Escola Politécnica da Universidade de São Paulo

PCS 3446 - Sistemas de Programação

Prof. João José Neto

PROJETO SEMESTRAL

Simulador de Sistema Operacional Dirigido por Eventos

Giovanni Abeni dos Santos
NUSP 9837121

12 de novembro de 2019

Sumário

Sumário **1**

1. INTRODUÇÃO **2**

2. TECNOLOGIA **3**

3. ESPECIFICAÇÕES DO SIMULADOR **4**

3.1. FUNCIONALIDADE 4

Entrada 4

Saída 4

Execução 4

3.2. ESTRUTURA DE ARQUIVOS 5

3.3. ARQUIVO DE CONFIGURAÇÕES 5

3.4. SOBRE O CÓDIGO 6

4. MÓDULOS **7**

4.1. Hardware - class Disc 7

4.2. Hardware - class Memory 7

4.3. Hardware - class Processor 8

4.4. OS - class Input 8

4.5. OS - class Segment 8

4.6. OS - class File 9

4.7. OS - class Job 9

4.8. OS - class Event 10

4.9. OS - class MultiprogrammingController 10

4.10. OS - class Scheduler 11

4.11. Classes Auxiliares 11

5. TESTES E VALIDAÇÕES **12**

5.1. GERAL 12

5.2. ENTRADA - input1.txt 13

5.3. SAÍDA 14

6. CONCLUSÃO E PRÓXIMOS PASSOS **20**

7. REFERÊNCIAS **20**

1. INTRODUÇÃO

Como projeto semestral da disciplina **Sistemas Operacionais**, foi solicitado o desenvolvimento de um simulador de SO, dividido em duas partes:

- **Parte 1**: construção de um simulador estocástico de processamento multiprogramado, dirigido por eventos
- **Parte 2**: modelagem de um sistema operacional com os seguintes requisitos:
    - Gerenciador de Memória
    - *Job Scheduler*
    - Gerenciador de Processos
    - Gerenciador de Informações

A **Parte 1** consiste em projetar um motor de eventos a fim de simular o funcionamento de um sistema operacional, integrando seus componentes (Memória, Processador, Disco, Leitoras e Impressoras). A teoria é baseada no artigo "*Computer System Simulation - an Introduction*", de Myron H. MacDougall.

Já a segunda parte é um refinamento da solução, a fim de considerar detalhes de cada *job* a ser processado.

2. TECNOLOGIA

Para implementação em código do simulador, foi escolhida uma pilha de desenvolvimento não muito usual para o objetivo: **NodeJS** com **TypeScript** (JavaScript fortemente tipado).

As motivações para escolha dessas tecnologias de altíssimo nível estão baseadas no objetivo futuro de aplicação do projeto:

- JavaScript está se tornando a **linguagem de programação mais difundida** no mercado;
- Uma orientação a objeto **fortemente tipada** facilita a abstração dos componentes físicos;
- São linguagens próprias para **aplicações Web**, o que facilitaria a futura distribuição do projeto;
- A distância do nível de máquina não atrapalha o entendimento ou interfere na abstração;
- Os projetos podem ser facilmente atualizados em distribuição através dos **Gerenciadores de Pacote** (ex: NPM);
- **Facilidade de instalação do ambiente**

OBSERVAÇÃO: Para rodar o projeto, é necessário ter o Node e o NPM instalados no computador. Para mais informações, acessar o repositório no GitHub.

3. ESPECIFICAÇÕES DO SIMULADOR

3.1. FUNCIONALIDADE

Entrada

O simulador recebe como entrada um arquivo .TXT contendo os detalhes dos *jobs* a serem executados. A estrutura do arquivo segue o seguinte protocolo:

```
; simulation start and end time
[INSTANTE_INICIAL (int ms)]
[INSTANTE_FINAL (int ms)]
[QUANTIDADE_DE_JOBS (int)]

; Id | Proc.Time | Qty.Segments |Seg[0]|...| I/O requests | Qty. Files | Files
[ID_JOB] [TEMPO_PROC.] [QTD_SEGMENTS] [SEGS…] [QTD_IO] [QTD_ARQS] [NOMES_ARQS…]
...
...
... (repetir para cara job)

; Jobs to be executed:
; Job Id | Arrival time
[ID_JOB] [INSTANTE_DE_CHEGADA]
...
... (repetir para cada job)

; obs: remember the blank white line after the programs.
```

Saída

Como resposta, o simulador retorna os logs de cada evento disparado, trazendo seu instante de execução, seu tipo, o job correspondente, e sua descrição. Quando é o caso, imprime-se também as alterações que o evento faz na memória.

