BCC722

Programação de Sistemas em Tempo Real

Escalonamento de Processos

Prof. Charles Garrocho

Conceitos básicos

- Objetivo da multiprogramação:
  - Utilização máxima da CPU
- Processos normalmente alternam picos de processamento (uso da CPU) e E/S
- Quando um processo começa um pico de E/S outro deve assumir a CPU para evitar ociosidade
- Por outro lado, nenhum processo deve controlar a CPU indefinidamente p/ melhorar a interação

Escalonador da CPU (*scheduler*)

- Controla a mudança de estado dos processos
- Escalonamento ocorre quando um processo:
  a) chaveia de “em execução” para “em espera”
  b) chaveia de “em execução” para “pronto”
  c) chaveia de “em espera” para “pronto”
  d) termina

Preempção ou não preempção

- Escalonamento não preemptivo
  - Processo só deixa a CPU se tiver que esperar por E/S ou intencionalmente
  - Implementação mais simples do escalonador
- Escalonamento preemptivo
  - Periodicamente o escalonador interrompe o processo em execução e muda-o para “pronto”
  - Escalonador mais complexo
  - Compartilhamento da CPU é garantido

Dispatcher (despachante)

- Módulo responsável por dar o controle da CPU a cada processo no escalonador
  - Troca de contexto de execução
  - Chaveamento para modo usuário
  - Desvio para o ponto apropriado do programa
- Latência de despacho:
  - Tempo gasto para o despachante interromper um processo e iniciar a execução de outro

Critérios de escalonamento

- **Taxa de utilização de CPU**: é a fração de tempo durante a qual ela está sendo ocupada;
- **Throughput** que são números de processos terminados por unidade de tempo;
- **Turnaround** que é o tempo transcorrido desde o momento em que o software entra e o instante em que termina sua execução;
- **Tempo de resposta**: intervalo entre a chegada ao sistema e inicio de sua execução;
- **Tempo de espera**: soma dos períodos em que o programa estava no seu estado pronto.

Objetivos do Escalonamento

- **Ser justo**: Todos os processos devem ser tratados igualmente, tendo possibilidades idênticas de uso do processador, devendo ser evitado o adiamento indefinido.
- **Maximizar a produtividade** (throughput): Procurar maximizar o número de tarefas processadas por unidade de tempo.
- **Ser previsível**: Uma tarefa deveria ser sempre executada com aproximadamente o mesmo tempo e custo computacional.
- **Minimizar o tempo de resposta** para usuários interativos.
- **Maximizar o número** possível de usuário interativos.
- **Balancear o uso de recursos**: o escalonador deve manter todos os recursos ocupados, ou seja, processos que usam recursos sub- utilizados deveriam ser favorecidos.

First-come, first-serve (FCFS)

Processo	Duração
P1	24
P2	3
P3	3

Ordem de chegada: P2, P3, P1

- Gantt Chart do escalonamento:

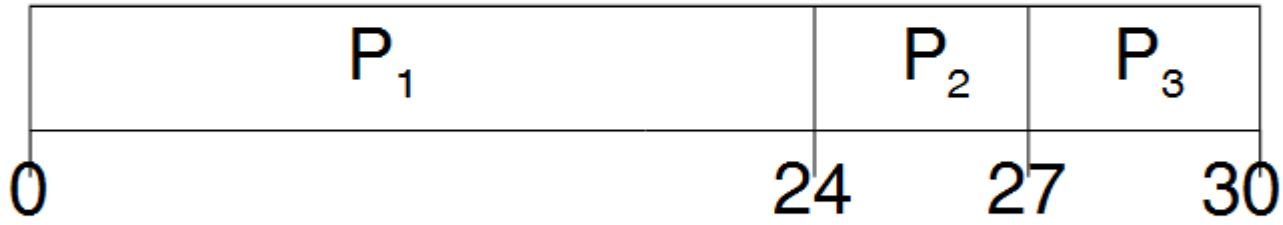
- Tempo de espera: P1 = 6; P2 = 0; P3 = 3
- Tempo de espera médio: (6 + 0 + 3)/3 = 3

First-come, first-serve (FCFS)

Processo	Duração
P1	24
P2	3
P3	3

Ordem de chegada: P1, P2, P3

- Gantt Chart do escalonamento:



