

UNIVERSIDADE DE SÃO PAULO

ARTHUR YANG TUNG

FILIPPE VALERIANO BATISTA DE OLIVEIRA

GUSTAVO FERREIRA BOTELHO DE SENA

RENAN MOURA NASCIMENTO

EXERCÍCIO DE PROGRAMAÇÃO 1 – SISTEMAS OPERACIONAIS

SÃO PAULO

2024

ARTHUR YANG TUNG – 14559819 – T94

FILIPPE VALERIANO BATISTA DE OLIVEIRA – 14570701 – T94

GUSTAVO FERREIRA BOTELHO SENA – 13686643 – T94

RENAN MOURA NASCIMENTO – 14748921 – T94

EXERCÍCIO DE PROGRAMAÇÃO 1 – SISTEMAS OPERACIONAIS

Relatório do exercício de programação da disciplina ACH2044 - Sistemas Operacionais, da Escola de Artes Ciências e Humanidades da Universidade de São Paulo.

Professor: Norton Trevisan Roman

SÃO PAULO

2024

Sumário

INTRODUÇÃO	1
ORIENTAÇÕES SOBRE O CÓDIGO.....	2
ANÁLISE DOS VALORES DE QUANTUM.....	4
Analisando o Gráfico Número Médio de Trocas vs Quantum	4
Analisando o Gráfico Número Médio de Instruções Executadas vs Quantum	5
Analisando o Gráfico de Trade-off (Trocas vs Instruções Executadas)	6
Analisando o Gráfico de Variação Percentual	7
CONCLUSÃO.....	8

INTRODUÇÃO

Este relatório tem como objetivo documentar o desenvolvimento e os resultados obtidos no Exercício de Programação 1 da disciplina ACH2044 - Sistemas Operacionais, oferecida pela Escola de Artes, Ciências e Humanidades da Universidade de São Paulo. No âmbito desta atividade, foram explorados conceitos fundamentais de escalonamento de processos em sistemas operacionais, com foco na análise do comportamento de diferentes valores de quantum.

Através de simulações práticas, o trabalho busca identificar o impacto do tempo de quantum sobre a troca de processos e o número de instruções executadas, observando como essas variáveis se comportam à medida que o quantum é ajustado. Os resultados apresentados são discutidos com base em gráficos e tabelas, proporcionando uma análise quantitativa e qualitativa do desempenho do sistema.

ORIENTAÇÕES SOBRE O CÓDIGO

Descrição dos arquivos:

1. arquivos: Essa pasta contém arquivos de configuração e dados de entrada que o código utiliza.
2. código: Essa pasta contém os arquivos de código-fonte principais para o escalonamento dos processos.

`bcp.py`: Define a classe BCP (Bloco de Controle de Processos), que armazena informações sobre cada processo.

`escalonador.py`: Implementa o algoritmo de escalonamento que gerencia a execução dos processos de acordo com suas prioridades e quantum.

`fila_bloqueados.py`: Define e gerencia a fila de processos que estão bloqueados, aguardando para serem executados novamente.

`fila_prontos.py`: Define e gerencia a fila de processos prontos para execução, ordenando-os de acordo com suas prioridades.

`leitor_arquivos.py`: Responsável por ler os arquivos de entrada (como `prioridades.txt`, `quantum.txt` e os arquivos dos processos) e preparar os dados para o escalonador.

`main.py`: O arquivo principal do programa. Ele orquestra a execução do escalonador, carregando as configurações e processos, e gerencia a execução dos processos. Este é o arquivo que você deve rodar para executar o programa.

`tabela_processos.py`: Cria e gerencia uma tabela com todos os processos ativos, permitindo o escalonador acessar as informações de cada processo.

3. saídas: Essa pasta contém os arquivos de saída gerados após a execução do programa. Ela pode estar vazia inicialmente, mas após a execução, o programa criará arquivos de log aqui com informações sobre os processos.

Para executar o código do projeto, siga estes passos:

1. Certifique-se de que você tem o Python instalado na sua máquina.
2. Abra um terminal ou prompt de comando e navegue até o diretório `codigo/`.

3. Execute o seguinte comando para rodar o programa: `python main.py`

O script `main.py` irá carregar os arquivos de entrada da pasta `arquivos/`, executar o escalonador de processos, e os resultados serão gravados na pasta `saídas/` como arquivos de log.