Execução

Após a instalação do **npm** e das dependências do projeto (`npm install`), o projeto pode ser rodado com o comando:

```
$ npm start
```

O console pedirá o caminho para o arquivo de entrada e executará a simulação.

3.2. ESTRUTURA DE ARQUIVOS

O código do projeto foi dividido pelo tipo de abstração que cada módulo representa num sistema operacional real.

```
↳ src
    ↳ hardware
        ↳ Disc.ts
        ↳ Memory.ts
        ↳ Processor.ts
    ↳ os
        ↳ Event.ts
        ↳ File.ts
        ↳ Input.ts
        ↳ Job.ts
        ↳ MultiprogrammingController.ts
        ↳ Scheduler.ts
        ↳ Segment.ts
    ↳ aux
        ↳ LinkedList.ts
        ↳ Item.ts
    ↳ settings
        ↳ Settings.ts
```

3.3. ARQUIVO DE CONFIGURAÇÕES

Os parâmetros que são passíveis de alteração devido a preferência do usuário foram agregados em um arquivo de configuração, presente em ***/settings/Settings.ts*** :

src > settings > TS Settings.ts > ...

```ts
export class Settings {
  /* Memory */
  public static MEMORY_RELOCATING_TIME = 20;
  public static MEMORY_SIZE = 128;

  /* Disc */
  public static DISC_POSITIONING_TIME = 5;
  public static DISC_LATENCY_TIME = 5;
  public static DISC_TRANSFER_RATE = 40;

  /* Processor */
  public static PROCESSOR_QUANTUM = 50;

  /* Multiprogramming */
  public static MULTIPROGRAMMING_LIMIT = 4;

  /* Control */
  public static MEMORY_PRINT_SEGMENTS = true;
}
```

A função de cada constante é auto-explicativa.

3.4. SOBRE O CÓDIGO

Todo projeto foi feito utilizando técnicas de Orientação a Objeto e Estruturas Básicas de Algoritmos (Pilhas, Filas e Listas Ligadas). Usou o *tslinter* para garantir a padronização das regras de diagramação de código.

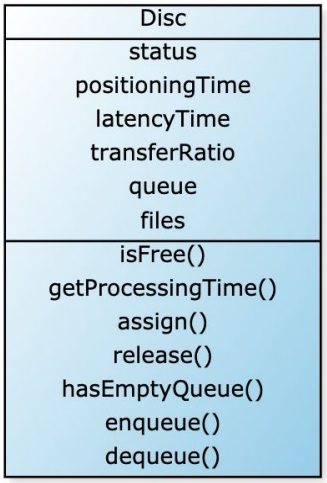
Obs. Optou-se por utilizar inglês nos códigos devido a maior padronização dos termos e possíveis futuros benefícios na manutenção.

4. MÓDULOS

O projeto foi separado em 13 classes, sendo 3 delas representando componentes do hardware de um computador, 7 delas representando as entidades de dados processadas pelo sistema operacional e outras 3 auxiliares.

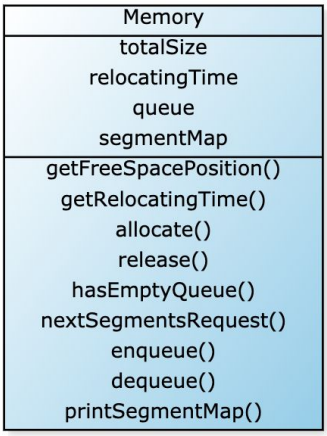
Abaixo segue uma explicação breve de cada classe.

4.1. Hardware - *class Disc*



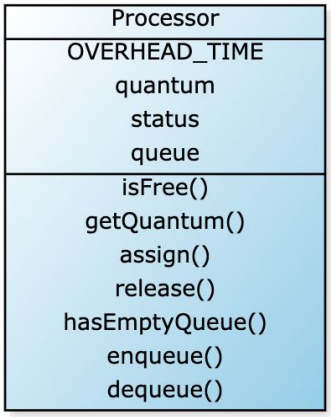
A classe Disc corresponde ao disco não volátil do computador. Ele possui as funções de controlar a fila de *jobs* e as requisições I/O no disco, além de guardar os arquivos, informações sobre seu status e tempos de latência/transferência.

4.2. Hardware - class Memory



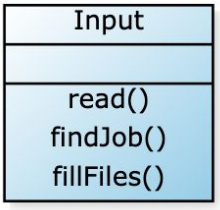
A classe Memory corresponde à memória volátil do computador. Sua função é alocar e desalocar segmentos de acordo com a fila de *jobs*.

