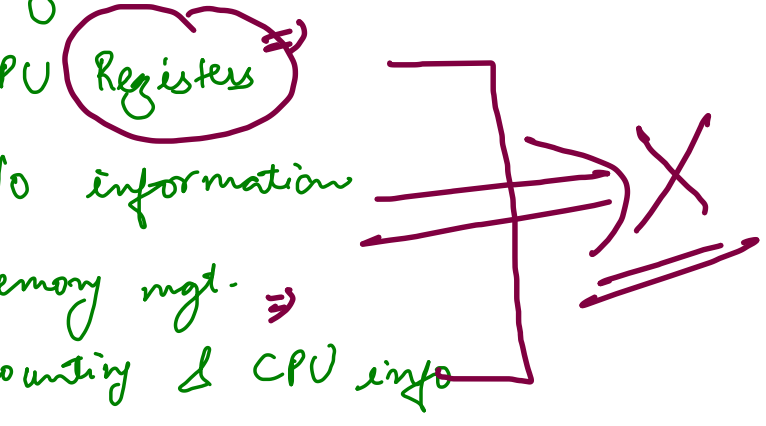
⇒ → compiled code

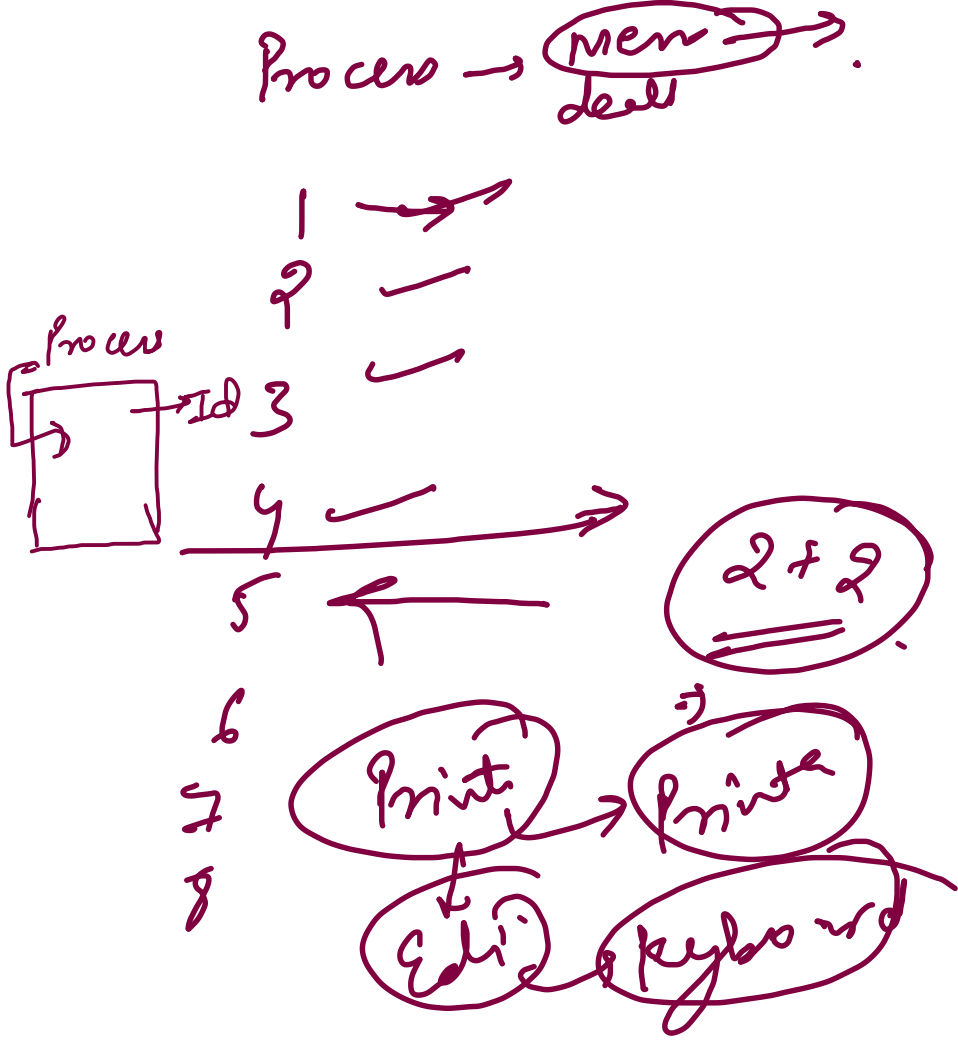
1. Load the program into memory
2. Allocate run-time stack ⇒
3. Allocate heap ⇒
4. I/O related tasks ⇒
5. OS hands off control to main ⇒

