Sistemas Operacionais

Processos - Parte 3

Prof. Dr. Fábio Rodrigues de la Rocha

UTFPR

Algoritmos de Escalonamento

Nas aulas anterior vimos o ESCALONADOR, que é uma parte do SO responsável pela escolha de qual tarefa deve ter a chance de executar.
Classificação:

- Escalonamento não-preemptivo;
  - chamadas de sistema (*system call*);
  - liberação explícita do controle (escalonamento cooperativo).
- Escalonamento preemptivo.
  - chamadas de sistema (*system call*);
  - Passagem do tempo.

UTFPR

Algoritmos de Escalonamento - Métricas

- Utilização da CPU – utilizar o máximo da CPU;
- Throughput – número de processos finalizados em um dado intervalo de tempo;
- Tempo de execução – tempo para finalizar a execução de um determinado processo no sistema;
- Tempo de espera – quantidade de tempo que o processo ficou na fila de prontos;
- Tempo de resposta – tempo entre a requisição e a saída do primeiro resultado (sistemas interativos).

UTFPR

Algoritmos de Escalonamento - Objetivo

- Utilização máxima da CPU
- Throughput máximo
- Tempo de execução mínimo
- Tempo de espera mínimo
- Tempo de resposta mínimo

Algoritmos de Escalonamento

Algoritmos de escalonamento:

- Round-Robin;
- FIFO;
- Menor ciclo de processador;
- Prioridades.
  - Filas múltiplas.

Algoritmos de Escalonamento - Fatia de tempo

Fatia de tempo

O algoritmo de fatia de tempo (também chamado **round-robin**) faz com que cada processo tenha direito a executar por uma fatia de tempo (**quantum**). Existe um relógio no sistema que gera periodicamente uma interrupção (a cada quantum) e neste momento o escalonador verifica qual o novo processo deve executar. Além disso, se um processo estiver executando e realizar uma chamada de sistema (operação demorada), ele será suspenso e em seu lugar outro processo será posto em execução.

Algoritmos de Escalonamento - Fatia de tempo

Perguntas importantes:

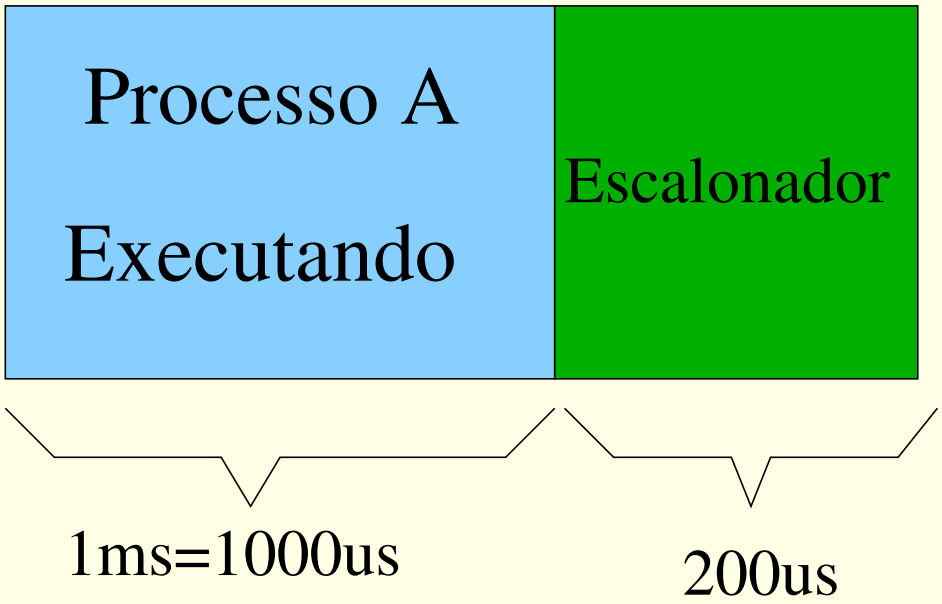
- Qual o tamanho da fatia de tempo ? Se a fatia de tempo for pequena ? Se for grande ?