4.3. Hardware - *class Processor*



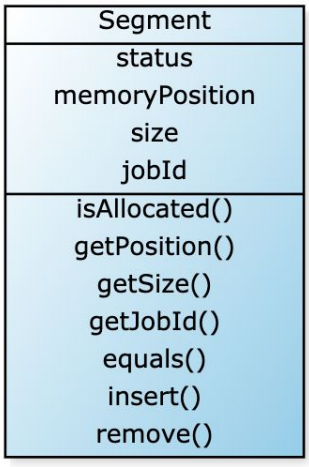
A classe Processor corresponde ao processador do computador. Sua função principal é fornecer informação acerca do quantum de tempo a ser fornecido ao *jobs*, bem como controlar a fila de execução.

4.4. OS - *class Input*



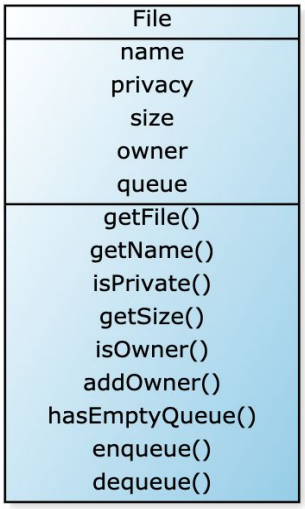
A classe Input é a que faz faz a leitura do arquivo de entrada, convertendo o TXT para as estruturas de dados utilizadas no simulador.

4.5. OS - *class Segment*



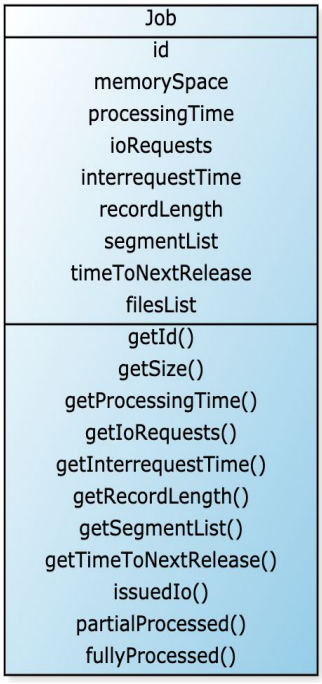
A classe Segment abstrai um segmento da memória solicitado por um *job*.

4.6. OS - *class File*



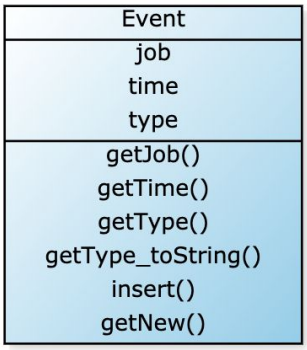
A classe File refere-se aos arquivos imagens de jobs que são salvos no disco (classe Disc). No geral, seus métodos tratam dos donos (jobs) do arquivo e de seu enfileramento.

4.7. OS - *class Job*



Sendo uma das classes mais importantes, a Job é a interface que guarda as informações do *jobs*, desde seu input até o momento após sua execução. A classe possui métodos para atualizar o tempo de processamento corrente do job (bem como checá-lo), consultar o tempo até o próximo release, e emitir interrupções de entrada e saída.

4.8. OS - *class Event*



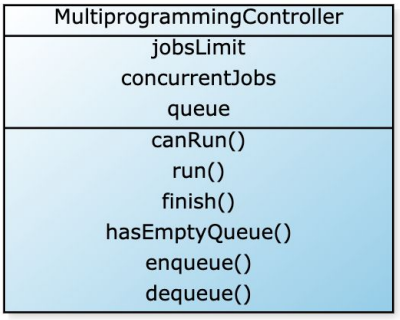
A classe Event é a entidade básica que guarda as informações de cada evento que dirige o simulador, bem como trata a inserção e o disparo de próximos eventos.

Os eventos são categorizados por dois enumeradores, um para seu tipo (numérico) e outro para sua descrição (textual):

```
export const EVENTS = {
  0: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  1: "Arrival                       ",
  2: "Request Memory                ",
  3: "Request Processor             ",
  4: "Release Processor (Issue I/O)",
  5: "Request I/O                   ",
  6: "Complete I/O                  ",
  7: "Completion                    ",
  8: "Time-out                      ",
};
```

```
export enum EventType {
  INVALID = 0,
  ARRIVAL = 1,
  REQUEST_MEMORY = 2,
  REQUEST_PROCESSOR = 3,
  ISSUE_IO = 4,
  REQUEST_IO = 5,
  RELEASE_IO = 6,
  COMPLETION = 7,
  TIME_OUT = 8,
}
```

4.9. OS - *class MultiprogrammingController*



A classe MultiprogrammingController tem a função de controlar a execução de jobs concomitantemente.