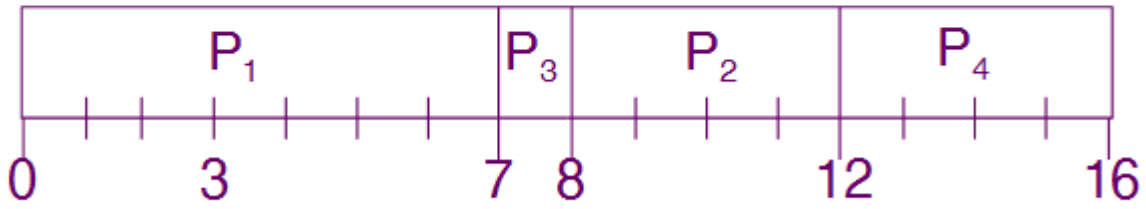
- Tempo de espera: P1 = 0; P2 = 24; P3 = 27
- Tempo de espera médio: (0 + 24 + 27)/3 = 17
- Efeito comboio: pequenos atrasados pelo grande

Shortest-Job-First (SJF)

- Associa-se a cada processo seu próximo pulso
  - Próximo processo: o de menor pulso
- SJF preemptivo:
  - Se um novo processo chega ao estado "pronto" com um tempo de alocação menor que o tempo restante do processo em execução, então há preempção (interrupção)
- SJF não preemptivo:
  - Uma vez que a CPU é atribuída a um processo, este não pode ser interrompido até completar a execução do processo.
- SJF é ótimo quanto ao tempo médio de espera

SJF não preemptivo

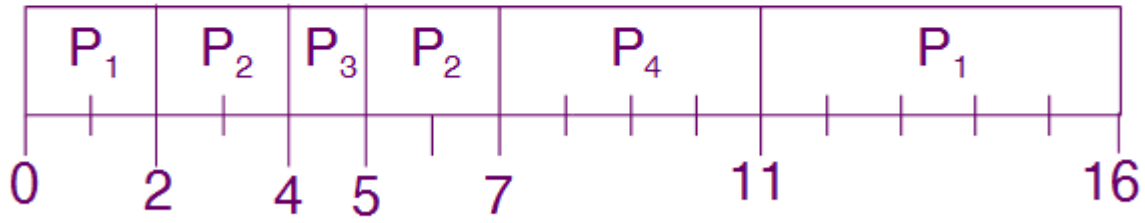
Processo	Chegada	Duração
P1	0	7
P2	2	4
P3	4	1
P4	5	4



Tempo de espera médio = (0 + 6 + 3 + 7)/4 = 4

SJF preemptivo

Processo	Chegada	Duração
P1	0	7
P2	2	4
P3	4	1
P4	5	4



Tempo de espera médio = (9 + 1 + 0 + 2)/4 = 3

Escalonamento com prioridades

- Um valor de prioridade (inteiro) p/ cada proc.
- A CPU é alocada p/ o proc. de maior prioridade
  - Usualmente, menor valor = maior prioridade
  - Preemptivo ou não preemptivo
- SJF é um escalonamento com prioridade, onde a prioridade é o tamanho previsto do pulso de CPU