	Média de Trocas	Média de Instruções
Quantum		
1	13.9	1.00
2	7.9	1.76
3	5.9	2.36
4	5.2	2.67
5	4.4	3.16
6	4.1	3.39
7	3.8	3.66
8	3.8	3.66
9	3.7	3.76
10	3.7	3.76

Tabela 1

Nessa tabela constam os valores de Quantum, média de trocas e média de instruções utilizados para as análises.

	Média de Trocas	Média de Instruções
count	21.000000	21.000000
mean	4.476190	3.520476
std	2.425882	0.871060
min	3.300000	1.000000
25%	3.400000	3.390000
50%	3.500000	3.970000
75%	4.100000	4.090000
max	13.900000	4.210000

Tabela 2

ANÁLISE DOS VALORES DE QUANTUM

NÚMERO MÉDIO DE TROCAS VS QUANTUM

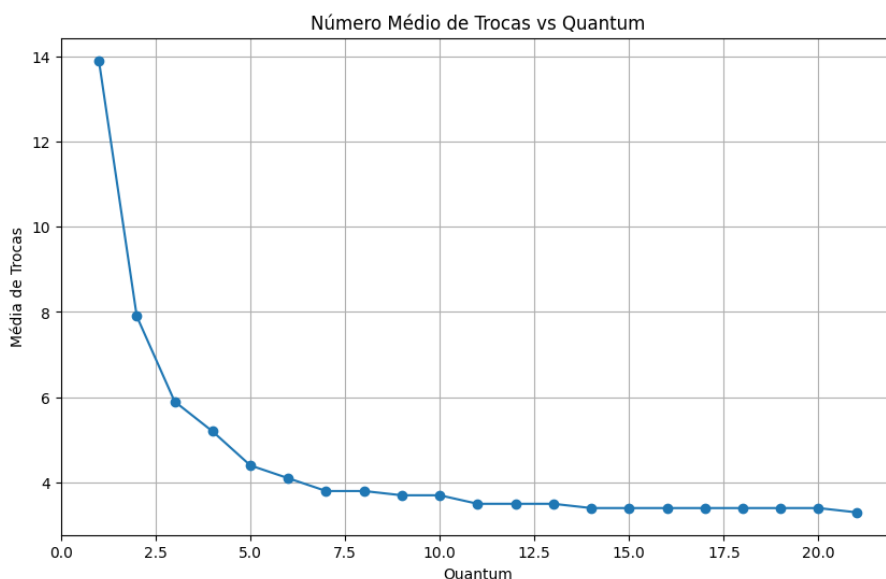


Gráfico 1

Analizando o Gráfico "Número Médio de Trocas vs Quantum"

O gráfico "Número Médio de Trocas vs Quantum" mostra que, à medida que o quantum aumenta, o número de trocas de processo diminui rapidamente nos valores iniciais, especialmente entre 1 e 5, indicando que quanta muito pequenos geram grande overhead no sistema devido às trocas frequentes. A partir de um quantum em torno de 7 a 8, o número de trocas se estabiliza, sugerindo que aumentos adicionais no quantum não trazem benefícios significativos em termos de redução de trocas. Portanto, um quantum de aproximadamente 8 parece ser o valor ideal para equilibrar a eficiência do sistema e minimizar o número de trocas de processo sem comprometer a responsividade.

NÚMERO MÉDIO DE INSTRUÇÕES EXECUTADAS VS QUANTUM

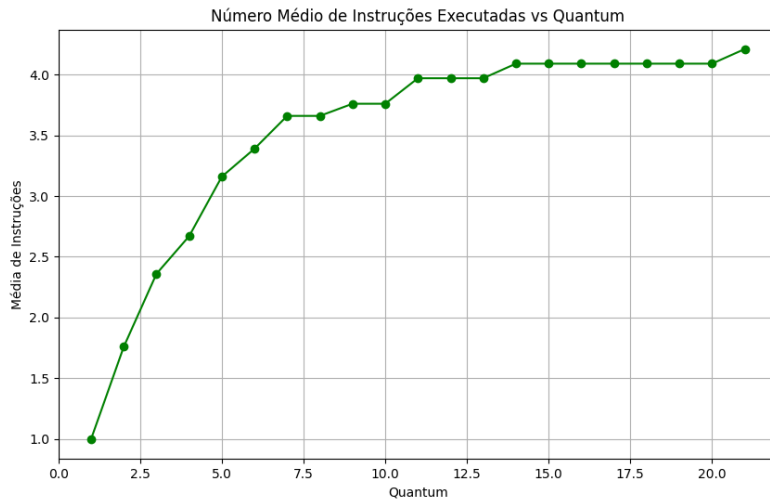


Gráfico 2

Analizando o Gráfico "Número Médio de Instruções Executadas vs Quantum"