4.10. OS - *class Scheduler*

Scheduler
initialTime finalTime multiprogrammingController memory processor disc eventList jobTable timing filesList
readInputs() run()

Sendo a classe que guarda a lógica principal do simulador, o Scheduler é responsável por instanciar todos os módulos de hardware explicados acima, além de iniciar a leitura das entradas e acompanhar a fila de eventos, tratando-os para cada caso.

Seu funcionamento é totalmente baseado no modelo sugerido no artigo de MacDougall.

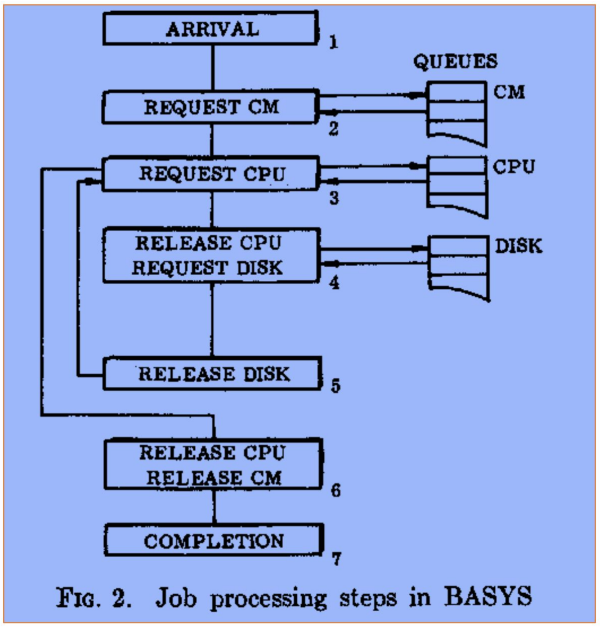


FIG. 2. Job processing steps in BASYS

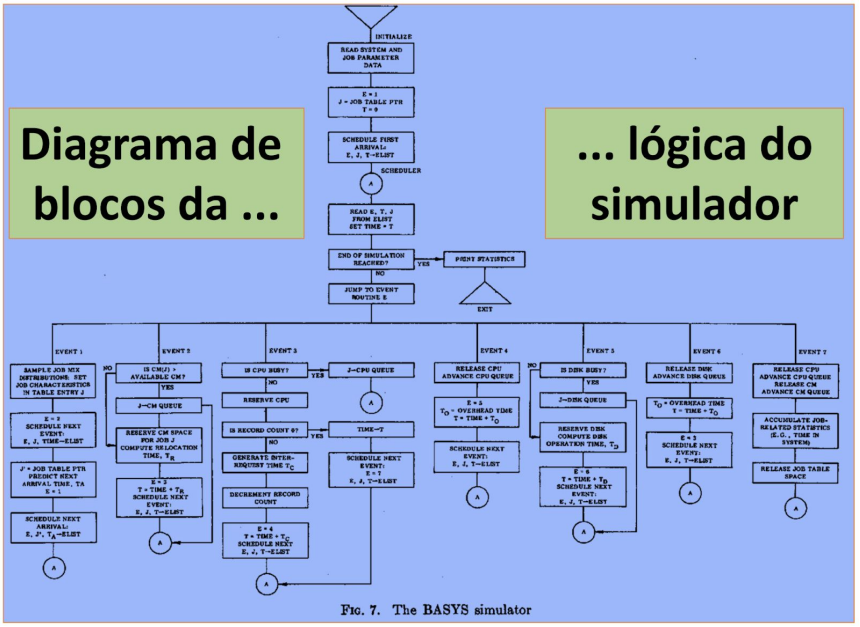


FIG. 7. The BASYS simulator

Diagramas de Exemplo - "*Computer System Simulation - an Introduction*", MH. McDougall.

4.11. Classes Auxiliares

Além das classes mostradas acimas, foram implementadas mais 3 classes de suporte:

- *class LinkedList<T>*: Lista ligada de elementos do tipo T
- *class Item<T>*: Nó de uma LinkedList<T>
- *class Settings*: Classe estática para guardar parâmetros editáveis

5. TESTES E VALIDAÇÕES

5.1. GERAL

Para se testar todas as funções do simulador, utilizou se uma entrada que pode abranger todos os casos de tratamento de eventos possíveis:

- Chegada de evento
- Requisição a Memória
    - Alocação
    - Enfileiramento
    - Release
- Requisição ao Processador
    - Atribuição
    - Enfileiramento
    - Release
- Requisição de Entrada e Saída
    - Atribuição
    - Enfileiramento
    - Release
- Finalização
- *Time out*

Sendo assim, o teste é composto de 4 *jobs* que chegam em tempos distintos ao sistema, contendo diferentes quantidades de segmentos, pedidos IO e arquivos. Os segmentos são dados tal que não é possível inserir todos os Jobs diretamente na memória, ocasionando seu enfileiramento. Os tempos de execução também são inseridos de forma a ocorrer timeouts.

