

Henry Rodrigues da Silva

Jacyara Bosse

Mathias Olivio Reolon

René Nolio Santa Cruz

Projeto prático de simulação de sistemas (X2)

Projeto Prático 07

Gerenciamento de memória virtual

Florianópolis, junho de 2019

# Sumário

<b>Introdução</b>	<b>5</b>
Organização do trabalho	5
<b>Formulação e análise do problema</b>	<b>6</b>
Objetivos	6
Hipóteses	6
<b>Planejamento do projeto</b>	<b>7</b>
<b>Formulação do modelo conceitual</b>	<b>7</b>
<b>Coleta de dados e macro-informações</b>	<b>8</b>
Tratamento de dados coletados	8
<b>Tradução do modelo</b>	<b>10</b>
<b>Verificação e validação</b>	<b>16</b>
<b>Projeto experimental final</b>	<b>16</b>
<b>Experimentação</b>	<b>17</b>
Resultados obtidos:	18
Hipótese 1	18
Hipótese 2	19
<b>Interpretação e análise dos resultados</b>	<b>22</b>
Replicações	22
<b>Comparação e identificação das melhores soluções</b>	<b>22</b>
<b>Documentação</b>	<b>24</b>
<b>Apresentação dos resultados e implementação</b>	<b>24</b>
<b>Referências bibliográficas</b>	<b>25</b>

## Lista de figuras

Figura 1: Etapas referentes a um estudo de modelagem e simulação (FREITAS, 2008)	5
Figura 2: Esboço geral do sistema	8
Figura 3: Distribuição utilizada: Distribuição triangular	9
Figura 4: Variáveis do modelo de simulação no Arena	10
Figura 5: Estatísticas coletadas pelo Arena	11
Figura 6: Interação entre os componentes do sistema	12
Figura 7: Acesso a memória modelado no Arena	13
Figura 8: Primeira parte do algoritmo de substituição de páginas	14
Figura 9: Segunda parte do algoritmo de substituição de páginas	14
Figura 10: Loop do processo	15
Figura 11: Resultados obtidos através do output analyzer	18
Figura 12: Resultados obtidos através do output analyzer	19

## Lista de Abreviaturas e Siglas

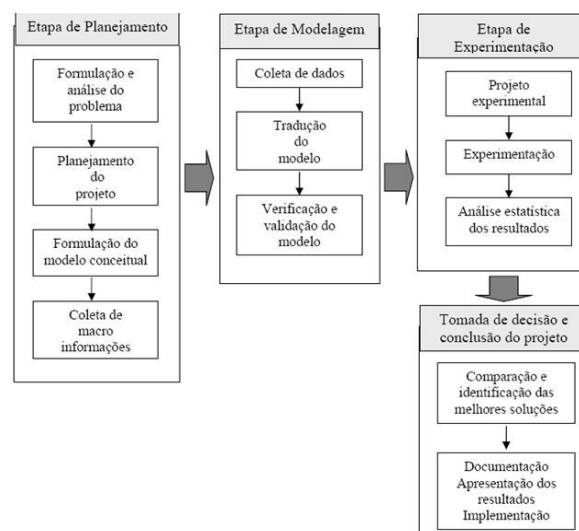
TLB	<i>Translation lookaside buffer</i> , é um cache de memória que armazena traduções recentes de memória virtual em endereços físicos.
RAM	<i>Random Access Memory</i> , ou memória de acesso randômico
LRU	<i>Least Recently Used</i> , é um algoritmo de substituição de páginas que procura substituir a página menos recentemente utilizada
LFU	<i>Least Frequently Used</i> , é um algoritmo de substituição de páginas que procura substituir a página menos frequentemente utilizada
Disco	Disco rígido ou HD, é a memória de massa

# 1. Introdução

Este trabalho consiste na modelagem e simulação de um sistema de gerenciamento de memória virtual por paginação em um nível. O objetivo principal é avaliar o impacto do algoritmo de substituição de páginas, do tamanho da página e da velocidade do disco em relação ao tempo médio de acesso à memória. O sistema inclui uma TLB, tabela de páginas, páginas na memória RAM e disco rígido. Os algoritmos de substituição de páginas avaliados são o LRU (Least Recently Used) e o LFU (Least Frequently Used); ambos utilizam política de substituição de páginas local. São testados discos lentos e rápidos, e as páginas podem variar de 4 KB a 16 KB. Temos 2 processos executando, que geram uma referência à memória algum tempo após sua última referência ter sido atendida, e enquanto isso, eles permanecem bloqueados. Os tempos entre acessos foram fornecidos no enunciado.

## Organização do trabalho

As etapas de projeto de simulação adotadas neste trabalho são as etapas definidas no livro de Freitas Filho (Freitas Filho, 2008) conforme na figura 1, e estão descritas nas próximas seções.



**Figura 1:** Etapas referentes a um estudo de modelagem e simulação (FREITAS, 2008)

## 2. Formulação e análise do problema

De acordo com a proposta descrita acima, o problema que está sendo estudado é a simulação de um sistema de referências à memória com algoritmos diferentes de substituição de página, que são eles:

- LRU (Least Recently Used): O LRU é um algoritmo de substituição de página que apresenta um bom desempenho substituindo a página menos recentemente usada. Esta política foi definida baseada na seguinte observação: se a página está sendo intensamente referenciada pelas instruções é muito provável que ela seja novamente referenciada pelas instruções seguintes e, de modo oposto, aquelas que não foram acessadas nas últimas instruções também é provável que não sejam acessadas nas próximas (CASSETTARI, 2013).
- LFU (Least Frequently Used): O LFU (Least Frequently Used) escolhe a página que foi menos acessada dentre todas as que estão carregadas em memória. Para isso, é mantido um contador de acessos associado a cada página (hit) para que se possa realizar esta verificação. Esta informação é zerada cada vez que a página deixa a memória. (CASSETTARI, 2013).

