

# Relatório de Trabalho

## Simulação de Escalonamento de Processos

Membros: Luigi Perotti Souza - 201910462

Professor: Benhur Stein

18 de janeiro de 2021

### Introdução

O seguinte relatório tem por finalidade descrever o desenvolvimento e resultados referentes ao primeiro Trabalho da Disciplina de Sistemas Operacionais. Para o desenvolvimento do programa foi utilizada a ferramenta IntelliJ IDEA, onde todo o código foi escrito na linguagem java com auxílio da ferramenta Json Simple disponível em: <https://code.google.com/archive/p/json-simple/> para codificação e decodificação de arquivos .json, utilizados na leitura de Jobs do trabalho.

### Metodologia

Para a execução do programa, todos os processos testados foram previamente carregados em memória, a partir de um arquivo .json com os seguintes parâmetros: 'program', onde será todas as instruções do processo, 'priority', que é a prioridade do processo durante o escalonamento, 'memory' será a quantidade de memória necessária durante o processo, 'IOSize' a quantidade de memória necessária para os campos de entrada e saída e por fim um parâmetro 'IO' que se trata de uma lista com os arquivos de E/S contendo um 'cost', que é o tempo de gravação necessária para a leitura ou escrita e a 'data', que é o valor da entrada ou saída. Para os testes foram utilizados os seguintes programas:

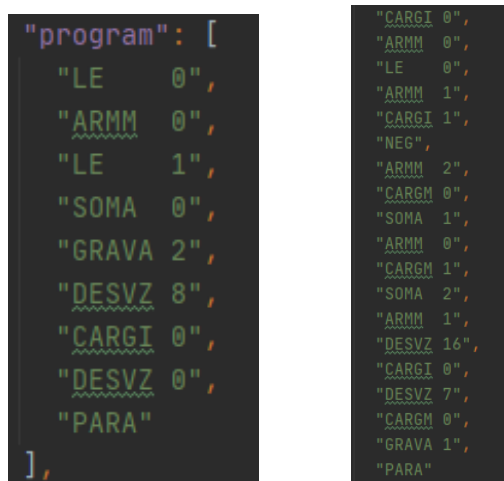


Figura 1: Processos utilizados

Sendo a primeira referente a apenas instruções limitadas por entrada e saída, enquanto a segunda terá instruções limitadas por CPU. Na primeira bateria de teste, foi executado um programa contendo quatro processos referentes ao programa limitado por entrada, onde cada um possuía os seguintes arquivos:

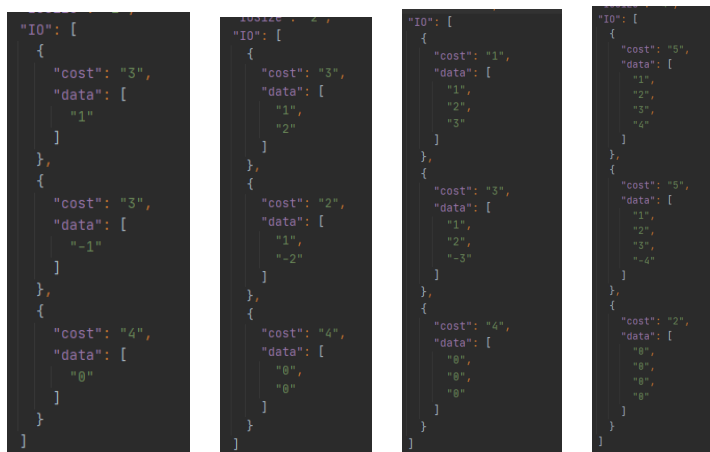
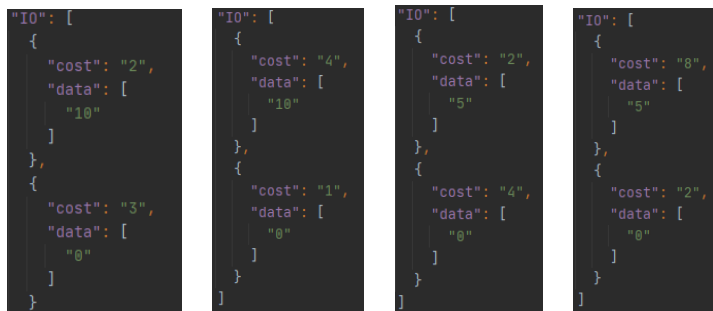