Os parâmetros globais são os mesmos do código fornecido na seção 3.3.

5.2. ENTRADA - *input1.txt*

```
; simulation start and end time
10
100000
4

; Id | Proc.Time | Qty.Segments |Seg[0]|...| I/O requests | Qty. Files | Files
1 400 3 10 20 10 3 2 tea coffee
2 550 5 5 10 5 5 45 4 0
3 1200 1 50 15 3 salt pepper pad
4 650 2 30 15 0 1 minesweeper

; Jobs to be executed:
; Job Id | Arrival time
1 30
2 60
3 800
4 100

; obs: remember the blank white line after the programs.
```

5.3. SAÍDA

Como resultado da simulação, tem-se:

```
Digite o caminho para o arquivo de entrada: ./tests/input1.txt
Lendo o arquivo ./tests/input1.txt...


Id: 1   Processing time: 400    Number of segments: 3   I/O requests: 3         Number of files: 2
Files:  tea,  coffee,

Id: 2   Processing time: 550    Number of segments: 5   I/O requests: 4         Number of files: 0
Id: 3   Processing time: 1200   Number of segments: 1   I/O requests: 15        Number of files: 3
Files:  salt,  pepper,  pad,

Id: 4   Processing time: 650    Number of segments: 2   I/O requests: 0         Number of files: 1
Files:  minesweeper,



Job: 1  Arrival time: 30
Job: 2  Arrival time: 60
Job: 3  Arrival time: 800
Job: 4  Arrival time: 100



Execution:

 Time   Event   Job     Action                          Results
 10                                                     Starting...
  30    1       1       Arrival                         Job arrived at the system.
  60    1       2       Arrival                         Job arrived at the system.
  30    2       1       Request Memory                  Memory allocated to the job.

                         * MEMORY SEGMENT MAP *
                        ================================
                        | Position      Size    Job ID  |
                        --------------------------------
                        | 0             10      1       |
                        | 10            20      1       |
                        | 30            10      1       |
                        ================================

                        Memory Queue:

  800   1       3       Arrival                         Job arrived at the system.
  50    3       1       Request Processor               Processor assigned to the job.
  60    2       2       Request Memory                  Memory allocated to the job.

                         * MEMORY SEGMENT MAP *
                        ================================
                        | Position      Size    Job ID  |
                        --------------------------------
                        | 0             10      1       |
                        | 10            20      1       |
                        | 30            10      1       |
                        | 40            5       2       |
                        | 45            10      2       |
                        | 55            5       2       |
                        | 60            5       2       |
                        | 65            45      2       |
                        ================================

                        Memory Queue:

  100   1       4       Arrival                         Job arrived at the system.
  80    3       2       Request Processor               Job entered processor queue.
  800   2       3       Request Memory                  Job entered memory queue.
```

A simulação inicia com os dois primeiros *jobs* chegando. Em seguida o primeiro solicita a memória e o terceiro chega. Depois, o primeiro entra no processador e o segundo solicita a memória. Por fim, o ***job* 4** chega e o ***job* 2** entra na fila do processador enquanto o ***job* 3** entra na fila da memória

```
 * MEMORY SEGMENT MAP *
================================
| Position      Size    Job ID  |
--------------------------------
| 0             10      1       |
| 10            20      1       |
| 30            10      1       |
| 40            5       2       |
| 45            10      2       |
| 55            5       2       |
| 60            5       2       |
| 65            45      2       |
================================

Memory Queue:   <- Job 3 [50]

100   2       4       Request Memory                  Job entered memory queue.

 * MEMORY SEGMENT MAP *
================================
| Position      Size    Job ID  |
--------------------------------
| 0             10      1       |
| 10            20      1       |
| 30            10      1       |
| 40            5       2       |
| 45            10      2       |
| 55            5       2       |
| 60            5       2       |
| 65            45      2       |
================================

Memory Queue:   <- Job 3 [50] <- Job 4 [30,15]

100   8       1       Time-out                        Job released processor.
100   3       2       Request Processor               Processor assigned to the job.
100   3       1       Request Processor               Job entered processor queue.
150   8       2       Time-out                        Job released processor.
150   3       1       Request Processor               Processor assigned to the job.
150   3       2       Request Processor               Job entered processor queue.
200   4       1       Release Processor (Issue I/O)   Job released the processor and issued the disc.
200   5       1       Request I/O                     Disc assigned to the job.
200   3       2       Request Processor               Processor assigned to the job.
4210  6       1       Complete I/O                    Job released the disc.
250   8       2       Time-out                        Job released processor.
4210  3       1       Request Processor               Processor assigned to the job.
250   3       2       Request Processor               Job entered processor queue.
4260  8       1       Time-out                        Job released processor.
4260  3       2       Request Processor               Processor assigned to the job.
4260  3       1       Request Processor               Job entered processor queue.
4270  4       2       Release Processor (Issue I/O)   Job released the processor and issued the disc.
4270  5       2       Request I/O                     Disc assigned to the job.
4270  3       1       Request Processor               Processor assigned to the job.
8280  6       2       Complete I/O                    Job released the disc.
4320  4       1       Release Processor (Issue I/O)   Job released the processor and issued the disc.
8280  3       2       Request Processor               Processor assigned to the job.
4320  5       1       Request I/O                     Disc assigned to the job.
8330  8       2       Time-out                        Job released processor.
8330  6       1       Complete I/O                    Job released the disc.
8330  3       2       Request Processor               Processor assigned to the job.
8330  3       1       Request Processor               Job entered processor queue.
8380  8       2       Time-out                        Job released processor.
8380  3       1       Request Processor               Processor assigned to the job.
8380  3       2       Request Processor               Job entered processor queue.
8430  8       1       Time-out                        Job released processor.
8430  3       2       Request Processor               Processor assigned to the job.
8430  3       1       Request Processor               Job entered processor queue.
```