1) Mem. allocat
2) Execu
3) Deallocatio

PCB - process control block

1) Process ID - unique identifier
2) State - process state
3) Priority -
4) Program counter - next instruction
5) CPU Registers
6) I/O information
7) Memory mgt.
8) Accounting & CPU info

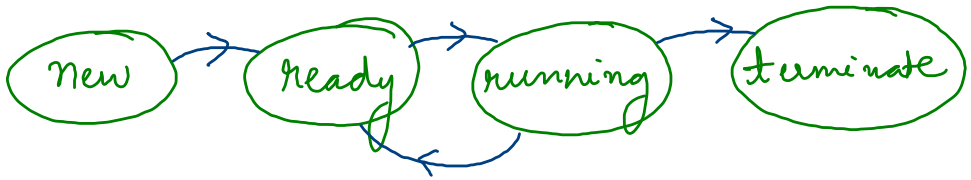




States of a process

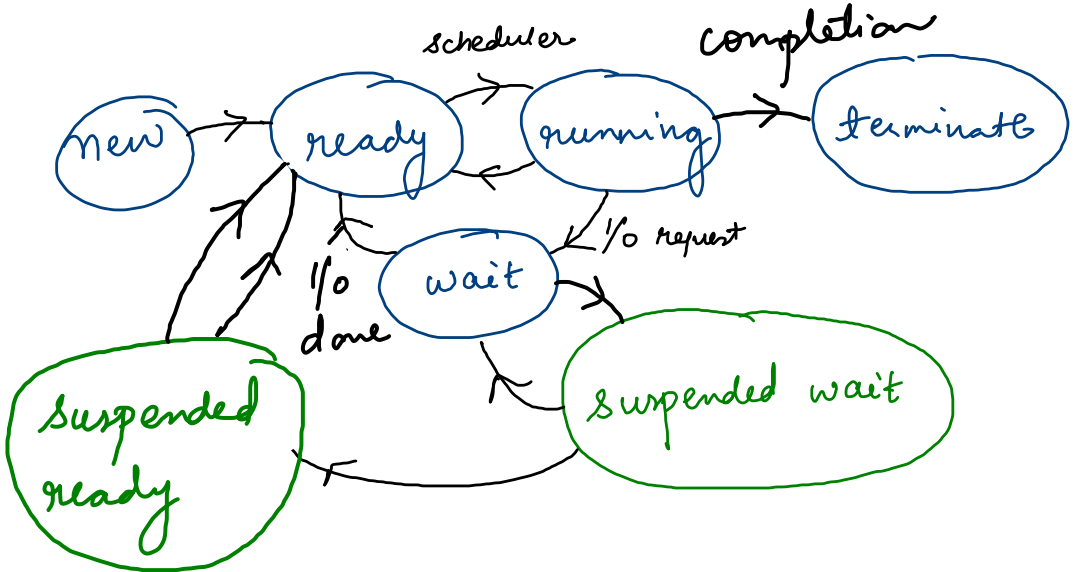
Primary States

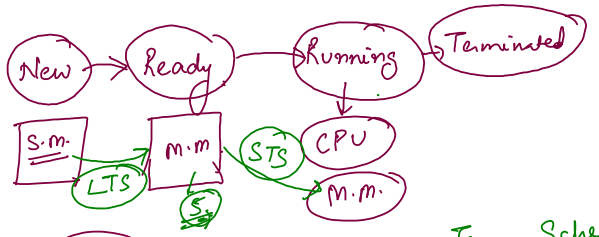
1) New
2) Ready
3) Running
4) Terminate



Additional states

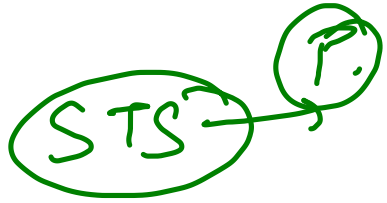
1) Wait
2) Suspended wait
3) Suspended ready





LTS - Long Term Scheduler

STS - Short Term Scheduler



⇒ degree of multiprogramming

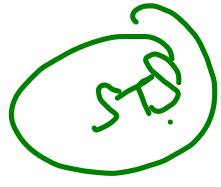
Job queue → New State

↓

Ready queue → m.m.

↓

Running



⇒ STS

LTS

M.M → M.M.