Figura 2: Arquivos de entrada e saída para os processos limitados a E/S

Mostrando que o primeiro programa possuía apenas uma entrada de arquivo, o segundo duas e assim até o quarto programa, variando o custo de

acesso a cada arquivo. Como neste exemplo havia poucas instruções no intervalo de cada chamada de sistema, foram testadas as configurações com Quantum de duas, três e cinco instruções, uma vez que com números elevados simplesmente o Quantum não acabaria e seria uma variável irrelevante para se calcular.

Para a segunda fase de teste, foram utilizados apenas o segundo exemplo, onde as instruções são limitadas pelo uso da CPU, então foram escolhidas as seguintes entradas:



The figure displays four JSON snippets, each representing a different process configuration. Each snippet is a JSON array with a single object element. The objects contain 'cost' and 'data' fields. The 'data' field is an array of strings. The snippets are as follows:

```

"IO": [
  {
    "cost": "2",
    "data": [
      "10"
    ]
  },
  {
    "cost": "3",
    "data": [
      "0"
    ]
  }
]

"IO": [
  {
    "cost": "4",
    "data": [
      "10"
    ]
  },
  {
    "cost": "1",
    "data": [
      "0"
    ]
  }
]

"IO": [
  {
    "cost": "2",
    "data": [
      "5"
    ]
  },
  {
    "cost": "4",
    "data": [
      "0"
    ]
  }
]

"IO": [
  {
    "cost": "8",
    "data": [
      "5"
    ]
  },
  {
    "cost": "2",
    "data": [
      "0"
    ]
  }
]

```

Figura 3: Arquivos de entrada e saída para os processos limitados a CPU

Como trata-se de um programa mais longo com poucas chamadas ao sistema operacional, a variação dos testes se deu ao valor N da entrada, onde quanto maior seu valor, mais laços serão executados. O custo apesar de variado, não é tão relevante, uma vez que os programas executarão a chamada de leitura e escrita uma vez apenas, sendo argumento importante nos testes a variação do quantum, onde foram testados com os valores de 5, 10 e 15 instruções.

No teste de processos variados, foram utilizados o segundo e quarto processo do exemplo na Figura 2, e o primeiro e terceiro referente ao uso de CPU da Figura 3, com testes variando de 2, 5, 10, e 15 instruções para o Quantum. Além dos exemplos descritos, foram testados os mesmos processos com uma prioridade pré estabelecida e estática para todo o andamento da execução.

# Resultados Obtidos

## Programa Limitado por E/S

Primeiramente foram testados os processos limitados por entrada e saída, conforme os arquivos de exemplo

### Prioridade Variada

Esse exemplo segue as entradas de exemplo na Figura 2, com todas as prioridades igualmente iniciadas no valor 0.5 e variando conforme a quantidade de Quantum utilizada.

- Para quantum de 2 Instruções:

Tempo total do Timer:	101
Tempo Ativo de CPU:	72
Tempo Ocioso:	29
Chamadas de Sistema:	34
Chamadas do tipo LE:	20
Chamadas do tipo GRAVA:	10
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	35
Número de Preempções:	5

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/1/2/3)
Hora de Término:	(23/49/78/101)
Tempo de Retorno:	(23/48/76/79)
Tempo de CPU:	(6/14/22/30)
Percentual de CPU:	(8%/19%/30%/41%)
Tempo Bloqueado:	(10/18/24/48)
Veze Bloqueado:	(4/7/10/13)
Veze Escalonado:	(3/8/12/12)
Número de vezes que a CPU foi perdida:	(0/1/2/2)

- Para quantum de 3 Instruções:

Tempo total do Timer:	110
Tempo Ativo de CPU:	72
Tempo Ocioso:	38
Chamadas de Sistema:	34
Chamadas do tipo LE:	20
Chamadas do tipo GRAVA:	10
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	28
Número de Preempções:	1