Exemplos: (considere um tempo de troca de contexto de 200us)
a) Fatia de 1ms
b) Fatia de 1s

UTFPR

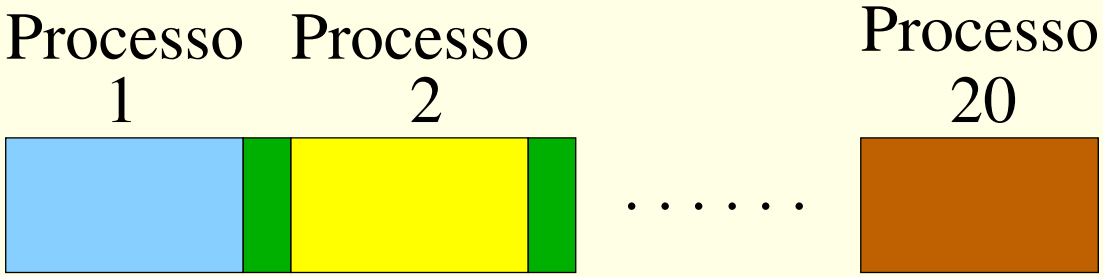
Algoritmos de Escalonamento - Fatia de tempo

Se a fatia for 1ms:



Algoritmos de Escalonamento - Fatia de tempo

Se a fatia for 1s:



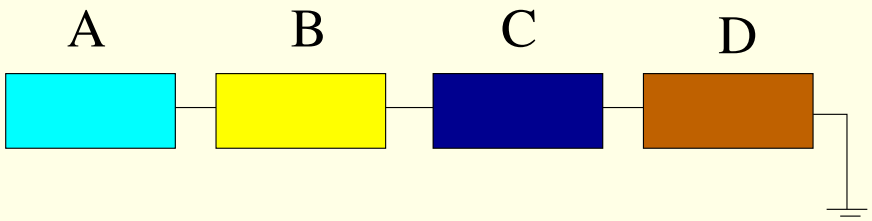
Algoritmos de Escalonamento - Ordem de Chegada

Ordem de chegada

O escalonador que emprega a técnica da "ordem de chegada" (também chamado de FIFO - *First-In First-Out*) escolhe o processo que deve ganhar o processador como sendo o processo que está na frente da lista de processos aptos para executar.

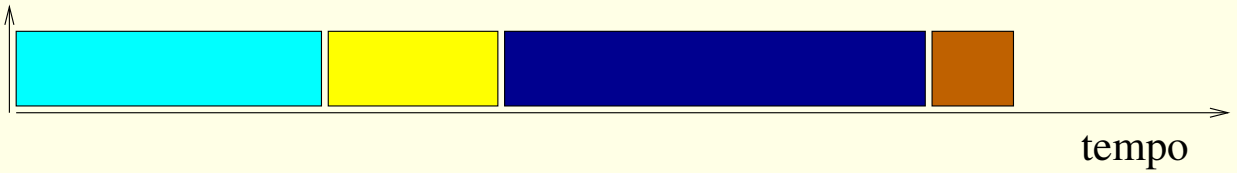
Algoritmos de Escalonamento - Ordem de Chegada