Ready → Running

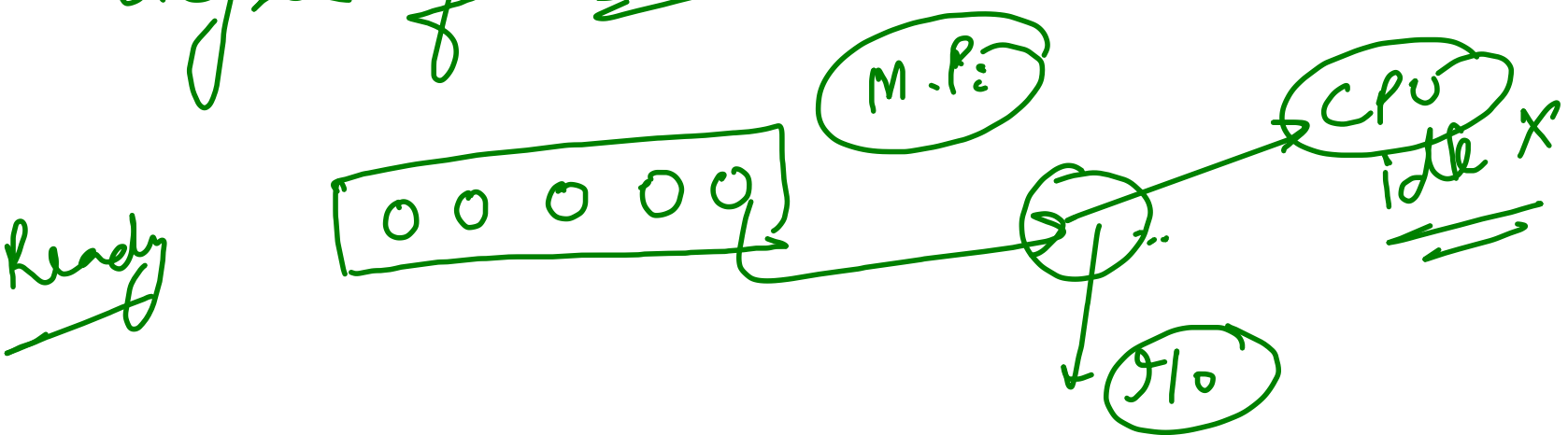
New → Ready

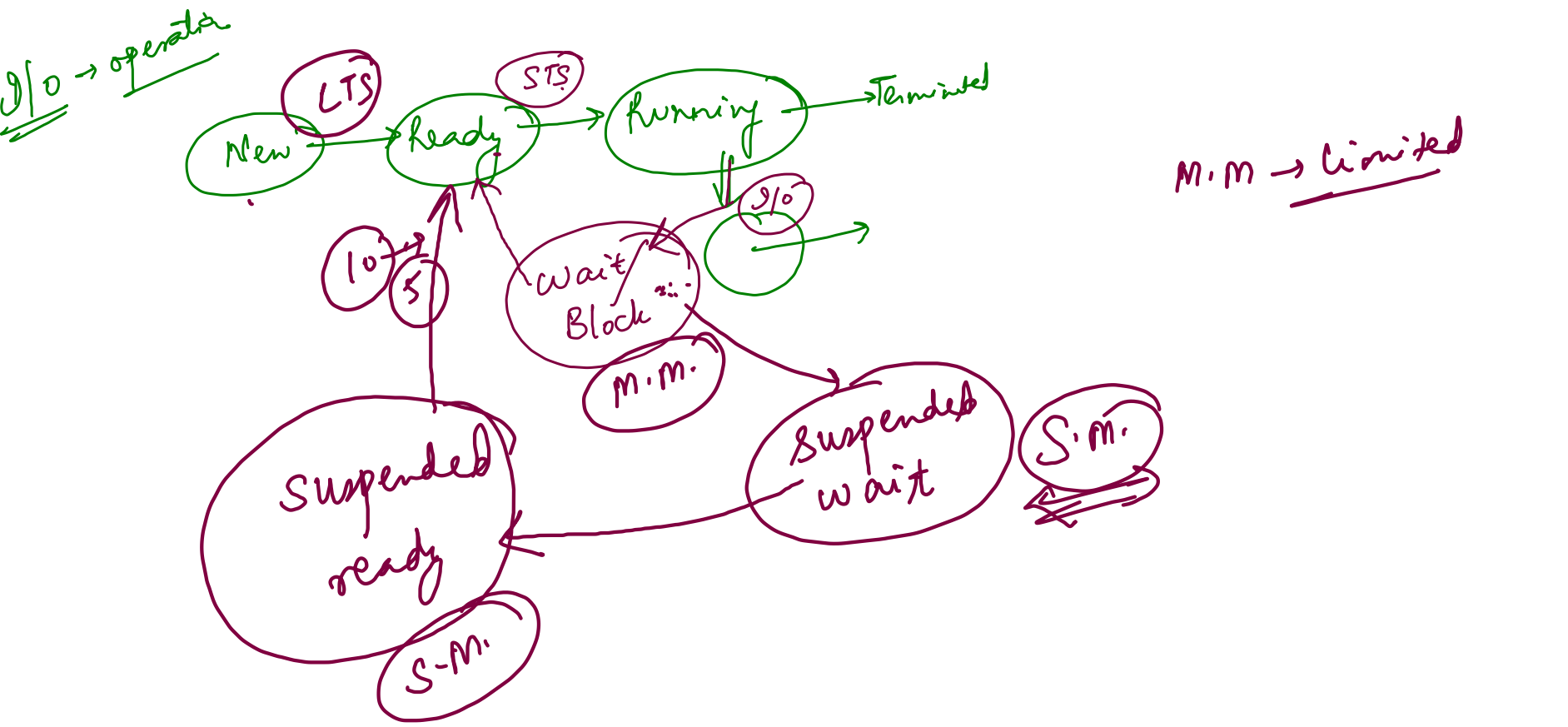
S.M.

