Uniprocessor Scheduling

↓

And not process Scheduling: By resource is Scheduled

(Book Ch-9)

→ Resource provided is execution time

→ Aim is to meet System objectives

↳ Response time

↳ Execution - 1st response

↳ Throughput

↳ Rate of job completion

↳ Efficiency

↳ Should not be idle

What are the Scheduling objectives?

→ Share time fairly

→ No proc Starvation

↳ Deprive of proc time

→ Low overhead

↳ Switching is example

→ Prioritize when needed

Types

→ Long term

→ Medium term

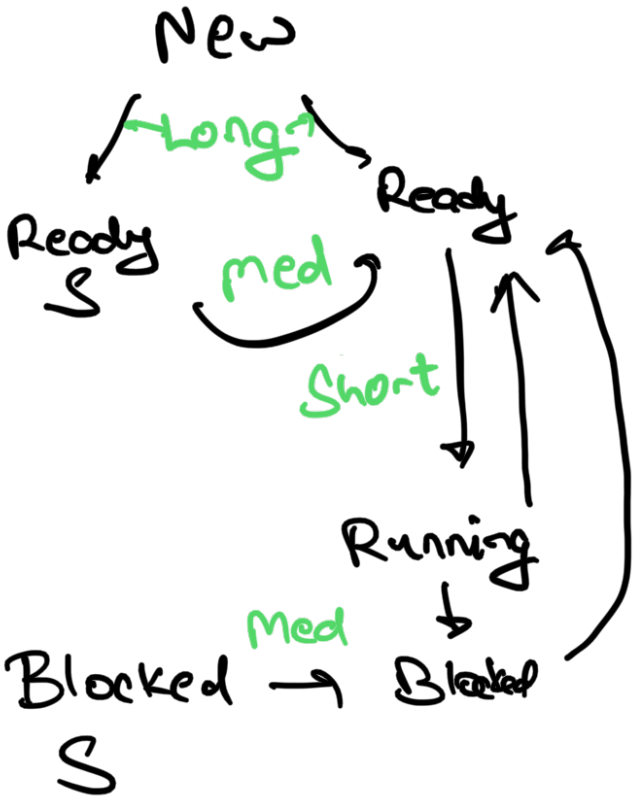
→ Short term

→ I/O scheduling

Long term → New to ready or ready suspend

Med term → Part of the swapping function

Short term → Dispatcher



Long term → FCFS

→ Priority

→ I/O requirement & expected exec time

↳ Not predictable :(

interactive processes

Purpose? → To control the degree of multiprogramming
→ Even for memory

Medium term → Swapping, Manage degree of multi-programming

Short term → Dispatcher, most frequent
→ During process switch

Criteria

User-Oriented → Response time is MVP
→ ASAP response

vs

System-Oriented → Effective + Efficient processor utilisation

Performance Based

vs

Non-Performance → Qualitative
- Satisfaction
- Expectation

independent of context

↓

Can be affected with the same performance level

Performance → Quantitative
- Response time
- Throughput

Interdependent Scheduling Criteria

User Oriented

Performance

→ Turnaround Time:

Completion - Submission
- Execution + Waiting
- Appropriate for Batch; not interactive
  - In interactive, time is user dependent

→ Response Time:

1st Response - Submission
- Web browsing is an example

Non-Performance

→ **Deadlines:**
↳ Maximise % of process met
↳ Real time OS & embedded sys

→ **Predictability:**
↳ Same time at same cost for same job

System Oriented

→ **Throughput:**
↳ Maximise proc per time

→ **Proc Utilization:**
↳ % time of Processor = Busy
↳ Imp in shared, not in real time

→ **Fairness:**
↳ No starvation

In Absence of user

→ Priorities:
↳ High-Priority

→ Balancing Resources:
↳ Med + Long term Sched
↳ Favour those that underutilize stressed resources

Priorities

→ Multiple ready queues; each queue has same priority across the queue

→ RQ_n → No chance to run due to starvation

King
↳ elder Prince I
↳ elder Prince II
↳ elder prince n

4 States

Starvation Solution? → Reassign priority on age or execution history

Scheduling Policies → Round Robin & Feedback are most practical

(FCFS, SPN, SRT, HRRN)

Selection Function → w: waiting time
e: execution time
s: Total service time

FCFS → Firstcome First serve

SPN → Shortest Proc Next

SRT → Shortest Response Time

HRRN → Highest Response Ratio Next

TAT: Service + Waiting (Finish - Arrival)

Norm TAT: Tr / Ts (Normal / Service)

→ Decision Mode:
Decides when selection fn is required

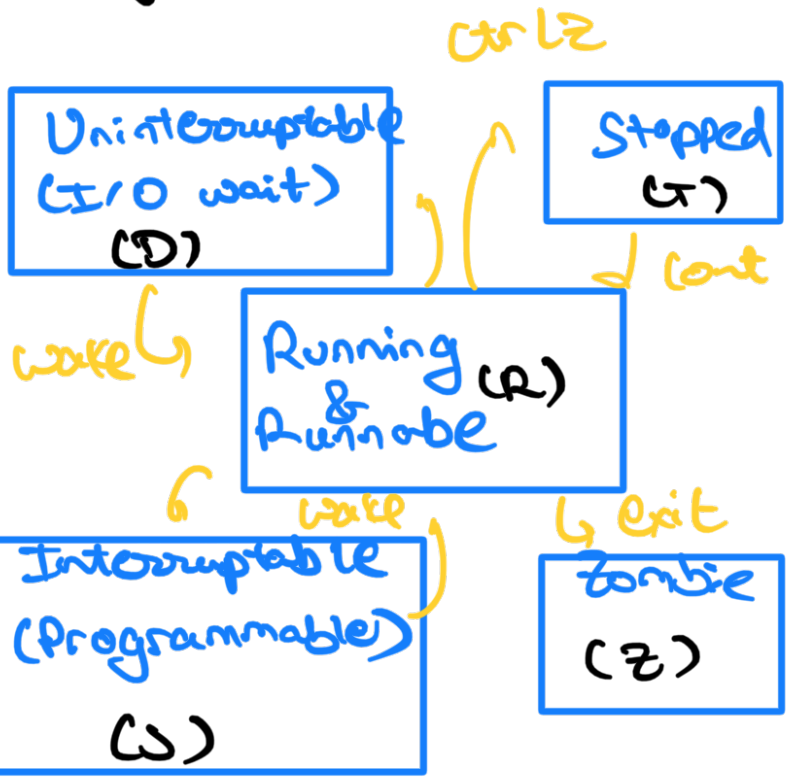
- Non Pre-emptive

↳ OS can't remove
↳ Priority-immune

Pre-emptive
↳ OS can remove
↳ Interrupt,
↳ Periodic
↳ Round Robin

↳ Increases overhead, but switching is worth the trade off

Linux Process States



→ In new: Idle (I)

lscpu → CPU info

ps -aux	→	Which user; what processes & stats
top	→	Dynamic ps ↳ continuously running
htop	→	Similar to top
fork	→	Create
exec	→	Execute
wait	→	Wait
exit	→	Exit

Summary: Uniprocessor scheduling- Objectives, types, criteria for scheduling – Priorities- Scheduling Policies – Decision mode & Selection fn – Linux States (R, D, S, T, Z)