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/1/2/3)
Hora de Término:	(21/44/67/110)
Tempo de Retorno:	(21/43/65/107)
Tempo de CPU:	(6/14/22/30)
Percentual de CPU:	(8%/19%/30%/41%)
Tempo Bloqueado:	(10/18/24/48)
Vezes Bloqueado:	(4/7/10/13)
Vezes Escalonado:	(3/7/10/8)
Número de vezes que a CPU foi perdida:	(0/0/0/1)

- Para quantum de 5 Instruções:

Tempo total do Timer:	105
Tempo Ativo de CPU:	72
Tempo Ocioso:	33
Chamadas de Sistema:	34
Chamadas do tipo LE:	20
Chamadas do tipo GRAVA:	10
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	28
Número de Preempções:	0

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/1/2/3)
Hora de Término:	(21/45/68/105)
Tempo de Retorno:	(21/44/66/102)
Tempo de CPU:	(6/14/22/30)
Percentual de CPU:	(8%/19%/30%/41%)
Tempo Bloqueado:	(10/18/24/48)
Veze Bloqueado:	(4/7/10/13)
Veze Escalonado:	(3/7/10/8)
Número de veze que a CPU foi perdida:	(0/0/0/0)

### **Prioridade Fixa**

Esse exemplo segue as entradas de exemplo na Figura 2, com todas as prioridades iniciadas respectivamente em: 0.1, 0.56, 0.57 e 0.9, onde tais valores não mudam ao decorrer do programa. Valores próximos de zero representam prioridade mais alta.

- Para quantum de 2 Instruções:

Tempo total do Timer:	101
Tempo Ativo de CPU:	72
Tempo Ocioso:	33
Chamadas de Sistema:	34
Chamadas do tipo LE:	20
Chamadas do tipo GRAVA:	10
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	31
Número de Preempções:	4

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/1/2/3)
Hora de Término:	(21/46/69/105)
Tempo de Retorno:	(21/45/47/102)

Tempo de CPU:	(6/14/22/30)
Percentual de CPU:	(8%/19%/30%/41%)
Tempo Bloqueado:	(10/18/24/48)
Veze Bloqueado:	(3/8/11/9)
Veze Escalonado:	(3/8/12/12)
Número de veze que a CPU foi perdida:	(0/1/2/1)

- Para quantum de 3 Instruções:

Tempo total do Timer:	110
Tempo Ativo de CPU:	72
Tempo Ocioso:	38
Chamadas de Sistema:	34
Chamadas do tipo LE:	20
Chamadas do tipo GRAVA:	10
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	28
Número de Preempções:	1

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/1/2/3)
Hora de Término:	(21/44/67/110)
Tempo de Retorno:	(21/43/65/107)
Tempo de CPU:	(6/14/22/30)
Percentual de CPU:	(8%/19%/30%/41%)
Tempo Bloqueado:	(10/18/24/48)
Veze Bloqueado:	(4/7/10/13)
Veze Escalonado:	(3/7/10/8)
Número de veze que a CPU foi perdida:	(0/0/0/1)

- Para quantum de 5 Instruções:

Tempo total do Timer:	105
Tempo Ativo de CPU:	72
Tempo Ocioso:	33

Chamadas de Sistema:	34
Chamadas do tipo LE:	20
Chamadas do tipo GRAVA:	10
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	28
Número de Preempções:	0

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/1/2/3)
Hora de Término:	(21/45/68/105)
Tempo de Retorno:	(21/44/66/102)
Tempo de CPU:	(6/14/22/30)
Percentual de CPU:	(8%/19%/30%/41%)
Tempo Bloqueado:	(10/18/24/48)
Veze Bloqueado:	(4/7/10/13)
Veze Escalonado:	(3/7/10/8)
Número de vezes que a CPU foi perdida:	(0/0/0/0)

## Programa Limitado por CPU

Nessa seção foram testados os processos limitados por CPU, conforme os arquivos de exemplo

### Prioridade Variada

Esse exemplo segue as entradas de exemplo na Figura 3, com todas as prioridades igualmente iniciadas no valor 0.5 e variando conforme a quantidade de Quantum utilizada.