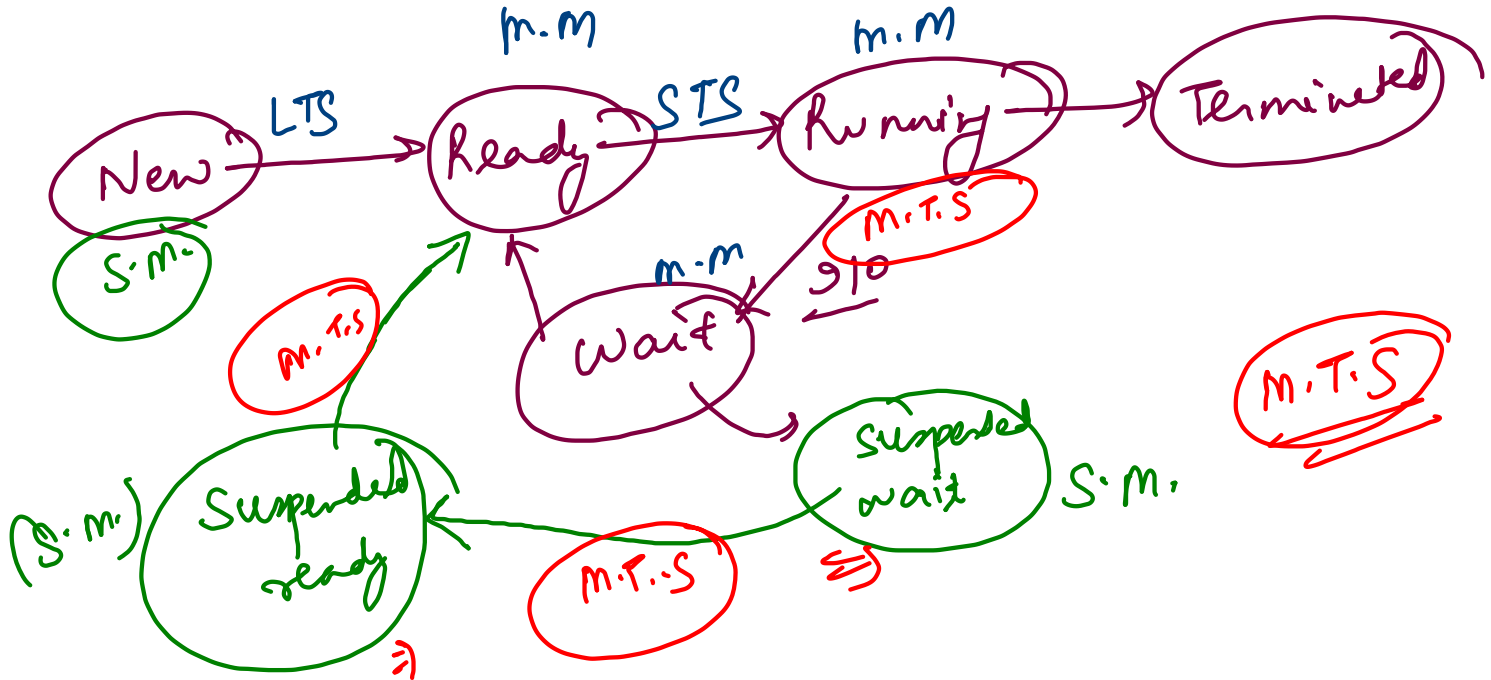
M.M.