Lista de processos Aptos para executar



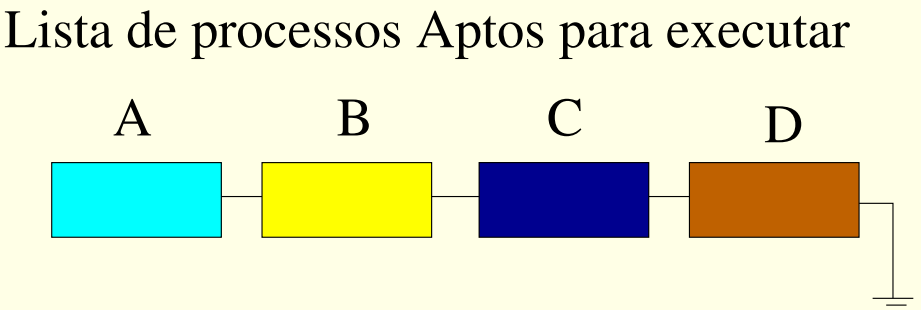
Na Execução

Processo:	Tempo que usa da CPU
A	12
B	8
C	15
D	5



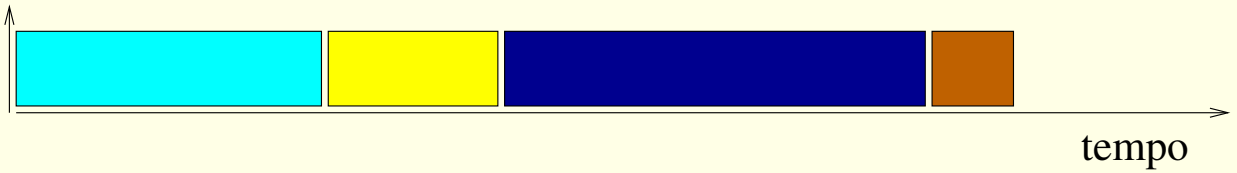
Qual o tempo médio que os processos esperaram na lista para iniciarem sua execução ?

Algoritmos de Escalonamento - Ordem de Chegada



Na Execução

Processo:	Tempo que usa da CPU
A	12
B	8
C	15
D	5



Qual o tempo médio que os processos esperaram na lista para iniciarem sua execução ?

Algoritmos de Escalonamento - Ciclo de processador menor antes

Ciclo de processador menor antes

O algoritmo FIFO visto no slide anterior tinha o problema de resultar num tempo de espera na fila grande. Isto por que ele escolhia os processos na ordem que eles estavam na lista. O algoritmo "Ciclo de processador menor antes" ou (SJF - *Shortest Job First*) escolhe os processos de uma lista ordenada pelo seu tempo de processamento. Assim, se o processo D tem um tempo pequeno, ele será escolhido, depois o B, A e C.

Algoritmos de Escalonamento - Ciclo de processador menor antes

Utilizando o mesmo conjunto de processos do FIFO, temos que o tempo médio de espera é: (0+5+13+25)/4 = 10.75 que é menor do que 16.75 resultante do método FIFO.
Pergunta importante ? Como o projetista do SO sabe quanto um processo demorada para executar ?
Se existir um processo com tempo de execução muito grande, o que ocorre ?

UTFPR

Algoritmos de Escalonamento - Ciclo de processador menor antes

Como determinar o tempo de processador necessário para um processo ?

- Estimativas
- Normalmente baseadas nos ciclos de execução anteriores

Algoritmos de Escalonamento - Prioridades

Prioridades

O Algoritmo Round Robin considera que todos os processos têm a mesma importância. Ocorre que em geral isso não é verdade e existe interesse em atribuir uma importância relativa a determinados processos e forma que estes tenham uma importância maior do que outros.

Chamamos de prioridade um valor que indica a importância que um processo possui. Assim, podemos ter processos que são mais importantes que outros. Esse valor de importância (que podemos assumir que 1 é mais importante de todos, depois vem o número 2 e assim por diante) pode ser utilizado pelo escalonador para decidir qual processo deve ter o direito de executar. Como exemplo podemo ter:

Algoritmos de Escalonamento - Prioridades

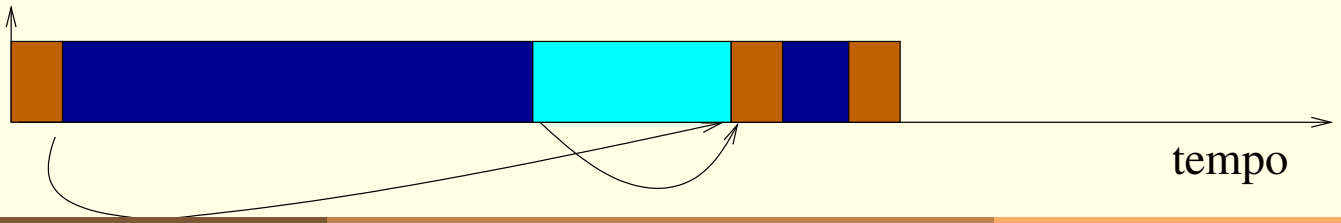
Processo	Prioridade	tempo que usa da CPU
A	3	12
B	4	8
C	2	15
D	1	5

Algoritmos de Escalonamento - Prioridades

O escalonamento por prioridades, assim como o SJF pode levar a uma situação onde um processo de baixa prioridade nunca é escolhido para executar (starvation). Veja o exemplo:

Na Execução

Processo:	Tempo que usa da CPU	Prioridade
A	12	3
B	8	4
C	15	2
D	5	1



Algoritmos de Escalonamento - Prioridades

Redução de prioridade - *Aging*