- Para quantum de 5 Instruções:

Tempo total do Timer:	303
Tempo Ativo de CPU:	298
Tempo Ocioso:	5
Chamadas de Sistema:	12



Chamadas do tipo LE:	4
Chamadas do tipo GRAVA:	4
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	54
Número de Preempções:	44

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/23/26)
Hora de Término:	(277/303/271/251)
Tempo de Retorno:	(277/300/248/225)
Tempo de CPU:	(97/97/52/52)
Percentual de CPU:	(32%/32%/17%/17%)
Tempo Bloqueado:	(5/5/6/10)
Veze Bloqueado:	(3/3/3/3)
Veze Escalonado:	(18/14/11/11)
Número de vezes que a CPU foi perdida:	(16/12/8/8)

- Para quantum de 10 Instruções:

Tempo total do Timer:	304
Tempo Ativo de CPU:	298
Tempo Ocioso:	6
Chamadas de Sistema:	12
Chamadas do tipo LE:	4
Chamadas do tipo GRAVA:	4
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	37
Número de Preempções:	26

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/57/60)
Hora de Término:	(246/304/302/257)

Tempo de Retorno:	(246/301/245/197)
Tempo de CPU:	(97/97/52/52)
Percentual de CPU:	(32%/32%/17%/17%)
Tempo Bloqueado:	(5/5/6/10)
Veze Bloqueado:	(3/3/3/3)
Veze Escalonado:	(11/12/7/7)
Número de veze que a CPU foi perdida:	(9/9/4/4)

- Para quantum de 15 Instruções:

Tempo total do Timer:	306
Tempo Ativo de CPU:	298
Tempo Ocioso:	8
Chamadas de Sistema:	12
Chamadas do tipo LE:	4
Chamadas do tipo GRAVA:	4
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	27
Número de Preempções:	17

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/85/103)
Hora de Término:	(187/299/306/268)
Tempo de Retorno:	(187/296/221/165)
Tempo de CPU:	(97/97/52/52)
Percentual de CPU:	(32%/32%/17%/17%)
Tempo Bloqueado:	(5/5/6/10)
Veze Bloqueado:	(3/3/3/3)
Veze Escalonado:	(8/8/5/6)
Número de veze que a CPU foi perdida:	(6/5/3/3)

## Prioridade Fixa

Esse exemplo segue as entradas de exemplo na Figura 3, com todas as prioridades iniciadas respectivamente em: 0.05, 0.5, 0.1 e 0.8, onde tais valores não mudam ao decorrer do programa. Valores próximos de zero representam prioridade mais alta.

- Para quantum de 5 Instruções:

Tempo total do Timer:	304
Tempo Ativo de CPU:	298
Tempo Ocioso:	6
Chamadas de Sistema:	12
Chamadas do tipo LE:	4
Chamadas do tipo GRAVA:	4
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	54
Número de Preempções:	46

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/23/26)
Hora de Término:	(190/120/3/196)
Tempo de Retorno:	(190/299/144/304)
Tempo de CPU:	(97/199/111/108)
Percentual de CPU:	(32%/32%/17%/17%)
Tempo Bloqueado:	(5/5/6/10)
Veze Bloqueado:	(3/3/3/3)
Veze Escalonado:	(18/17/11/10)
Número de vezes que a CPU foi perdida:	(16/14/8/8)

- Para quantum de 10 Instruções:

Tempo total do Timer:	303
Tempo Ativo de CPU:	298
Tempo Ocioso:	5
Chamadas de Sistema:	12

Chamadas do tipo LE:	4
Chamadas do tipo GRAVA:	4
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	37
Número de Preempções:	26

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/128/3/194)
Hora de Término:	(184/303/118/301)
Tempo de Retorno:	(184/175/115/107)
Tempo de CPU:	(97/97/52/52)
Percentual de CPU:	(32%/32%/17%/17%)
Tempo Bloqueado:	(5/5/6/10)
Veze Bloqueado:	(3/3/3/3)
Veze Escalonado:	(11/12/7/7)
Número de vezes que a CPU foi perdida:	(9/9/4/4)

- Para quantum de 15 Instruções:

Tempo total do Timer:	306
Tempo Ativo de CPU:	298
Tempo Ocioso:	8
Chamadas de Sistema:	12
Chamadas do tipo LE:	4
Chamadas do tipo GRAVA:	4
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	27
Número de Preempções:	17

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/150/3/153)
Hora de Término:	(157/304/134/286)