1) Jeske kaunsa process

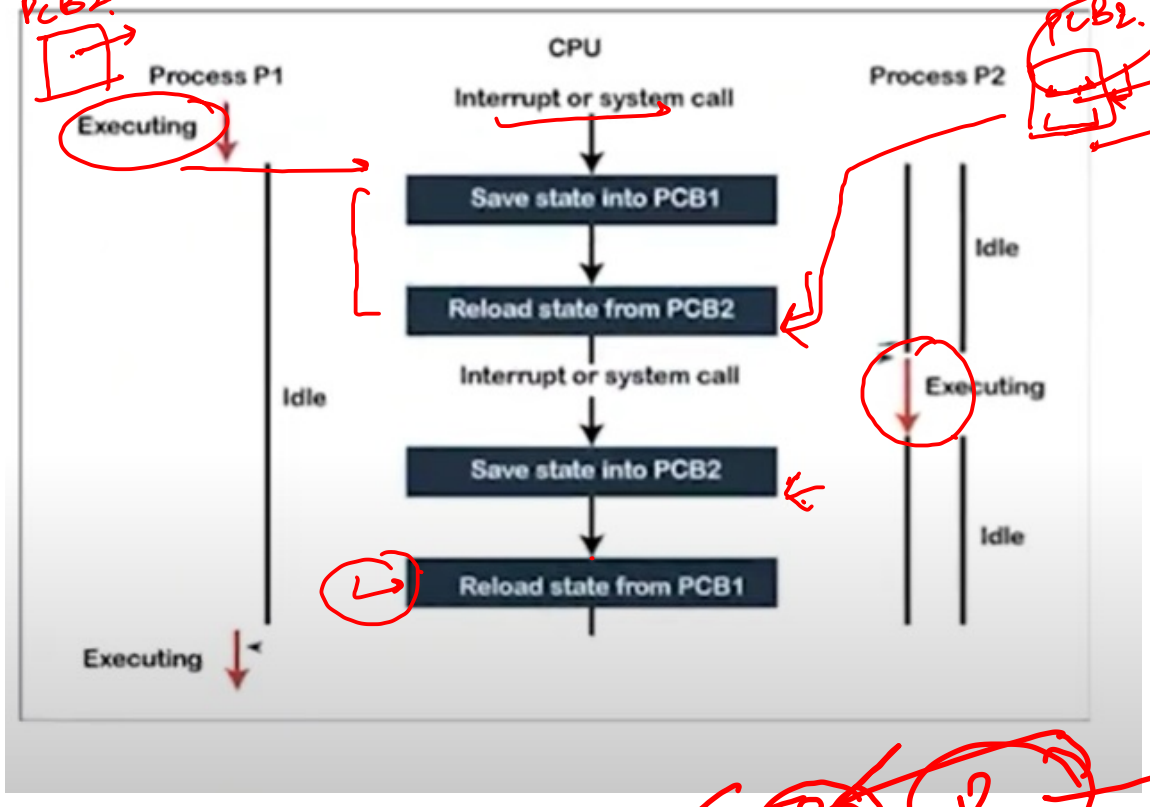
degree of M.P. ⇒




I/O → operation
New
LTS
Ready
STS
Running
Terminated
I/O
I/O
Wait
Block
M.M.
Suspended ready
S.M.
Suspended wait
S.M.
M.M → Limited


M.M
M.M
New
LTS
Ready
STS
Running
Terminated
S.M.
M.T.S
I/O
M.M
M.T.S
Wait
M.T.S
(S.M.)
Suspended ready
Suspended wait
S.M.
M.T.S

Context switching



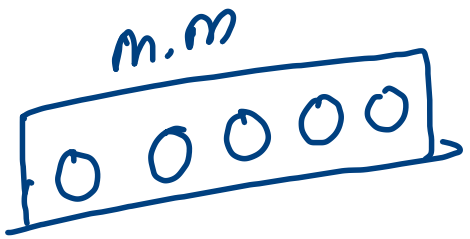
1) High priority
2) Interrupt
3) User to kernel (switch) / I/O operation

