



OPERATING SYSTEM

UNIT 2

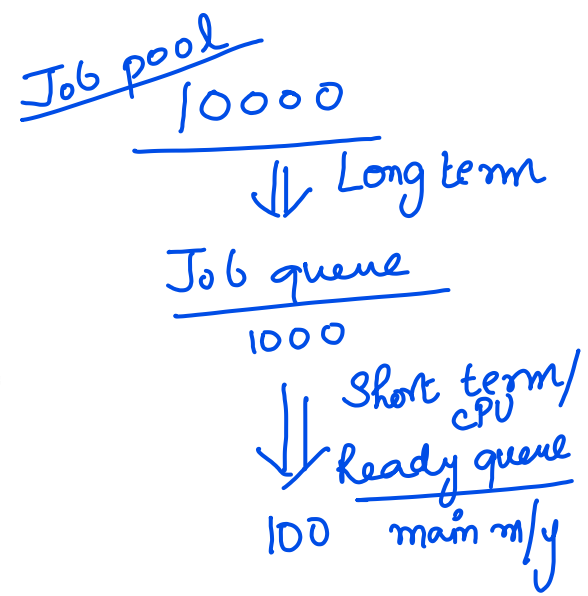
$P_3 P_2 P_1$

$P_1 P_2 P_3$

CPU
1

PROCESS SCHEDULING

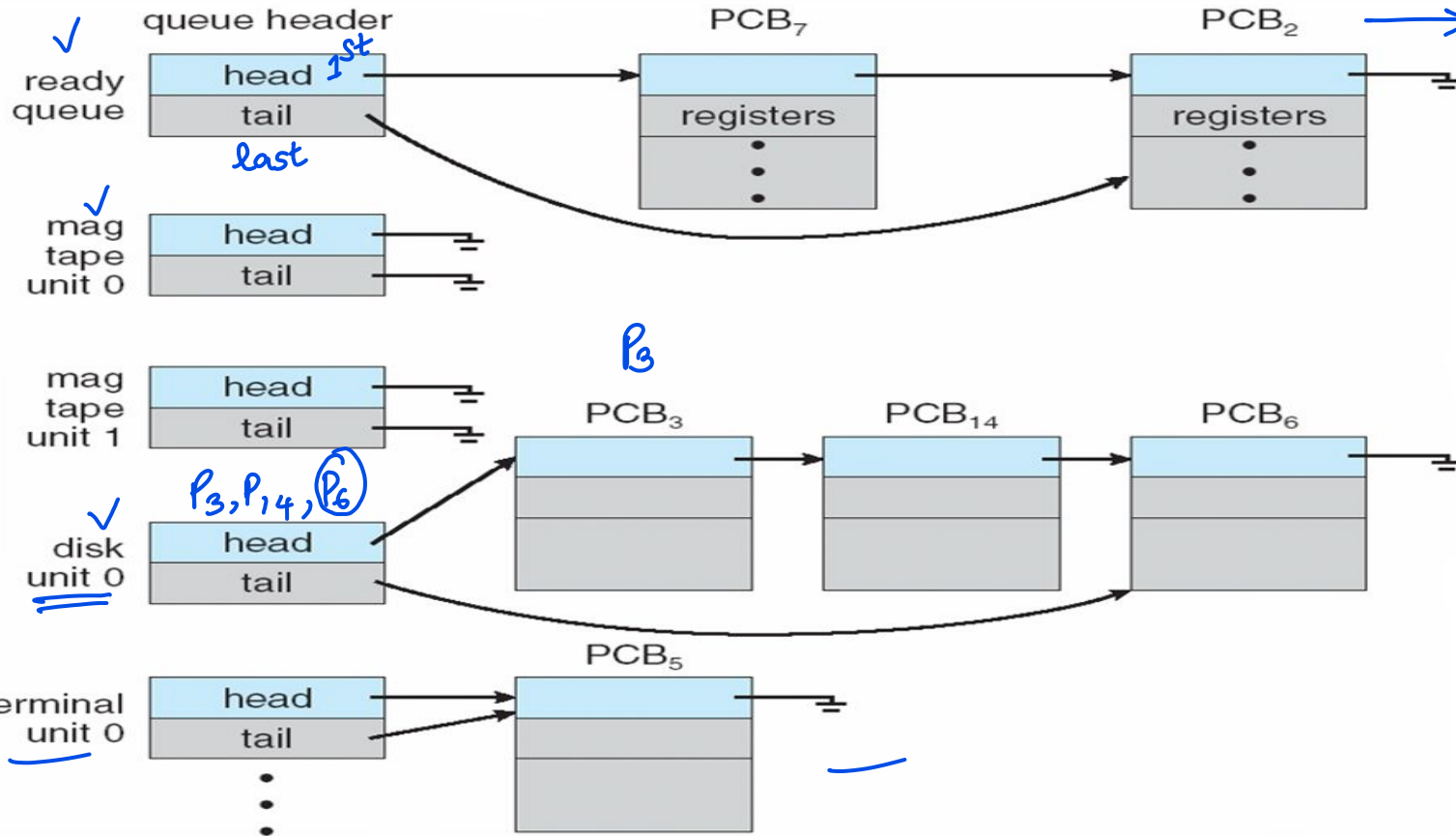
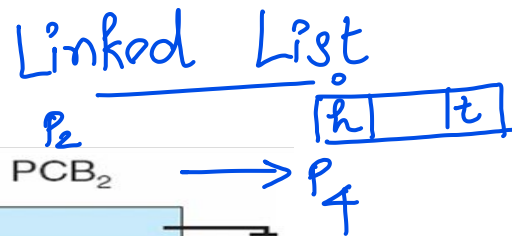
- Maximize CPU use, quickly switch processes onto CPU for time sharing
- **Process scheduler** selects among available processes for next execution on CPU
- Maintains **scheduling queues** of processes
 - 1) ● **Job queue** – set of all processes in the system
 - 2) ● **Ready queue** – set of all processes residing in main memory, ready and waiting to execute
 - 3) ● **Device queues** – set of processes waiting for an I/O device
 - Processes migrate among the various queues