Nesse segundo trecho, os jobs 1 e 2 revezam o processador, e sofrem interrupções de IO.

```
8430  3       2       Request Processor               Processor assigned to the job.
8430  3       1       Request Processor               Job entered processor queue.
8440  4       2       Release Processor (Issue I/O)   Job released the processor and issued the disc.
8440  5       2       Request I/O                     Disc assigned to the job.
8440  3       1       Request Processor               Processor assigned to the job.
12450 6       2       Complete I/O                    Job released the disc.
8490  4       1       Release Processor (Issue I/O)   Job released the processor and issued the disc.
12450 3       2       Request Processor               Processor assigned to the job.
8490  5       1       Request I/O                     Disc assigned to the job.
12500 8       2       Time-out                        Job released processor.
12500 6       1       Complete I/O                    Job released the disc.
12500 3       2       Request Processor               Processor assigned to the job.
12500 3       1       Request Processor               Job entered processor queue.
12550 8       2       Time-out                        Job released processor.
12550 3       1       Request Processor               Processor assigned to the job.
12550 3       2       Request Processor               Job entered processor queue.
12600 8       1       Time-out                        Job released processor.
12600 3       2       Request Processor               Processor assigned to the job.
12600 3       1       Request Processor               Job entered processor queue.
12610 4       2       Release Processor (Issue I/O)   Job released the processor and issued the disc.
12610 5       2       Request I/O                     Disc assigned to the job.
12610 3       1       Request Processor               Processor assigned to the job.
16620 6       2       Complete I/O                    Job released the disc.
12660 7       1       Completion                      Job released processor and memory.

 * MEMORY SEGMENT MAP *
================================
| Position      Size    Job ID  |
--------------------------------
| 40            5       2       |
| 45            10      2       |
| 55            5       2       |
| 60            5       2       |
| 65            45      2       |
================================

Memory Queue:   <- Job 3 [50] <- Job 4 [30,15]

16620 3       2       Request Processor               Processor assigned to the job.
16670 8       2       Time-out                        Job released processor.
16670 3       2       Request Processor               Processor assigned to the job.
16720 8       2       Time-out                        Job released processor.
16720 3       2       Request Processor               Processor assigned to the job.
16730 4       2       Release Processor (Issue I/O)   Job released the processor and issued the disc.
16730 5       2       Request I/O                     Disc assigned to the job.
20740 6       2       Complete I/O                    Job released the disc.
20740 3       2       Request Processor               Processor assigned to the job.
20790 8       2       Time-out                        Job released processor.
20790 3       2       Request Processor               Processor assigned to the job.
20840 8       2       Time-out                        Job released processor.
20840 3       2       Request Processor               Processor assigned to the job.
20850 7       2       Completion                      Job released processor and memory.

 * MEMORY SEGMENT MAP *
================================
| Position      Size    Job ID  |
--------------------------------
| 45            10      2       |
| 55            5       2       |
| 60            5       2       |
| 65            45      2       |
================================

Memory Queue:   <- Job 3 [50] <- Job 4 [30,15]

20850 2       3       Request Memory                  Memory allocated to the job.
```