1) Job queue

i) process in new state

(ii) Process in secondary memory

(iii) LTS



2) Ready queue

(i) process in ready state ⇒

(ii) process in main memory ⇒

(iii) STS

Running

STS

Schedulers

Long-Term Scheduler	Short-Term Scheduler	Medium-Term Scheduler
It is a Job Scheduler	It is a CPU Scheduler	It is a process swapping scheduler
It takes process from the job pool	It takes process from the ready state	It takes process from running or wait/dead state
Its speed is lesser than short-term scheduler	It is fastest among the two other schedulers	Its speed is in between long-term and short-term
It controls the degree of multiprogramming	It has less control over the degree of multiprogramming	It reduces the degree of multiprogramming

1) CPU Burst - CPU time ↑

2) I/O Burst - I/O time ↑

1) Need of scheduling

CPU utilization (Max)

Processes → CPU utilization (max)

2) Scheduling metrics

(i) Arrival time (AT) → → (11:00 AM)

(ii) Completion time (CT) → → (12:00 PM)

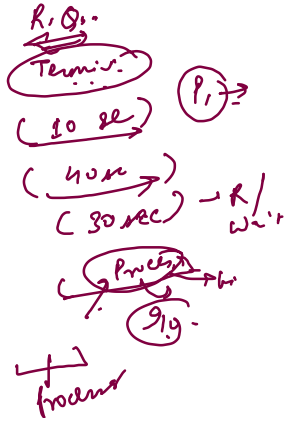
(iii) Burst time (BT) time req. to get execution → (15 min)

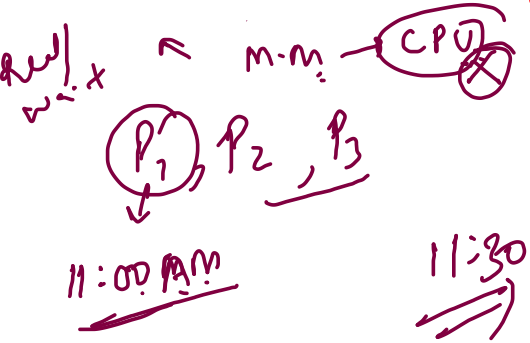
(iv) Turn around time (TAT) = (CT − AT) (1 hr)

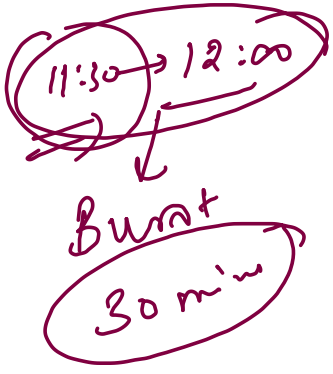
(v) Waiting time (WT) = (TAT − BT) (45 min)

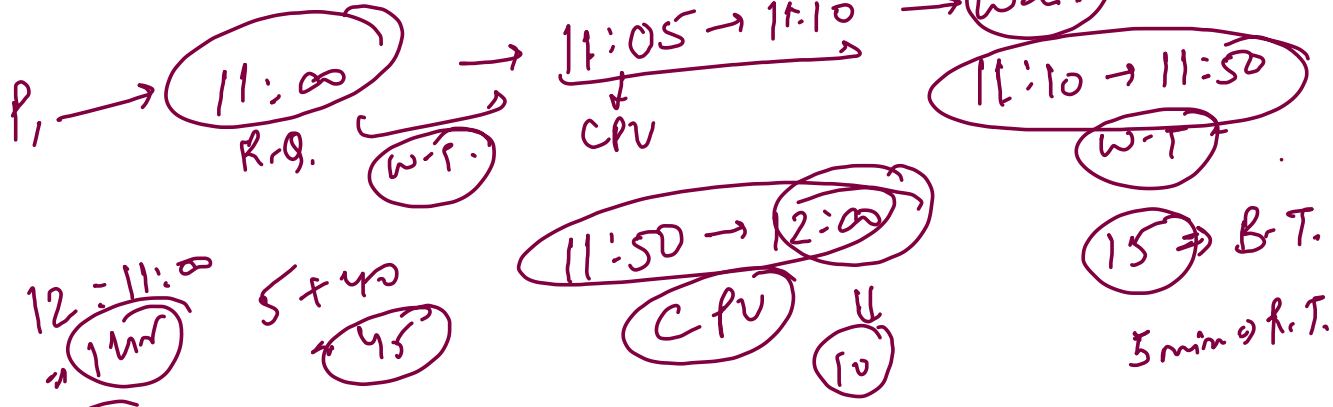
(vi) Response time (RT) = (process gets CPU for 1 time − AT) ( )

