

Algoritmo FCFS (First-Come First-Served)

- Chegada dos processos
 - P1: 12
 - P2: 2
 - P3: 4
- Tempos de espera na fila (Waiting Time)
 - P1: 0 (primeiro a chegar)
 - P2: 12
 - P3: $(12+2) = 14$
- Turnaround time
 - P1: 12
 - P2: 14
 - P3: 18
- Média do Turnaround time
 - $(12+14+18)/3 = 14,67$

Algoritmo FCFS (First-Come First-Served)

- Exercício 1
 - Calcule waiting time, turnaround time e média para os seguintes casos:
 - P1:4 P2:7 P3:2 P4:10
 - Ordem de chegada na fila dada pela identificação do processo

- Exercício 2

<i>P</i>	<i>Proc. Time</i>	<i>Arrival Time</i>
P1	10	0
P2	5	5
P3	8	10
P4	3	12
P5	10	16

Algoritmo SJF (Shortest Job First)

- Característica
 - Sempre executa o processo da fila que demanda o menor tempo
 - Cada processo é relacionado ao seu tempo de execução
 - Quanto menor o tempo de execução, maior a prioridade dada pelo escalonador para execução
- Problema
 - Determinar o tempo de execução de um processo antes de executá-lo (pode ser feito através de análise histórica de execução do processo)

Algoritmo SJF (Shortest Job First)

- Processos
 - P1: 12 \rightarrow 3o.
 - P2: 2 \rightarrow 1o.
 - P3: 4 \rightarrow 2o.
- Tempos de espera na fila
 - P2: 0 (menor tempo de execução)
 - P3: 2 (segundo menor tempo de execução)
 - P1: $(4+2) = 6$
- Turnaround time
 - P1: 18
 - P2: 2
 - P3: 6
- Média (somatório dos turnaround times/nro de procs)
 $(18+2+6)/3 = 8,67$

Algoritmo SJF (Shortest Job First)

- Exercício 3
 - Calcule waiting time, turnaround time e média para os seguintes casos:
 - P1:4 P2:7 P3:2 P4:10
- Exercício 4

<i>P</i>	<i>Proc. Time</i>	<i>Arrival Time</i>
P1	10	0
P2	5	5
P3	8	10
P4	3	12
P5	10	16

Algoritmos baseados em prioridades

- **Características**

- Cada processo possui uma prioridade
- Processos com maior prioridade são executados antes de processos com menor prioridade

- **Problema**

- Postergação indefinida: um processo pode nunca ser executado
- **Solução:** incrementar periodicamente a prioridade dos processos (*aging*)

- Algoritmos preemptivos vs não preemptivos

- **preemptivos:** se o processo submetido tiver prioridade maior que a do processo em execução, então o processo em execução perde a CPU para o processo com maior prioridade
- **não preemptivos:** se o processo submetido tiver prioridade maior que a do processo em execução, então o processo com maior prioridade entra na ready-list e espera acabar a utilização da CPU pelo processo em execução

Algoritmos baseados em prioridades

- Exercício 5
 - SJF com preempção
 - Prioridade é para processo com menor tempo de processamento
 - Calcular:
 - Waiting time
 - Turnaround time
 - Média de turnaround time dos processos

P	PT	AT
P1	10	0
P2	3	5
P3	10	7
P4	5	17

Algoritmo Round-robin

- Característica

- Lista de processos é implementada como uma fila circular
- **quantum**: tempo máximo no qual um processo pode utilizar a CPU ininterruptamente -> evita monopólio da CPU por um único processo
- Funcionamento:
 1. pega o 1o. processo da ready list e o executa
 2. processo executa por uma fatia de tempo, é preemptado e reinserido no final da ready list
 3. volta para o 1o. passo

- Problemas

- overhead para salvamento e restauração de contexto
- Definição do quantum:
 - **muito grande**: se aproxima do FCFS
 - **muito pequeno**: aumenta overhead de troca de contexto (valores comuns: 10-100 miliseg)

Algoritmo Round-robin

- Exercício 6:
 - Quantum = 5
 - Calcular:
 - Waiting time,
 - Turnaround time
 - Média de turnaround time

P	PT	AT
P1	10	0
P2	5	0
P3	8	4
P4	3	10