

Sistemas Operacionais - Trabalho 1

Implementação de um escalonador de processos

O primeiro trabalho da disciplina consiste na implementação de um simulador para o escalonamento de processos em um sistema operacional hipotético. Para o escalonamento dos processos, assume-se que estes possam encontrar-se nos estados *Ready*, *Running*, *Blocked* ou *Exit*.

Assume-se que os processos já foram admitidos pelo sistema operacional, dessa forma o simulador deverá apenas executar os processos de acordo com um algoritmo de escalonamento baseado em prioridades (descrito adiante) que considera os seguintes parâmetros:

- Nome do processo;
- Surto de CPU: é o tempo de CPU utilizado antes da realização de uma operação de entrada e saída;
- Tempo de E/S: é o tempo em que um processo fica bloqueado em função de uma solicitação de entrada e saída ao sistema operacional;
- Tempo total de CPU: é o tempo de processamento do processo, incluindo o tempo de surto (se este ocorrer);
- Ordem: é um valor que define a prioridade dos processos em caso de empate. Ordem é um valor atribuído inicialmente e que irá variar seguindo uma política rotativa.
- Prioridade: é um valor arbitrário definido para cada processo, onde um valor maior representa maior prioridade, ou seja, mais tempo em que poderá executar;

Descrição do algoritmo

O sistema operacional hipotético utiliza um algoritmo de escalonamento baseado em prioridades. Em um sistema onde diversos processos estão em execução, cada processo possui um determinado número de créditos (inicialmente o número de créditos é igual à prioridade do processo). O processo com o maior número de créditos e que estiver na fila de prontos é selecionado pelo escalonador. Um processo que é escalonado utiliza tempo de CPU, sendo esse tempo um surto de CPU anterior à um bloqueio (no caso de um processo com E/S) ou o tempo total de CPU. Em processos com operações de E/S, o surto é descontado do tempo total de CPU.

A cada vez em que for escalonado devido a um temporizador (1ms), o processo em execução perde um crédito. Quando seus créditos chegarem em zero, ou houver algum bloqueio em função de operação de entrada e saída, o escalonador deve selecionar outro processo para execução. Se nenhum processo na fila de prontos possuir créditos, o algoritmo faz uma atribuição de créditos a todos os processos (incluindo processos bloqueados), de acordo com a fórmula $cred = cred/2 + prio$ ¹. O seu escalonador deverá apresentar o escalonamento dos processos, sua utilização de CPU, o tempo de execução (*turnaround time*), bem como os estados nos quais os processos encontram-se em uma linha de tempo.

O parâmetro *ordem* é utilizado como um critério de desempate no caso de múltiplos processos possuírem a mesma prioridade. Além disso, um processo que acabou de perder sua prioridade (a prioridade atual é zero), será o que possui a menor ordem atual, tendo os outros processos uma atualização de sua ordem.

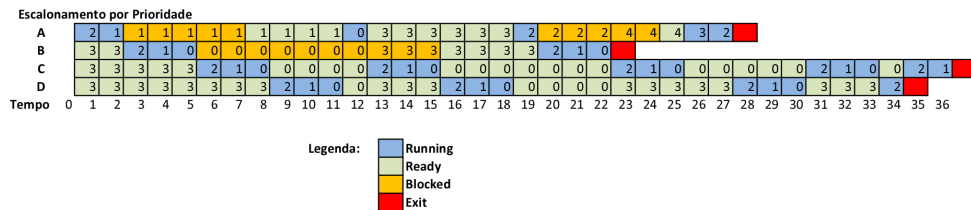
Exemplo de escalonamento

Considerando o seguinte conjunto de processos, onde alguns são puramente CPU bound (processos C e D) e outros possuem operações de E/S:

Processo	Surto de CPU	Tempo de E/S	Tempo total de CPU	Ordem	Prioridade
A	2ms	5ms	6ms	1	3
B	3ms	10ms	6ms	2	3
C	-	-	14ms	3	3
D	-	-	10ms	4	3

O escalonamento resultante é:

¹ *cred* representa o número de créditos e *prio* representa a prioridade inicial do processo.



Observações

1. O simulador deverá suportar qualquer número de processos, especificados em uma lista (cada elemento da lista possui os parâmetros apresentados na descrição).
2. A saída do escalonamento e informações adicionais deverão ser apresentadas em modo texto no terminal.
3. Qualquer linguagem de programação poderá ser utilizada. Espera-se que boas práticas de programação sejam utilizadas e que seja utilizada programação modular.
4. Os grupos deverão possuir 3 ou 4 integrantes. É responsabilidade do grupo organizar-se para o desenvolvimento e apresentação da atividade.
5. A data de entrega do trabalho é 10/04. As apresentações serão realizadas nos dias 10/04 e 15/04.
6. A entrega deverá ser realizada pelo Moodle em um arquivo .tar.gz ou .zip, contendo o código fonte e um arquivo texto com os nomes completos dos integrantes. Apenas um dos integrantes do grupo deverá efetuar a entrega.