Escalonamento com prioridades

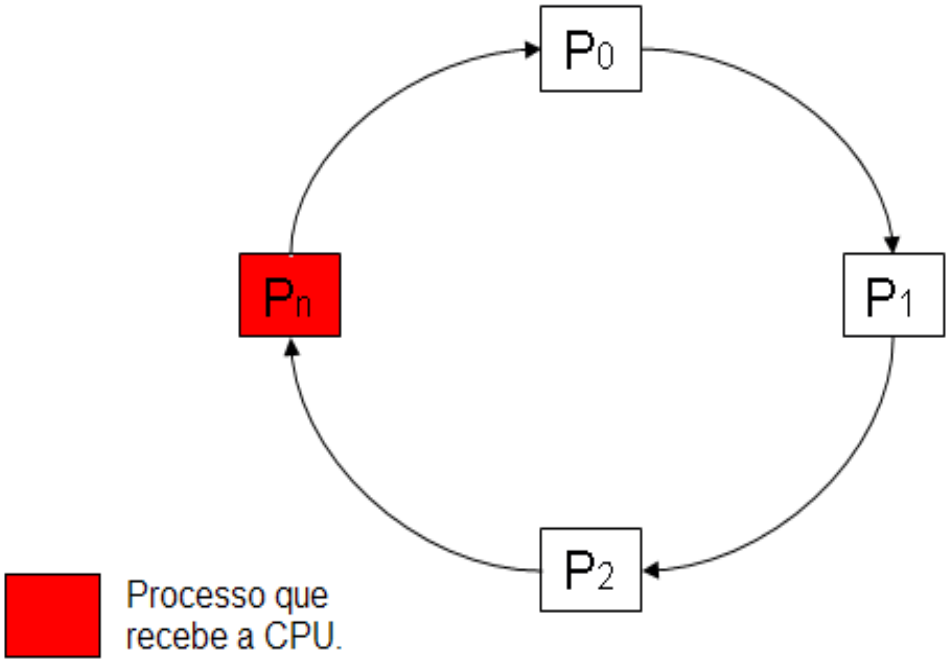
- Problema: inanição (*starvation*)
  - Processos de baixa prioridade podem nunca ser executados
- Solução: envelhecimento (*aging*)
  - Prioridade de cada processo que fica na fila aumenta à medida que o tempo passa

Round Robin (RR)

- Cada processo recebe a CPU por um tempo pequeno (quantum)
  - Usualmente, de 10 a 100 milissegundos
  - Ao fim do tempo, processo é trocado e vai para a fim da fila de prontos
- Imune a problemas de **starvation** que são tarefas que nunca são executadas em função de ter prioridade inferior as demais.

Round Robin (RR)

Todos os processos são armazenados em uma fila circular. Como no exemplo abaixo.



Round Robin (RR)

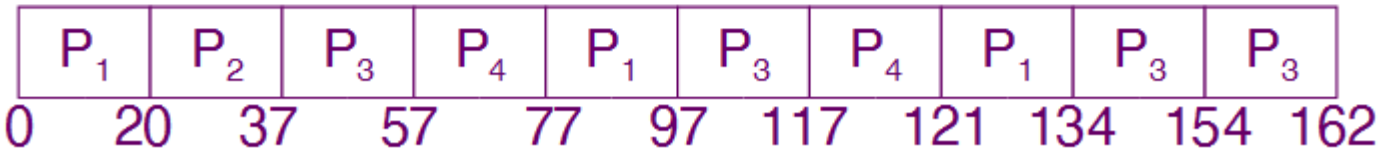
- **Exemplo**: Se o quantum é 100 milisegundos e a tarefa leva 250 milisegundos para completar, o agendamento round-robin suspenderá a tarefa após os primeiros 100 milisegundos e dara a outra tarefa da fila, o mesmo tempo.
- Esse tarefa sera executa portanto após 3 agendamentos a saber (100 ms + 100 ms + 50 ms).
- A interrupção da tarefa é conhecida como preempção.

Round Robin (RR)

- Exemplo (quantum = 20 unidades de tempo)

Processo	Duração
P1	53
P2	17
P3	68
P4	24

- Gantt chart:



Escalonamento (filas) multi-nível

- Escalonamento em multinivel consiste em dividir os processos em diferentes grupos, com diferentes requisitos de tempo e resposta.
- A cada grupo é associada uma fila de prioridades, conforme a sua importância.
- O escalonamento deve ser feito entre as filas:
  - **Prioridades da fila:** atender primeiro a fila de processos interativos e depois aos da fila de processos batch (em lote);
  - **Time slice:** cada fila recebe uma quantidade de tempo da CPU para escalonamento entre os seus processos.

Filas multinível com realimentação

- Um processo pode se mover entre filas
  - Forma de implementar envelhecimento, p.ex.

- Escalonamento pode ser definido por:
  - no. de filas
  - algoritmo de escalonamento de cada fila
  - método usado para promover/rebaixar um processo
  - método usado para decidir em que fila cada processo entra no sistema

Ferramentas Nice e Renice

O comando **nice** altera o nível de prioridade dos processos. Exemplo:

charles:~# nice -n +15 top

Já o comando **renice** altera em tempo real a prioridade dos processos. Exemplo:

charles:~# renice -n -19 -p 1