

Sistemas Operacionais

André Luiz da Costa Carvalho

andre@icomp.ufam.edu.br

Aula 06 - Escalonamento por Loteria

Aula de Hoje

- 1 Escalonamento por Loteria
- 2 Mecanismos de Bilhetes
- 3 Implementação
 - Distribuição de Bilhetes
- 4 Stride Scheduler

Aula de Hoje

Na aula de hoje, veremos um outro tipo de escalonador: o **escalonador de tempo proporcional** (ou **fair share**).

São baseados em um conceito simples: Ao invés de focar em otimizar tempo de resposta ou execução, o escalonador deve focar em garantir que cada processo tenha acesso a uma porcentagem específica da CPU.

Definida pela natureza do workload.

É uma idéia antiga, mas que foi ressuscitada recentemente e é usada em sistemas modernos.

Aula de Hoje

- 1 Escalonamento por Loteria
- 2 Mecanismos de Bilhetes
- 3 Implementação
 - Distribuição de Bilhetes
- 4 Stride Scheduler

Bilhetes representando a sua parte

Escalonadores proporcionais são baseados no conceito de loteria de **bilhetes**.

Bilhetes representam a parte da CPU que cada processo/usuário tem acesso.
A porcentagem dos bilhetes representa a porcentagem do recurso a que se tem acesso.

Imagine 2 processos A e B, com A tendo 75 bilhetes e B 25. Isso significa que A deve receber 75% da CPU e B 25%.

Escalonamento por loteria

Escalonamento proporcional (ou por **loteria**) define o próximo processo probabilisticamente e não estatisticamente.

Um sorteio é feito a cada "Rodada".

Sorteio é simples: sabendo o número N de bilhetes, basta escolher um número entre zero e N que define o vencedor.

O processo vencedor entra na CPU.

Exemplo

Processos A e B com A tendo 75 bilhetes e B 25. Loteria deu:

63 85 70 39 76 17 29 41 36 39 10 99 68 83 63 62 43 0 49 49

O escalonamento resultante é:

A A A A A A A A A A A A A A A A
 B B B B

Exemplo

A aleatoriedade da loteria leva a um resultado probabilisticamente próximo do esperado, mas sem garantias.

No exemplo anterior, B tem apenas 4 de 20 time slices (20%).

Porém, quanto mais tempo de competição, mais próximo das porcentagens desejadas.

Na prática, bilhetes (tickets) podem ser usados para representar graus de participação.

Aleatoriedade

Um dos aspectos mais interessantes do escalonamento por loteria é a aleatoriedade. Abordagens com componente aleatório tem pelo menos três grandes vantagens.

- ❶ **Evita a existência de casos críticos:** Boa parte das heurísticas sempre possuem um tipo de workload que as levam ao pior comportamento possível. Ex: SJF com processos pequenos constantes, LRU com workloads cíclicos.
- ❷ **Leve:** Algoritmos baseados em aleatoriedade são extremamente leves, precisando de pouca memória e CPU para controlar os procesos.
- ❸ **Rápido:** Se a geração de números aleatórios é eficiente o resto do processo é rápido.

Aula de Hoje

- 1 Escalonamento por Loteria
- 2 Mecanismos de Bilhetes
- 3 Implementação
 - Distribuição de Bilhetes
- 4 Stride Scheduler

Mecanismos de bilhetes

Diversos mecanismos para manipular a loteria:

- **Ticket Currency:** Um usuário pode distribuir seus bilhetes usando as grandezas que quiser, que o sistema traduz para um valor global.

Digamos que dois usuários estejam usando o sistema, e cada um tenha 100 bilhetes.

- A roda dois processos, A1 e A2, e dá 500 bilhetes para cada.
- B roda um processo, B1 e dá 10 tickets para ele.

Convertendo aos valores que cada usuário tem:

Mecanismos de bilhetes

Diversos mecanismos para manipular a loteria:

- **Ticket Currency:** Um usuário pode distribuir seus bilhetes usando as grandezas que quiser, que o sistema traduz para um valor global.

Digamos que dois usuários estejam usando o sistema, e cada um tenha 100 bilhetes.

- A roda dois processos, A1 e A2, e dá 500 bilhetes para cada.
- B roda um processo, B1 e dá 10 tickets para ele.

Convertendo aos valores que cada usuário tem:

- A1 e A2 terão 50 bilhetes globais.
- B1 terá 100 bilhetes globais.

Transferencia e Inflação

Um processo também pode **transferir** temporariamente seus bilhetes para outro.

Útil em ambientes cliente servidor, onde o cliente pode repassar seus tickets para um servidor que está fazendo algo para ele.

Pode rolar também **inflação**:

Em ambientes em que os **processos confiam um no outro**, um processo que precise pode aumentar artificialmente o número de bilhetes que ele possui, sem precisar avisar outros processos.

Aula de Hoje

- 1 Escalonamento por Loteria
- 2 Mecanismos de Bilhetes
- 3 Implementação**
 - Distribuição de Bilhetes
- 4 Stride Scheduler

Implementação

Grande vantagem do escalonamento por loteria é sua simplicidade.