Nessa terceira parte, os ***jobs 1*** e ***2*** são completados, liberando memória para a alocação do ***job 3***.

```
 * MEMORY SEGMENT MAP *
================================
| Position      Size    Job ID  |
--------------------------------
| 0             50      3       |
================================

Memory Queue:   <- Job 4 [30,15]
```

```
20870 3       3       Request Processor               Processor assigned to the job.
20920 8       3       Time-out                        Job released processor.
20920 3       3       Request Processor               Processor assigned to the job.
20945 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
20945 5       3       Request I/O                     Disc assigned to the job.
24955 6       3       Complete I/O                    Job released the disc.
24955 3       3       Request Processor               Processor assigned to the job.
25005 8       3       Time-out                        Job released processor.
25005 3       3       Request Processor               Processor assigned to the job.
25030 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
25030 5       3       Request I/O                     Disc assigned to the job.
29040 6       3       Complete I/O                    Job released the disc.
29040 3       3       Request Processor               Processor assigned to the job.
29090 8       3       Time-out                        Job released processor.
29090 3       3       Request Processor               Processor assigned to the job.
29115 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
29115 5       3       Request I/O                     Disc assigned to the job.
33125 6       3       Complete I/O                    Job released the disc.
33125 3       3       Request Processor               Processor assigned to the job.
33175 8       3       Time-out                        Job released processor.
33175 3       3       Request Processor               Processor assigned to the job.
33200 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
33200 5       3       Request I/O                     Disc assigned to the job.
37210 6       3       Complete I/O                    Job released the disc.
37210 3       3       Request Processor               Processor assigned to the job.
37260 8       3       Time-out                        Job released processor.
37260 3       3       Request Processor               Processor assigned to the job.
37285 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
37285 5       3       Request I/O                     Disc assigned to the job.
41295 6       3       Complete I/O                    Job released the disc.
41295 3       3       Request Processor               Processor assigned to the job.
41345 8       3       Time-out                        Job released processor.
41345 3       3       Request Processor               Processor assigned to the job.
41370 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
41370 5       3       Request I/O                     Disc assigned to the job.
45380 6       3       Complete I/O                    Job released the disc.
45380 3       3       Request Processor               Processor assigned to the job.
45430 8       3       Time-out                        Job released processor.
45430 3       3       Request Processor               Processor assigned to the job.
45455 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
45455 5       3       Request I/O                     Disc assigned to the job.
49465 6       3       Complete I/O                    Job released the disc.
49465 3       3       Request Processor               Processor assigned to the job.
49515 8       3       Time-out                        Job released processor.
49515 3       3       Request Processor               Processor assigned to the job.
49540 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
49540 5       3       Request I/O                     Disc assigned to the job.
53550 6       3       Complete I/O                    Job released the disc.
53550 3       3       Request Processor               Processor assigned to the job.
53600 8       3       Time-out                        Job released processor.
53600 3       3       Request Processor               Processor assigned to the job.
53625 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
53625 5       3       Request I/O                     Disc assigned to the job.
57635 6       3       Complete I/O                    Job released the disc.
57635 3       3       Request Processor               Processor assigned to the job.
57685 8       3       Time-out                        Job released processor.
57685 3       3       Request Processor               Processor assigned to the job.
57710 4       3       Release Processor (Issue I/O)   Job released the processor and issued the disc.
```

```
57710 5       3       Request I/O                   Disc assigned to the job.
61720 6       3       Complete I/O                  Job released the disc.
61720 3       3       Request Processor             Processor assigned to the job.
61770 8       3       Time-out                      Job released processor.
61770 3       3       Request Processor             Processor assigned to the job.
61795 4       3       Release Processor (Issue I/O) Job released the processor and issued the disc.
61795 5       3       Request I/O                   Disc assigned to the job.
65805 6       3       Complete I/O                  Job released the disc.
65805 3       3       Request Processor             Processor assigned to the job.
65855 8       3       Time-out                      Job released processor.
65855 3       3       Request Processor             Processor assigned to the job.
65880 4       3       Release Processor (Issue I/O) Job released the processor and issued the disc.
65880 5       3       Request I/O                   Disc assigned to the job.
69890 6       3       Complete I/O                  Job released the disc.
69890 3       3       Request Processor             Processor assigned to the job.
69940 8       3       Time-out                      Job released processor.
69940 3       3       Request Processor             Processor assigned to the job.
69965 4       3       Release Processor (Issue I/O) Job released the processor and issued the disc.
69965 5       3       Request I/O                   Disc assigned to the job.
73975 6       3       Complete I/O                  Job released the disc.
73975 3       3       Request Processor             Processor assigned to the job.
74025 8       3       Time-out                      Job released processor.
74025 3       3       Request Processor             Processor assigned to the job.
74050 4       3       Release Processor (Issue I/O) Job released the processor and issued the disc.
74050 5       3       Request I/O                   Disc assigned to the job.
78060 6       3       Complete I/O                  Job released the disc.
78060 3       3       Request Processor             Processor assigned to the job.
78110 8       3       Time-out                      Job released processor.
78110 3       3       Request Processor             Processor assigned to the job.
78135 4       3       Release Processor (Issue I/O) Job released the processor and issued the disc.
78135 5       3       Request I/O                   Disc assigned to the job.
82145 6       3       Complete I/O                  Job released the disc.
82145 3       3       Request Processor             Processor assigned to the job.
82195 8       3       Time-out                      Job released processor.
82195 3       3       Request Processor             Processor assigned to the job.
82220 7       3       Completion                    Job released processor and memory.

                       * MEMORY SEGMENT MAP *
                      ================================
                      | Position      Size    Job ID  |
                      --------------------------------
                      ================================

                      Memory Queue:   <- Job 4 [30,15]

82220 2       4       Request Memory                Memory allocated to the job.

                       * MEMORY SEGMENT MAP *
                      ================================
                      | Position      Size    Job ID  |
                      --------------------------------
                      | 30            15      4       |
                      | 0             30      4       |
                      | 30            15      4       |
                      ================================

                      Memory Queue:

82240 3       4       Request Processor             Processor assigned to the job.
82290 8       4       Time-out                      Job released processor.
82290 3       4       Request Processor             Processor assigned to the job.
82340 8       4       Time-out                      Job released processor.
82340 3       4       Request Processor             Processor assigned to the job.
82390 8       4       Time-out                      Job released processor.
82390 3       4       Request Processor             Processor assigned to the job.
82440 8       4       Time-out                      Job released processor.
82440 3       4       Request Processor             Processor assigned to the job.
```

