**Preemptive vs Non Preemptive**

→ Overhead increases; but process monopolizing is prevented in preemptive

**FCFS (FIFO)**

→ Most fair in real world

→ Not the most useful in OS though.

→ Process with max waiting time is selected.

**Selection fn^**

→ max [w]

**Type**

→ Non-preemptive

→ Sevice Time ~ CPU Burst Time

**CPU Bound vs I/O Bound Process**

→ Most time executing

On CPU → CPU-Bound

On I/O → I/O Bound

Chat Process

→ Normalised Turnaround time → Relative delay

↳ Tr/Ts < 1 not possible ↳ Arrival&Service

↳ Service time is not greater than Turnaround time

Problems

→ Process with low service time & high waiting buffer

→ Favours CPU Bound processes

↳ I/O will be waiting; soon blocked for I/O operation

↳ If CPUBound is Blocked, CPU remains idle

→ FCFS is not the best approach in it's pure form

→ FCFS with priority queue is an option.

Practice Question

| | | |
|---|---|---|
| A | 0 | 3 |
| B | 1 | 5 |
| C | 3 | 2 |
| D | 9 | 5 |
| E | 12 | 5 |

| Arrival | Finish |
|---|---|
| 0 | 3 |
| 1 | 8 |
| 3 | 10 |
| 9 | 15 |
| 12 | 20 |

| TAT | TS/TS | |
|---|---|---|
| 3 | 1 | |
| 7 | 7/5 | 1.4 |
| 7 | 7/2 | 3.5 |
| 6 | 6/5 | 1.2 |
| 8 | 8/5 | 1.6 |

Round Robin

→ Use clk based preemption to prevent short process suffering

→ Also known as time slicing.

q = 4

| | | |
|---|---|---|
| A | 0 | 3 |
| B | 2 | 6 |
| C | 4 | 4 |
| D | 6 | 5 |
| E | 8 | 2 |

→ T: 0 


T: 3 


T: 7 


T: 11 


T: 15 


As only 2 was left in q

T: 17 


T: 19 


T: 20 

→

| | Finished | Turnaround |
|---|---|---|
| A | 3 | 3 |
| B | 17 | 15 |
| C | 11 | 7 |
| D | 20 | 14 |
| E | 19 | 11 |

Tr / Ts

3/3 = 1

15/6 = 2.5

7/4 = 1.75

14/5 = 2.8

11/2 = 5.5

| | | | Finish | Tr | |
|---|---|---|---|---|---|
| A | 0 | 3 | 4 | 4 | Mean |
| B | 2 | 6 | 18 | 16 | Tr/Ts |
| C | 4 | 4 | 17 | 13 | ↓ |
| D | 6 | 5 | 20 | 14 | 2.71 |
| E | 8 | 2 | 15 | 7 | |

Problems with Round Robin

→ ① Length of time quantum

- If very short
  - Short move quickly
  - More overhead
- If very long
  - Long process gets more time
  - Degenerates to FCFS

Ideal Time Quantum

→ Slightly greater than time required for a typical process execution

→ (2) Relative treatment for CPU & I/O Bound processes

- I/O in Block; CPU would use entire time quantum & returns to Ready
- I/O Bound would have poor performance

Virtual Round Robin

→ Auxiliary queue to Higher priority to I/O bound process sent in blocked

Shortest Process Next or Shortest Job

→ Non preemptive

→ Short process jumps ahead of long process

First

| | | |
|---|---|---|
| A | 0 | 3 |
| B | 2 | 6 |
| C | 4 | 4 |
| D | 6 | 5 |
| E | 8 | 2 |

T=0 


T=3 


T=9 


T=11 


T=15 


T=20 

→ Better than round robin

Problems with SPF

→ Estimate required processing time for each process

→ For batch; programmer can estimate the value for time

→ For interactive, OS keeps running avg

→ Starvation of longer processes

**Summary:** Preemptive vs Non Preemptive - CPU Bound vs I/O Bound - Scheduling strategies - FCFS - Problems - Short proc & I/O bound suffer - Round robin - Every proc for a time quantum - Problems - Quantum can't be very short or very long - I/O suffer relatively - Degenerates to FCFS - Virtual round robin - Priority queue of blocked I/O - SPF - Estimate to interactive - Longest