

Primeiro Exercício-Programa

Sistemas Operacionais (ACH 2044), 2024

Professor Norton Trevisan Roman

Ana Clara Segal Vidal Pessanha (14677464)

Gabriela Alcaide (14746492)

Giovanna Almeida Albuquerque (13687515)

Júlia Du Bois Araújo Silva (14584360)

Rodrigo Gonçalves Cardoso (14658330)

Organização e Funcionamento do código

Nosso grupo escolheu a linguagem Python para implementar o escalonador de processos, e, resumidamente, a estrutura do código se mostra da seguinte forma:

- Funções de leitura:
 - leituraQuantum: lê o quantum (tempo de execução de cada processo) do arquivo “quantum.txt” e guarda na variável “quantum”.
 - OBS.: Para a realização dos testes para determinar o melhor valor de `n_com`, iremos modificar o valor do quantum diretamente nesse arquivo.
 - leituraPrioridades: lê a prioridade dos processos do arquivo “prioridades.txt” e vai adicionando na lista “prioridades”
 - readAllFiles: varre a pasta `./programas` para identificar os arquivos disponíveis, lê o quantum a ser usado utilizando a função “leituraQuantum”, lê as prioridades dos processos utilizando a função “leituraPrioridades” e todos os outros arquivos que representam os programas são armazenados na lista “programasNomes”, que contém os caminhos para esses arquivos.
- Classe BCP (Bloco de Controle de Processos): cada processo tem seu BCP, que armazena informações de seu estado, como nome, estado (“Pronto”, “Executando” ou “Bloqueado”, prioridade, contador de programa (PC), estado dos registradores X e Y e há quanto tempo o processo está bloqueado;
- Classe Processo: associa cada processo a um BCP, lendo suas instruções e as armazenando para serem executadas.
- Classe TabelaDeProcessos: mantém duas filas, uma para os processos prontos e outra para os processos bloqueados e garante que a fila de prontos sempre vai estar ordenada de acordo com a prioridade dos processos.
- Escalonador: função principal do programa. Controla a execução dos processos, executando suas instruções em quanta definidos. Para isso, verifica instruções específicas (X=, Y=, E/S, SAIDA) e altera o estado do processo conforme necessário.

- Criação do logfile: cada ação do escalonador é registrada em um arquivo de log (logXX.txt)

Testes

A fim de identificar qual o valor de n_com (número de comandos por programa) mais adequado para esta máquina fictícia, usamos o intervalo de 1 a 21 n_com , considerando que 21 é o número máximo de instruções por programa. Para identificar o melhor valor, consideramos que o quantum será de n_com comandos, alterando o valor do quantum no arquivo “quantum.txt”. Ao executar o escalonador para cada um dos quanta, obtivemos 21 logfiles, com as médias descritas na tabela abaixo:

N_COM (QUANTUM)	MÉDIA DE TROCAS	MÉDIA DE INSTRUÇÕES
1	14,9	0,4892
2	8,1	0,8607
3	6,1	1,1525
4	5,2	1,3076
5	4,4	1,5454
6	4,2	1,6585
7	4,0	1,7894
8	3,8	1,7894
9	3,7	1,8378
10	3,7	1,8378
11	3,5	1,9428
12	3,5	1,9428
13	3,5	1,9428
14	3,5	2
15	3,4	2
16	3,4	2