O gráfico "Número Médio de Instruções Executadas vs Quantum" mostra que, com um quantum pequeno, o número de instruções executadas é baixo devido ao excesso de trocas de contexto. À medida que o quantum aumenta, a quantidade de instruções executadas cresce, refletindo maior eficiência, pois os processos têm mais tempo para rodar antes de serem interrompidos.

No entanto, após certo ponto (entre 12 e 15), o aumento do quantum não traz mais ganhos significativos, indicando que o sistema atinge um equilíbrio entre o tempo de execução dos processos e as trocas de contexto. O ideal é ajustar o quantum para maximizar as instruções executadas sem causar longos tempos de espera para os outros processos, otimizando assim o uso do processador.

TROCAS DE PROCESSO VC INSTRUÇÕES EXECUTADAS POR QUANTUM

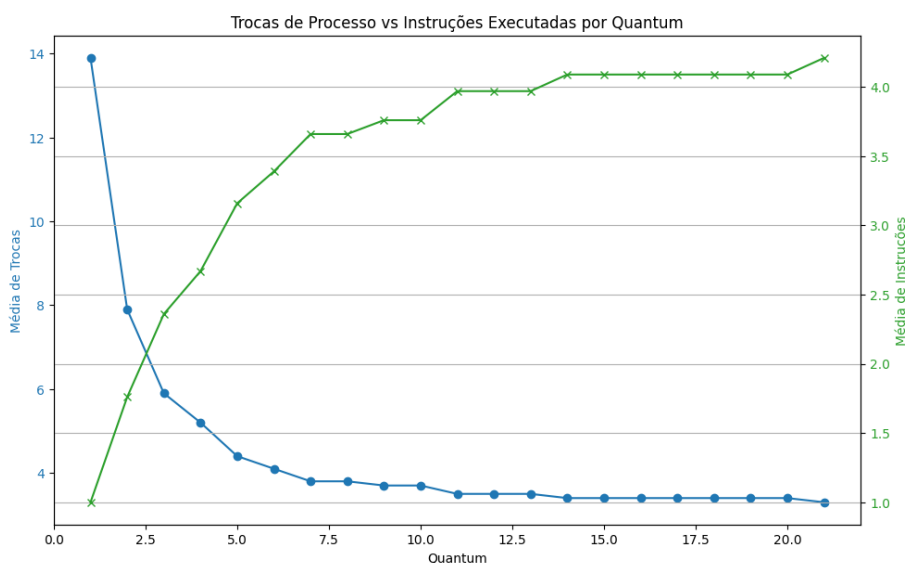


Gráfico 3

Analisando o Gráfico de Trade-off (Trocas vs Instruções Executadas)

O gráfico "Trocas de Processo vs Instruções Executadas por Quantum" demonstra o trade-off entre a redução das trocas de processo e o aumento das instruções executadas à medida que o quantum cresce.

Com um quantum pequeno, há muitas trocas de contexto, gerando alto overhead, mas mantendo a responsividade. À medida que o quantum aumenta, as trocas diminuem e o número de instruções executadas por quantum cresce. No entanto, um quantum muito grande reduz o número de trocas a um ponto que pode afetar a responsividade do sistema.

O ponto ideal está em equilibrar um número razoável de trocas com uma quantidade eficiente de instruções executadas, maximizando a eficiência do sistema sem comprometer sua responsividade.