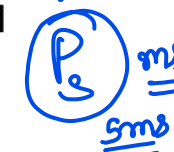
Key → $P_1 P_2 P_4$

READY QUEUE AND I/O QUEUES

$P_7 \rightarrow P_2$



Schedulers ⇒

Ready queue → P_1, P_2, P_4, P_5, P_8


- 1 **Short-term scheduler** (or **CPU scheduler**) – selects which process should be executed next and allocates CPU

- Sometimes the only scheduler in a system
- Short-term scheduler is invoked frequently (milliseconds) ⇒ (must be fast)

Job queue

- 2 **Long-term scheduler** (or **job scheduler**) – selects which processes should be brought into the ready queue

- Long-term scheduler is invoked infrequently (seconds, minutes) ⇒ (may be slow)✓
- The long-term scheduler controls the degree of multiprogramming

- Processes can be described as either:

- **I/O-bound process** – spends more time doing I/O than computations, many short CPU bursts
- **CPU-bound process** – spends more time doing computations; few very long CPU bursts

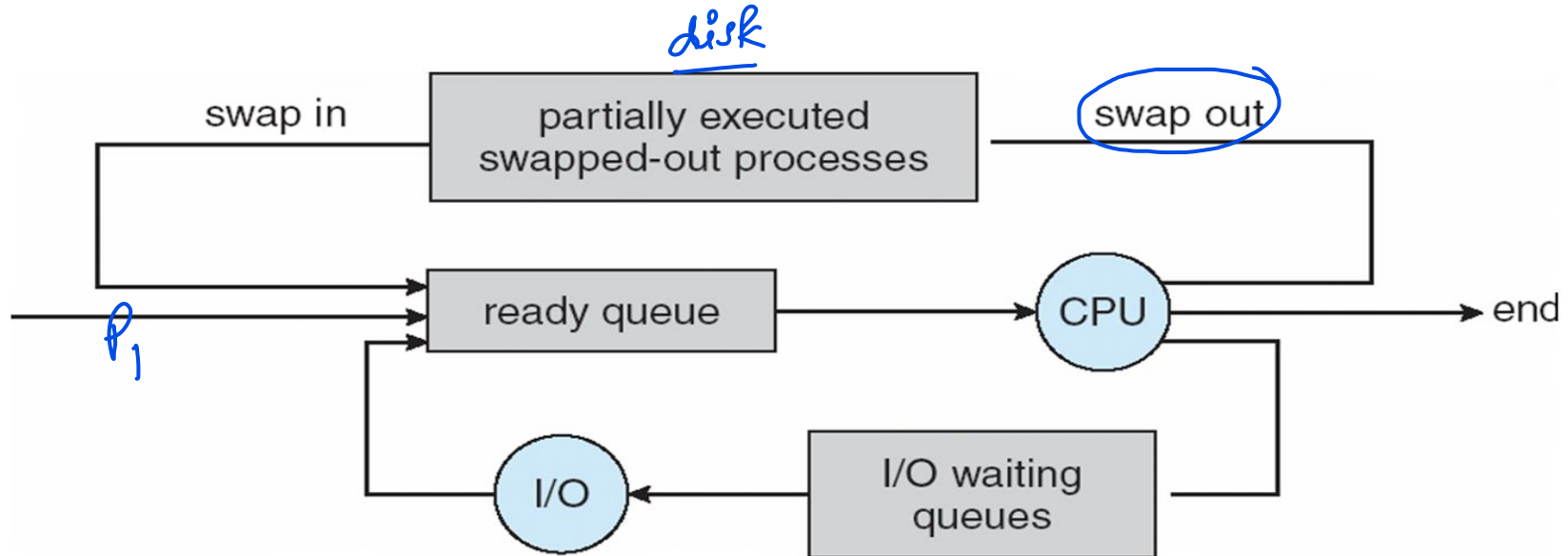
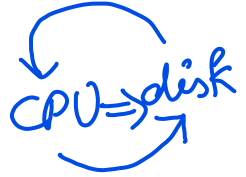
- Long-term scheduler strives for good **process mix**

MEDIUM TERM SCHEDULER ✓

Ready
100

- **Medium-term scheduler** ✓ can be added if degree of multiple programming needs to decrease

- Remove process from memory, store on disk, bring back in from disk to continue execution: swapping



⇒ CONTEXT SWITCH ✓ Process switching

- When CPU switches to another process, the system must **save the state** of the old process and load the **saved state** for the new process via a **context switch**
- **Context** of a process represented in the PCB ✓
- Context-switch time is overhead; the system does no useful work while switching
 - The more complex the OS and the PCB → the longer the context switch ✓
- Time dependent on hardware support
 - Some hardware provides multiple sets of registers per CPU → multiple contexts loaded at once ✓