- Basta um bom gerador de números aleatórios, uma estrutura de processos e um contador do total de processos.



Para decidir qual processo será escolhido, deve-se sortear um número entre zero e o total de bilhetes (400 nesse caso).

Como você acha que pode ser o processamento deste método?

Como [otimizar](#) este processamento?

Exemplo

Dois processos com o mesmo número de bilhetes (100) e o mesmo tempo de execução R .

Neste cenário, o ideal seria que ambos terminassem ao mesmo tempo ([fairness](#)).

- Contudo, a aleatoriedade não garante que isso ocorrerá sempre.

Injustiça

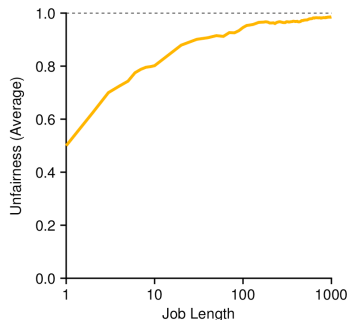
Precisamos quantificar a **injustiça**:

- **Unfairness** U é o tempo em que o primeiro processo termina dividido pelo tempo em que o segundo termina.
- Se ambos tem o mesmo tempo de execução $R = 10$, e o primeiro termina no tempo 10 e o segundo no tempo 20, o $U = \frac{10}{20} = 0.5$.

Injustiça

Precisamos quantificar a **injustiça**:

- **Unfairness** U é o tempo em que o primeiro processo termina dividido pelo tempo em que o segundo termina.
- Se ambos tem o mesmo tempo de execução $R = 10$, e o primeiro termina no tempo 10 e o segundo no tempo 20, o $U = \frac{10}{20} = 0.5$.



Bilhetes

Assim como as prioridades, a distribuição de bilhetes está no cerne da qualidade dos escalonadores.

Muito dos critérios que usamos para escalonamento por prioridade podem ser utilizados em escalonamento por loteria.

A grande diferença é que:

- Loteria garante probabilisticamente que não haverá starvation.
- "Empréstimo" de bilhetes da flexibilidade aos processos.

Aula de Hoje

- 1 Escalonamento por Loteria
- 2 Mecanismos de Bilhetes
- 3 Implementação
 - Distribuição de Bilhetes
- 4 Stride Scheduler

Stride Scheduler

Por que aleatório?

- Simples e aproximadamente correto.

Contudo, não há garantias que as proporções serão mantidas exatamente.

Por isso, foi proposto o **Stride Scheduling**, ou escalonador de passo, um escalonador determinístico baseado em justiça (*fairshare*).

Stride Scheduler

Cada processo tem um *stride*, que é inversamente proporcional ao número de bilhetes.

No Exemplo:



O *stride* de cada processo pode ser calculado dividindo um número grande pela quantidade de bilhetes de cada processo.

Por exemplo com 10.000 o *stride* de A, B e C seria 100,200 e 40, respectivamente. A cada vez que o processo roda, um contador de passos é incrementado pelo valor de *stride*, controlando o quanto ele foi rodado.

Stride Scheduling

O Escalonador então usa o stride e o passo para determinar qual processo é o próximo.

A idéia é sempre escolher o processo com o menor contador de passos, e assim que ele rodar, atualizar seu passo pelo valor de **stride**.

Exemplo

No nosso exemplo A,B e C tem strides de 100,200 e 40 e passos 0. Então qualquer processo pode ser escolhido. (Assuma que foi A).

Quando A acabar, seu passo vai ser incrementado por 100 (o stride). Então rodamos B, que fica com passo 200. Então C, que fica com 40.

Neste momento o algoritmo escolherá o passo mais baixo, C, e rodará de novo para passo 80.

Exemplo

Pass(A) (stride=100)	Pass(B) (stride=200)	Pass(C) (stride=40)	Who Runs?
0	0	0	A
100	0	0	B
100	200	0	C
100	200	40	C
100	200	80	C
100	200	120	A
200	200	120	C
200	200	160	C
200	200	200	...

Figure 9.3: **Stride Scheduling: A Trace**

As proporções são exatamente iguais ao esperado pelos bilhetes!

Desvantagens

Por que usar a loteria se existe o stride?

Stride tem uma grande desvantagem: O passo dos processos é **global**!

O que isto quer dizer?

Desvantagens

Por que usar a loteria se existe o stride?

Stride tem uma grande desvantagem: O passo dos processos é **global**!

O que isto quer dizer?

O que acontece quando chega um processo novo?

Na aula de hoje

Vimos o conceito de escalonamento de tempo proporcional.

Vimos duas implementações: Loteria e Stride.

Vimos uma métrica de Justiça (fairness)

Na aula de hoje

Onde são usados?

Estes escalonadores são problemáticos para CPU diretamente. Tendo problemas com processos iterativos (E/S).

Ainda tem o problema de definir os bilhetes.

São usados em ambientes virtualizados, por exemplo, como o VMWare entre outros.