	Variação de Trocas	Variação de Instruções
Quantum		
1	NaN	NaN
2	-43.165468	76.000000
3	-25.316456	34.090909
4	-11.864407	13.135593
5	-15.384615	18.352060
6	-6.818182	7.278481
7	-7.317073	7.964602
8	0.000000	0.000000
9	-2.631579	2.732240
10	0.000000	0.000000
11	-5.405405	5.585106
12	0.000000	0.000000
13	0.000000	0.000000
14	-2.857143	3.022670
15	0.000000	0.000000
16	0.000000	0.000000
17	0.000000	0.000000
18	0.000000	0.000000
19	0.000000	0.000000
20	0.000000	0.000000
21	-2.941176	2.933985

VARIAÇÃO PERCENTUAL DE TROCAS E INSTRUÇÕES POR QUANTUM

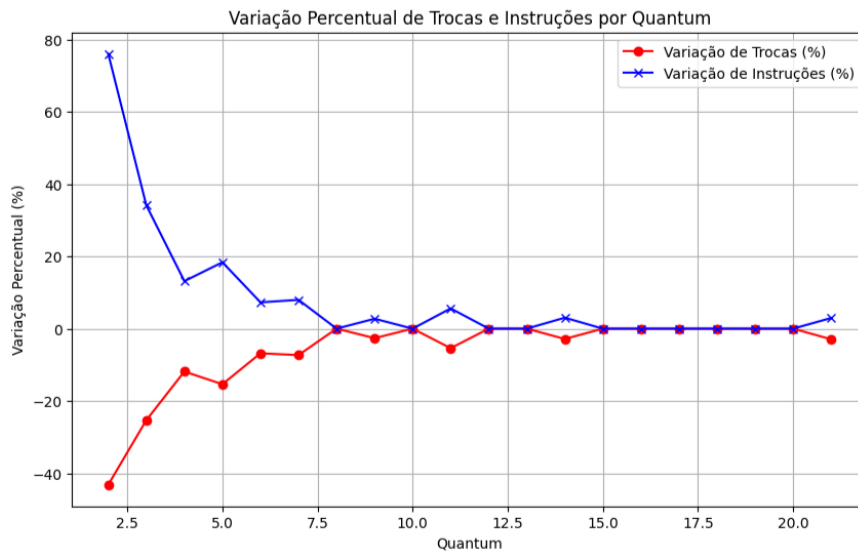


Gráfico 4

Analisando o Gráfico de Variação Percentual

O gráfico "Variação Percentual de Trocas e Instruções por Quantum" mostra quando aumentar o quantum deixa de trazer ganhos significativos. No início, a variação nas trocas e nas instruções é grande, mas conforme o quantum cresce, ambas as variações se aproximam de zero.

Isso indica que a partir de um certo ponto, aumentar o quantum não gera mais benefícios relevantes, já que a redução das trocas de contexto e o aumento nas instruções executadas se estabilizam. O ponto em que essas variações percentuais atingem zero representa um valor de quantum ideal, após o qual mais aumentos não trazem melhorias significativas na eficiência do sistema.

CONCLUSÃO

1. Trocas de Processos: Os gráficos mostram que o número de trocas de processo cai de forma rápida à medida que o quantum aumenta, especialmente nos valores de quantum pequenos (1 a 5). A partir de um quantum de aproximadamente 7 ou 8, o número de trocas se estabiliza, sugerindo que aumentos maiores no quantum não resultam em reduções significativas no número de trocas.
2. Eficiência de Instruções: O gráfico de instruções executadas por quantum mostra que o número de instruções executadas também aumenta conforme o quantum cresce, mas começa a se estabilizar em torno de quantum 10. Isso sugere que, a partir desse ponto, o processador já está sendo utilizado de forma eficiente, e aumentar o quantum traz pouco ganho adicional em termos de instruções executadas.
3. Equilíbrio entre Trocas e Instruções: Com base no gráfico de trade-off, podemos ver que há um ponto de equilíbrio em torno de quantum 7 a 10, onde o número de trocas é significativamente reduzido, e o número de instruções executadas por quantum já atingiu uma eficiência razoável.
4. Ponto Ótimo: A análise de variação percentual reforça que, por volta de quantum 8, a variação percentual tanto no número de trocas quanto nas instruções executadas começa a estabilizar, indicando que aumentos no quantum além desse ponto trazem poucos benefícios adicionais.

O valor mais adequado de quantum parece estar entre 7 e 10. Nesses valores, o número de trocas de processo é baixo, o número de instruções executadas por quantum é eficiente, e não há grandes variações nos resultados com o aumento do quantum. Para um sistema que busca balancear desempenho e responsividade, um quantum de 8 pode ser considerado o ideal.