Nas duas páginas anteriores, ocorre o processamento e o tratamento das entras e saídas do ***job 3***, bem como sua completude. Em seguida, a memória é liberada para alocação do ***job 4***.

```
82490 8       4       Time-out                        Job released processor.
82490 3       4       Request Processor               Processor assigned to the job.
82540 8       4       Time-out                        Job released processor.
82540 3       4       Request Processor               Processor assigned to the job.
82590 8       4       Time-out                        Job released processor.
82590 3       4       Request Processor               Processor assigned to the job.
82640 8       4       Time-out                        Job released processor.
82640 3       4       Request Processor               Processor assigned to the job.
82690 8       4       Time-out                        Job released processor.
82690 3       4       Request Processor               Processor assigned to the job.
82740 8       4       Time-out                        Job released processor.
82740 3       4       Request Processor               Processor assigned to the job.
82790 8       4       Time-out                        Job released processor.
82790 3       4       Request Processor               Processor assigned to the job.
82840 8       4       Time-out                        Job released processor.
82840 3       4       Request Processor               Processor assigned to the job.
82890 7       4       Completion                      Job released processor and memory.

                      * MEMORY SEGMENT MAP *
                      ================================
                      | Position      Size    Job ID  |
                      --------------------------------
                      ================================

                      Memory Queue:

82890                                                 No more jobs to simulate.


   *** JOB SUMMARY ***

Job ID  Arrival Time    End Time        Processor Period        T       W
1       30              12660           200                     12630   63.15
2       60              20850           500                     20790   41.58
3       800             82220           800                     81420   101.78
4       100             82890           600                     82790   137.98

 Tavg = 39526    Wavg = 68.90


***** END OF SIMULATION ****
```

Finalmente, o ***job 3*** é completado e o simulador calcula os resultados da simulação. A tabela **JOB SUMMARY** traz os tempos de cada *job*, bem como os valores do ***turnaround time* médio** (`Tavg = 39525 ms`) e a taxa média de ***turnaround time* ponderado** (`Wavg = 68.90`).

6. CONCLUSÃO E PRÓXIMOS PASSOS

O resultado do projeto foi satisfatório dado que o objetivo de organizar e simular os eventos a partir do jobs de entrada foi cumprido. Apesar do bom resultado, é necessário realizar mais testes com casos diversos, a fim de garantir que o tratamento dos eventos está correto.

Ainda assim, de forma geral, o projeto já pode ser considerado suficiente como objeto de estudo e validação.

7. REFERÊNCIAS

1. Myron H. MacDougall (1987). *Simulating Computer Systems: Techniques and Tools*. MIT Press.
2. Aula 09 a 13 - PCS3446 - Prof. Dr. João José Neto: Administração de Processos
3. Implementação em Java do mesmo desafio, encontrada no GitHub.