17	3,4	2
18	3,4	2
19	3,4	2
20	3,4	2
21	3,4	2,0606

Gráficos

Gráfico de Quantum vs Média de Instruções

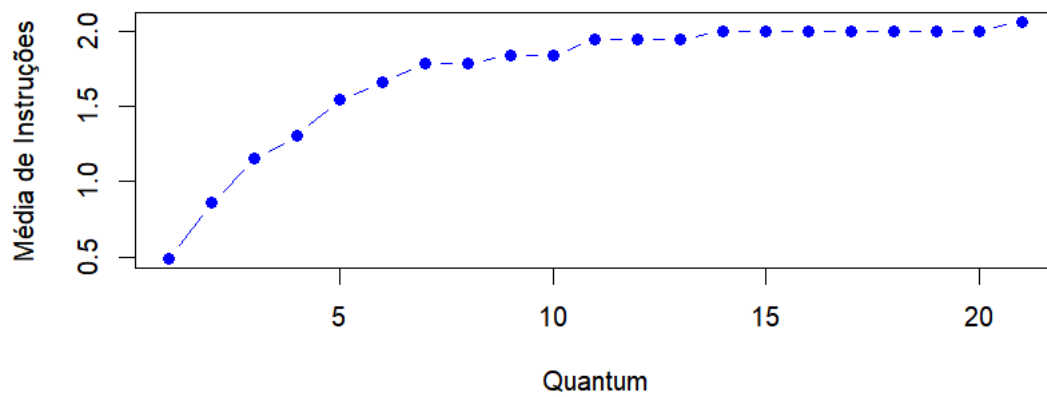
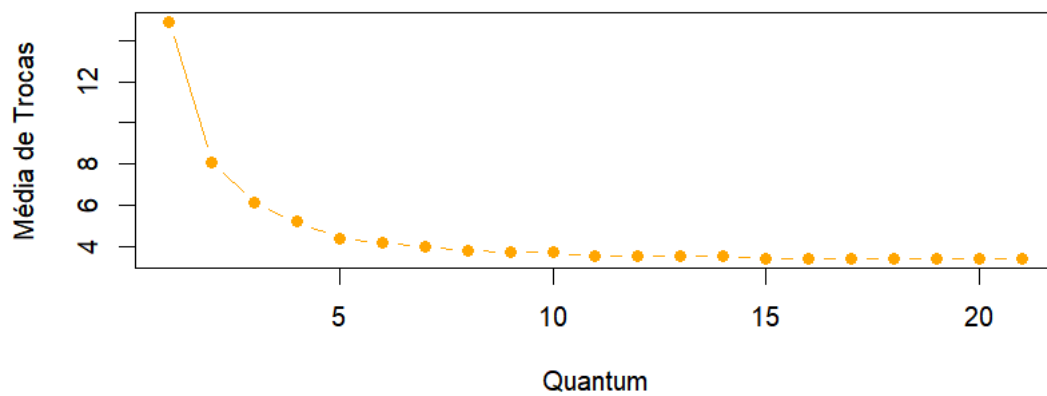
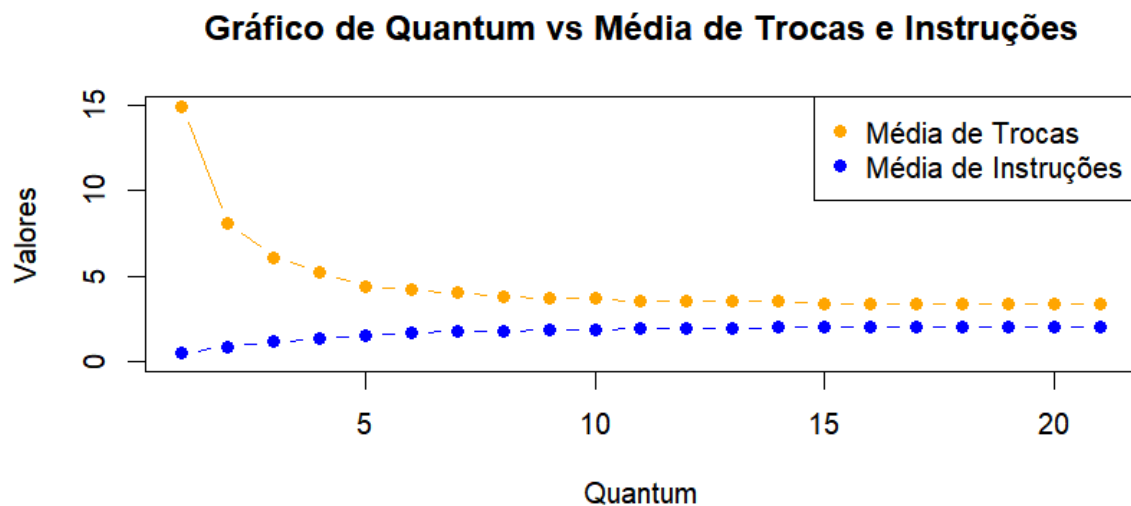


Gráfico de Quantum vs Média de Trocas





Análise

Do Gráfico de Quantum vs Média de Instruções, a quantidade de instruções aumenta conforme o valor do quantum aumenta, até se estabilizar em torno de quantum igual a 10. Com isso podemos afirmar que para valores altos de quantum, não haverá um aumento significativo na média de instruções executadas.

No Gráfico de Quantum vs Média de Trocas, podemos observar que a média de trocas entre os programas cai drasticamente conforme o valor do quantum aumenta, sendo que, a partir de quantum igual a 5, a média se estabiliza em um valor muito baixo (4 trocas ou menos).

Analisando as duas métricas em conjunto no Gráfico de Quantum vs Média de Trocas e Instruções, podemos notar que o melhor valor de n_{com} está mais próximo de **11**. Nesse ponto, a média de trocas de contexto é 3,5 e a média de instruções se aproxima de 2. Aumentar ainda mais o quantum provavelmente não traria melhorias significativas de desempenho.