Tempo de Retorno:	(157/154/131/133)
Tempo de CPU:	(97/97/52/52)
Percentual de CPU:	(32%/32%/17%/17%)
Tempo Bloqueado:	(5/5/6/10)
Veze Bloqueado:	(3/3/3/3)
Veze Escalonado:	(7/7/6/6)
Número de veze que a CPU foi perdida:	(5/5/3/3)

## Programa com ambos tipos de Processo CPU

Nesta seção foram testados programas onde os processos carregados possuíam variações de ambos modelos apresentados. Conforme as Figuras 2 e 3, foram utilizados respectivamente os primeiro e terceiro exemplos e segundo e quarto processo do outro programa.

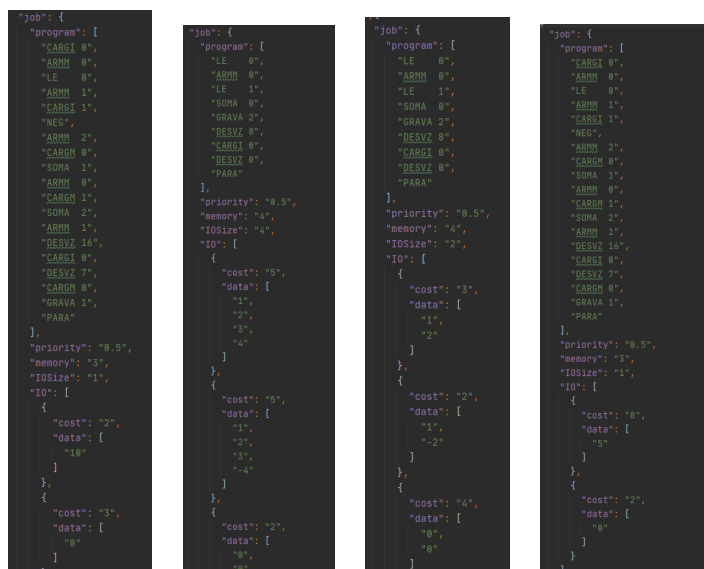


Figura 4: Processos ordenados (0/1/2/3) utilizados nesta sessão

## Prioridade Variada

Esse exemplo segue as entradas de exemplo na Figura 2 e 3, com todas as prioridades igualmente iniciadas no valor 0.5 e variando conforme a quantidade

de Quantum utilizada.

- Para quantum de 2 Instruções:

Tempo total do Timer:	200
Tempo Ativo de CPU:	193
Tempo Ocioso:	7
Chamadas de Sistema:	26
Chamadas do tipo LE:	14
Chamadas do tipo GRAVA:	8
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	95
Número de Preempções:	71

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/2/3/4)
Hora de Término:	(200/148/65/191)
Tempo de Retorno:	(200/146/62/187)
Tempo de CPU:	(97/30/14/52)
Percentual de CPU:	(50/15/7/26)
Tempo Bloqueado:	(5/48/18/10)
Veze Bloqueado:	(3/13/7/3)
Veze Escalonado:	(44/16/8/27)
Número de vezes que a CPU foi perdida:	(43/3/1/24)

- Para quantum de 5 Instruções:

Tempo total do Timer:	211
Tempo Ativo de CPU:	193
Tempo Ocioso:	18
Chamadas de Sistema:	26
Chamadas do tipo LE:	14
Chamadas do tipo GRAVA:	8
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	47

Número de Preempções: 24

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/4/5)
Hora de Término:	(169/211/162/193)
Tempo de Retorno:	(169/208/158/188)
Tempo de CPU:	(97/30/14/52)
Percentual de CPU:	(50/15/7/26)
Tempo Bloqueado:	(5/48/18/10)
Veze Bloqueado:	(3/13/7/3)
Veze Escalonado:	(18/11/7/11)
Número de vezes que a CPU foi perdida:	(16/0/0/8)

• Para quantum de 10 Instruções:

Tempo total do Timer:	220
Tempo Ativo de CPU:	193
Tempo Ocioso:	27
Chamadas de Sistema:	26
Chamadas do tipo LE:	14
Chamadas do tipo GRAVA:	8
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	32
Número de Preempções:	12