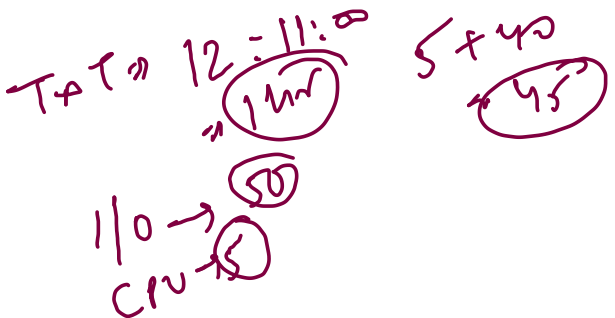
(service)

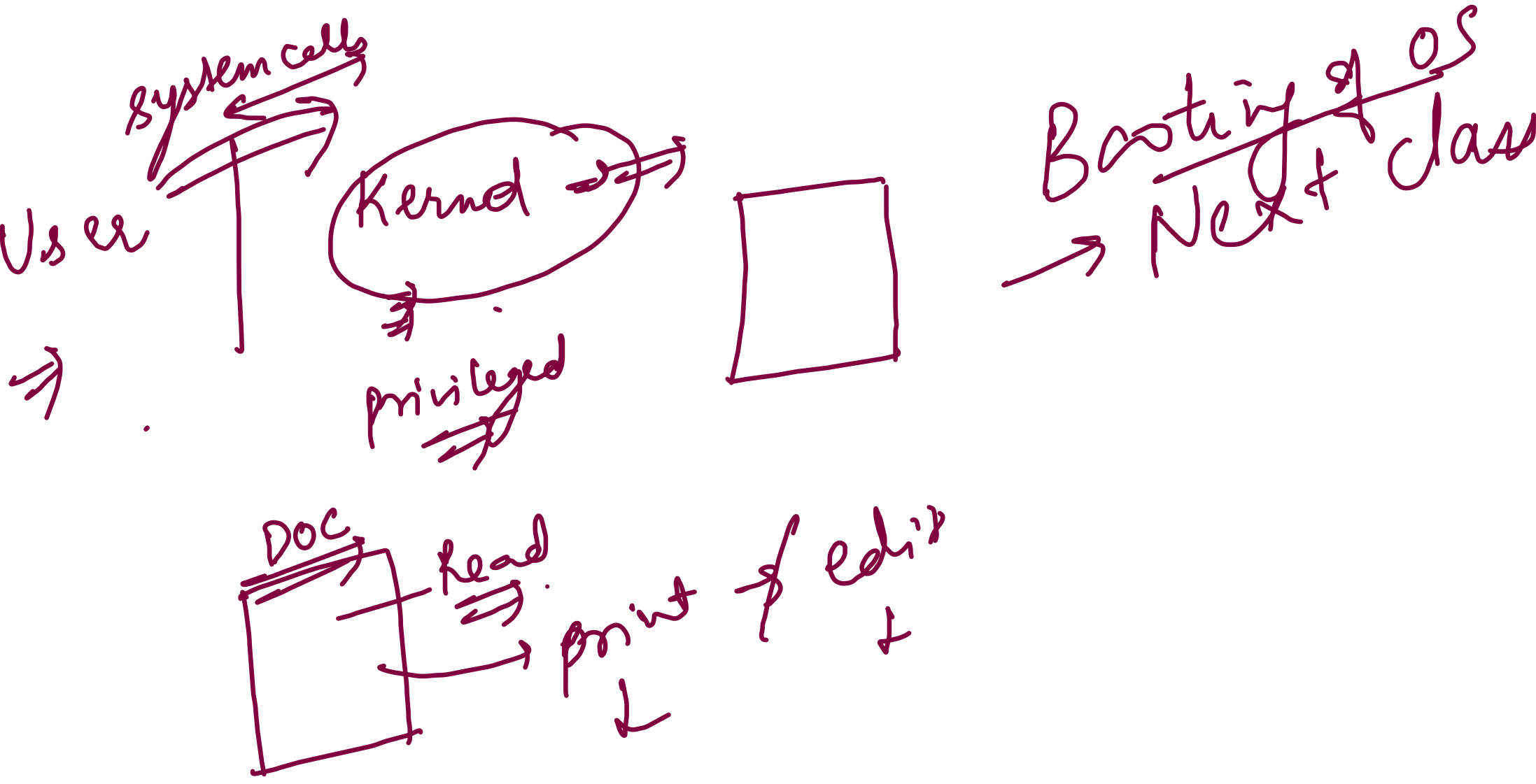











system calls
User
Kernel
privileged
Booting of OS
Next class
DOC
Read
print
edit

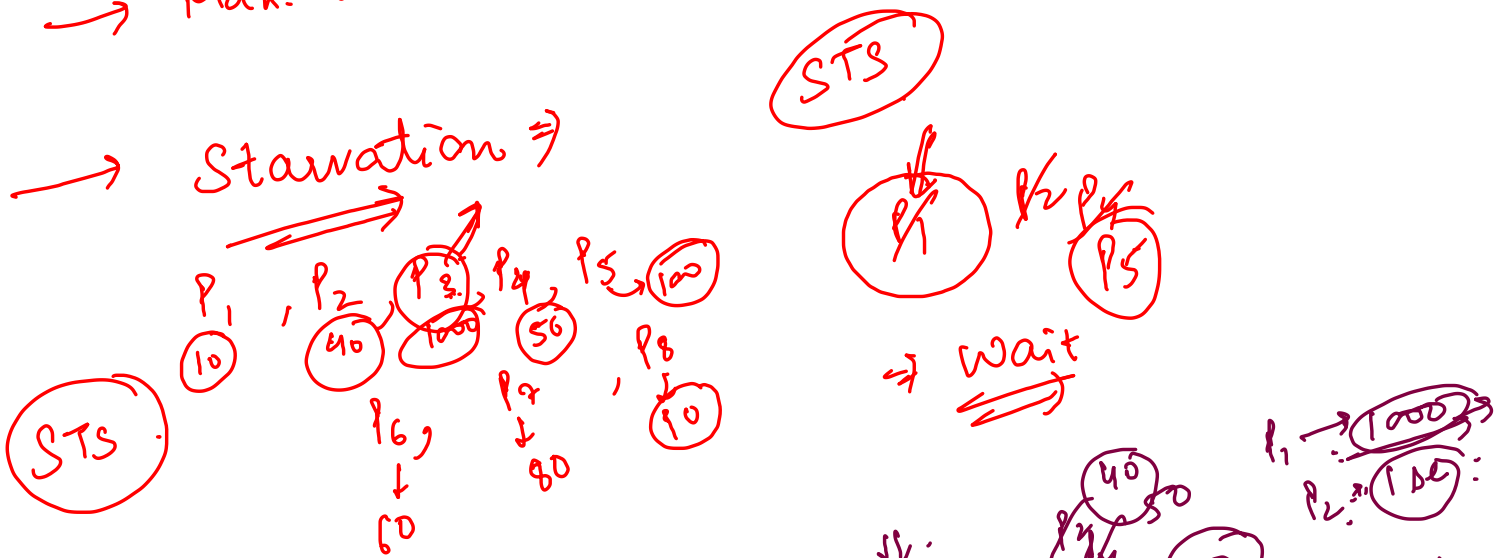
Scheduling

Ready → Running

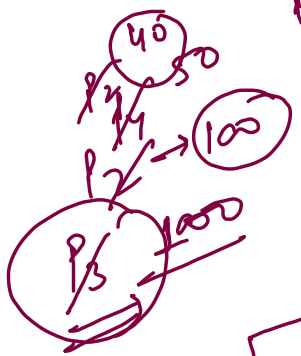
STS

→ Max. utilization ↑

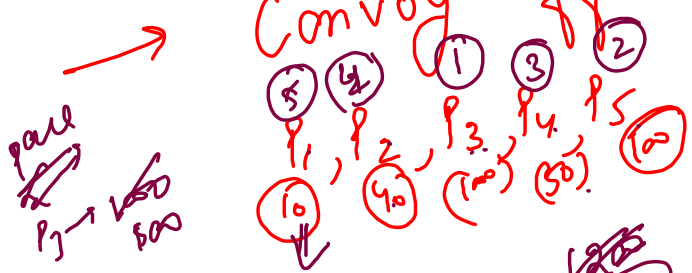
→ Starvation ⇒

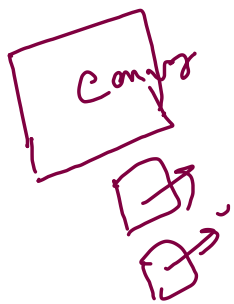


⇒ wait



→ Convoy effect





1190

10

1) Starvation - When high priority processes keeps executing & low priority processes get blocked for indefinite time.

2) Convoy effect - when one slow proces, slows down the performance of entire system.