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/4/5)
Hora de Término:	(149/220/184/175)
Tempo de Retorno:	(149/217/180/170)
Tempo de CPU:	(97/30/14/52)
Percentual de CPU:	(50/15/7/26)
Tempo Bloqueado:	(5/48/18/10)
Veze Bloqueado:	(3/13/7/3)

Veze Escalonado:	(10/8/7/7)
Número de vezes que a CPU foi perdida:	(8/0/0/4)

- Para quantum de 15 Instruções:

Tempo total do Timer:	225
Tempo Ativo de CPU:	193
Tempo Ocioso:	32
Chamadas de Sistema:	26
Chamadas do tipo LE:	14
Chamadas do tipo GRAVA:	8
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	28
Número de Preempções:	9

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/4/5)
Hora de Término:	(147/225/183/179)
Tempo de Retorno:	(147/222/179/174)
Tempo de CPU:	(97/30/14/52)
Percentual de CPU:	(50/15/7/26)
Tempo Bloqueado:	(5/48/18/10)
Veze Bloqueado:	(3/13/7/3)
Veze Escalonado:	(8/7/7/6)
Número de vezes que a CPU foi perdida:	(6/0/0/3)

## Prioridade Fixa

Esse exemplo segue o mesmo exemplo das entradas, contudo, com todas as prioridades iniciadas respectivamente em: 0.01, 0.2, 0.5 e 0.6, onde tais valores não mudarão ao decorrer do programa. Valores próximos de zero representam prioridade mais alta.

- Para quantum de 2 Instruções:



Tempo total do Timer:	199
Tempo Ativo de CPU:	193
Tempo Ocioso:	6
Chamadas de Sistema:	26
Chamadas do tipo LE:	14
Chamadas do tipo GRAVA:	8
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	86
Número de Preempções:	62

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/2/4/5)
Hora de Término:	(183/102/93/199)
Tempo de Retorno:	(183/100/89/194)
Tempo de CPU:	(97/30/14/52)
Percentual de CPU:	(50/15/7/26)
Tempo Bloqueado:	(5/48/18/10)
Veze Bloqueado:	(3/13/7/3)
Veze Escalonado:	(42/16/8/20)
Número de vezes que a CPU foi perdida:	(40/3/1/18)

- Para quantum de 5 Instruções:

Tempo total do Timer:	199
Tempo Ativo de CPU:	193
Tempo Ocioso:	6
Chamadas de Sistema:	26
Chamadas do tipo LE:	14
Chamadas do tipo GRAVA:	8
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	41
Número de Preempções:	17

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/4/5)
Hora de Término:	(150/106/152/199)
Tempo de Retorno:	(150/103/148/194)
Tempo de CPU:	(97/30/14/52)
Percentual de CPU:	(50/15/7/26)
Tempo Bloqueado:	(5/48/18/10)
Veze Bloqueado:	(3/13/7/3)
Veze Escalonado:	(18/13/7/3)
Número de veze que a CPU foi perdida:	(16/0/0/1)

• Para quantum de 10 Instruções:

Tempo total do Timer:	198
Tempo Ativo de CPU:	193
Tempo Ocioso:	5
Chamadas de Sistema:	26
Chamadas do tipo LE:	14
Chamadas do tipo GRAVA:	8
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	37
Número de Preempções:	12

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/4/5)
Hora de Término:	(127/162/196/198)
Tempo de Retorno:	(127/159/192/193)
Tempo de CPU:	(97/30/14/52)
Percentual de CPU:	(50/15/7/26)
Tempo Bloqueado:	(5/48/18/10)
Veze Bloqueado:	(3/13/7/3)
Veze Escalonado:	(10/13/7/7)
Número de veze que a CPU foi perdida:	(8/0/0/4)

- Para quantum de 15 Instruções:

Tempo total do Timer:	199
Tempo Ativo de CPU:	193
Tempo Ocioso:	6
Chamadas de Sistema:	26
Chamadas do tipo LE:	14
Chamadas do tipo GRAVA:	8
Chamadas do tipo STOP:	4
Chamadas Ilegais	0
Trocas de Processo:	34
Número de Preempções:	9

*Descrição para cada processo (0/1/2/3):*

Hora de Início:	(0/3/4/5)
Hora de Término:	(137/199/196/197)
Tempo de Retorno:	(137/196/192/192)
Tempo de CPU:	(97/30/14/52)
Percentual de CPU:	(50/15/7/26)
Tempo Bloqueado:	(5/48/18/10)
Vezes Bloqueado:	(3/13/7/3)
Vezes Escalonado:	(8/13/7/6)
Número de vezes que a CPU foi perdida:	(6/0/0/3)

## Conclusão

Analizando os resultados obtidos, foi possível fazer uma análise referente a cada tipo de programa com seus respectivos processos. Iniciando pelo modelo restrito a E/S, foi possível destacar a eficiência em um Quantum de baixa quantidade nesses processos, uma vez que se trata de um programa com poucas instruções de CPU, eventualmente o mesmo bloqueia seus processos seguidamente com chamadas de sistema e muitas vezes, ao mesmo tempo. Diferença destaca no Quantum de 5 instruções, onde em nenhum momento o mesmo acabaria (pois não há intervalos iguais ou maiores de cinco instruções entre uma chamada de sistema e outra), resultando na CPU entrando em

modo sleep uma vez que houveram várias interrupções quase que simultaneamente, evento que com um quantum baixo aumenta a rotação entre os processos e assim, reduzindo o tempo ocioso da CPU. Como nesse caso a prioridade será recalculada e, eventualmente, os processos perderão a CPU por preempção e a prioridade se igualará durante o re cálculo. Por outro lado, os mesmos processos com a prioridade fixa tiveram um desempenho parecido entre si, resultado se dá pelo fato que o processo mais curto iniciou com uma prioridade maior, a fim de finalizá-lo rapidamente para liberar a CPU para os processos mais longos, contudo, no caso do Quantum mais baixo, os processos entraram em interrupção de forma conjunta como antes, porém com a prioridade fixa, os processos mais curtos foram escolhidos primeiro resultando em um maior tempo ocioso na espera dos processos mais longos se encerrarem, uma vez que bloqueado, os outros processos já foram terminados sem a possibilidade de um escalonamento. O Quantum mais alto teve o mesmo resultado.

Por outro lado, os processos limitados pela CPU receberam um impacto bem maior sob o Quantum que sobre o custo de acesso a arquivo. Como todos os processos só possuíam duas chamadas de sistema (uma de leitura quando inicia e uma de escrita quando acaba), então o principal fator que faria o escalonamento entre os processos é o Quantum, onde todos eventualmente perderam o processo por preempção. Nesse caso, de prioridade variável, foi desvantajoso o aumento do Quantum, uma vez que a prioridade de todos tenderiam ao mesmo valor, processos curtos competiriam CPU com processos mais longos, resultando assim que os mesmo demoraram mais até mesmo para serem chamados pela primeira vez pela CPU, aumentando o tempo ocioso e assim o tempo de execução do programa. Por outro lado, os processos com prioridade fix obtiveram um desempenho melhor em relação aqueles com prioridade dinâmica, uma vez que era possível selecionar os processos mais curtos para obter uma maior prioridade, possibilitando a liberação da CPU para os mais longos e trabalhosos, reduzindo o tempo ocioso para apenas o necessário aos comandos de leitura e escrita.

No caso dos processos com limitações variadas, o modelo com prioridade variável obteve um desempenho bem inferior conforme o aumento do quantum. Tal problema pode ser observado pelo fato que os processos limitados por E/S consomem diversas interrupções nas chamadas de leitura e escrita, problema semelhante a primeira análise, onde estes por não utilizarem totalmente de seu Quantum, obtém um prioridade maior e assim, chamam ainda mais interrupções. Diferença essa que se torna imperceptível

no mesmo exemplo com prioridade fixa, onde os processos de CPU que consomem mais Quantum, tem sua prioridade alternada entre si e os processos de E/S, fazendo assim que, durante uma interrupção de E/S, possa ser chamado e executado pela CPU, diminuindo o tempo ocioso da CPU. Tanto Quantum alto como baixo obtiveram quase o mesmo resultado em relação ao timer.

## Anexos

- Leitura de Json: <https://code.google.com/archive/p/json-simple/>
- Código fonte: [https://github.com/LuigiSouza/operating\\_system\\_simulator](https://github.com/LuigiSouza/operating_system_simulator)