Essa modelagem pretende alcançar os seguintes objetivos e testar as seguintes hipóteses:

### Objetivos

- Avaliar os seguintes impactos:
  - Algoritmo de substituição de páginas
  - Tamanho da página
  - Velocidade do disco sobre o desempenho do sistema de gerenciamento de memória virtual por paginação em um nível
- Obter estatísticas sobre o tempo em que os processos ficam bloqueados
- Tempo médio de transferência de uma página e ainda sobre a taxa de ocupação da memória física.

### Hipóteses

#### Hipótese 1

$H_0$ : Tbloq LRU = Tbloq LFU

$H_1$ : Tbloq LFU  $\neq$  Tbloq LRU

### **Hipótese 2**

$H_0$ :  $T_{bloq\ p4KB} \geq T_{bloq\ p16KB}$

$H_1$ :  $T_{bloq\ p4KB} < T_{bloq\ p16KB}$

\*Tbloq = Tempo médio que os processos ficam bloqueados pelo gerenciador de memória

\*p4KB = Páginas de tamanho de 4KB

\*p16KB = Páginas de tamanho de 16KB

## **3. Planejamento do projeto**

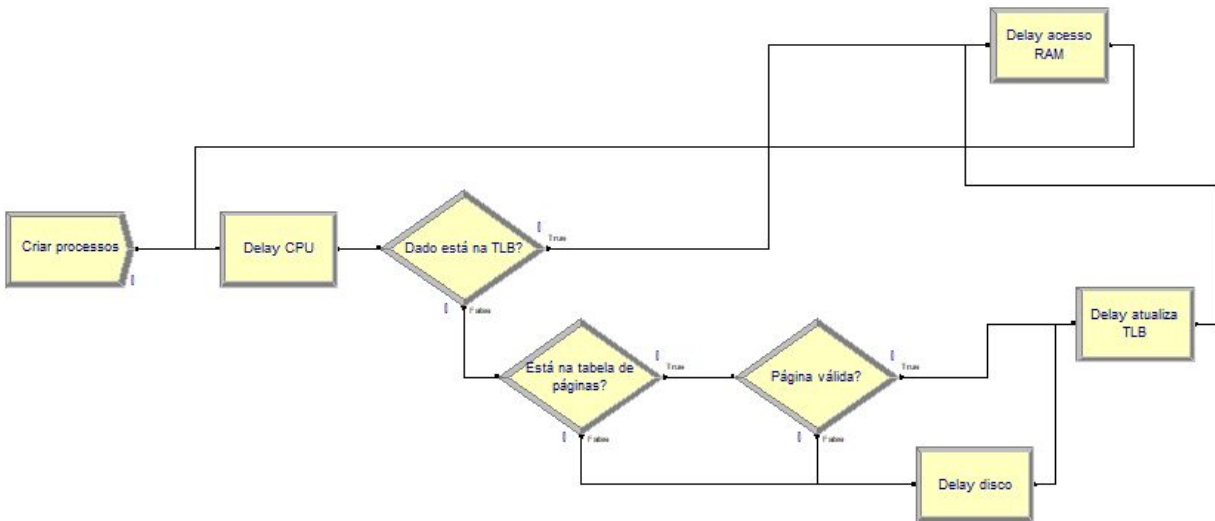
Os recursos de hardware necessários para este projeto foram disponibilizados pela UFSC e pelos membros do grupo. Conforme orientado em aula, foi utilizado o software de simulação Arena sob a versão para estudantes, que está limitada a 150 entidades e 150 blocos. O projeto modelado utilizou poucos blocos e entidades, não sendo afetado pelas limitações desta versão do Arena.

Conforme podemos ver detalhadamente na etapa de projeto experimental final, foram simulados 8 cenários em total, correspondentes à combinação dos três seguintes fatores:

- Algoritmo de Substituição de páginas (LRU e LFU)
- Tamanho da página (4 KB e 16 KB)
- Taxa transf. disco/mem (500 ns / KB e 1  $\mu$ s / KB)

## **4. Formulação do modelo conceitual**

Nesta etapa do projeto, foi discutida como seria realizada a modelagem através das definições do enunciado, implementando as primeiras versões do projeto de simulação. Também nesta etapa foram levantadas as variáveis que seriam utilizadas para realizar a simulação. Conforme orientado em aula, optamos pela simulação discreta orientada a eventos.



**Figura 2. Esboço geral do sistema. Fonte: elaboração própria.**

A figura 2 mostra um esboço em alto nível do sistema modelado. As variáveis principais a serem modeladas são o tamanho das páginas e a velocidade de transferência do disco. Os dados de entrada estão disponíveis no arquivo “tempo\_entre\_referencias\_memoria\_ns.txt”. Os resultados serão apresentados por meio dos arquivos .dat, que foram utilizados no output analyser para testar as hipóteses. Para avaliar a correlação dos fatores sobre a métrica de desempenho, foi utilizado o design expert.

## 5. Coleta de dados e macro-informações

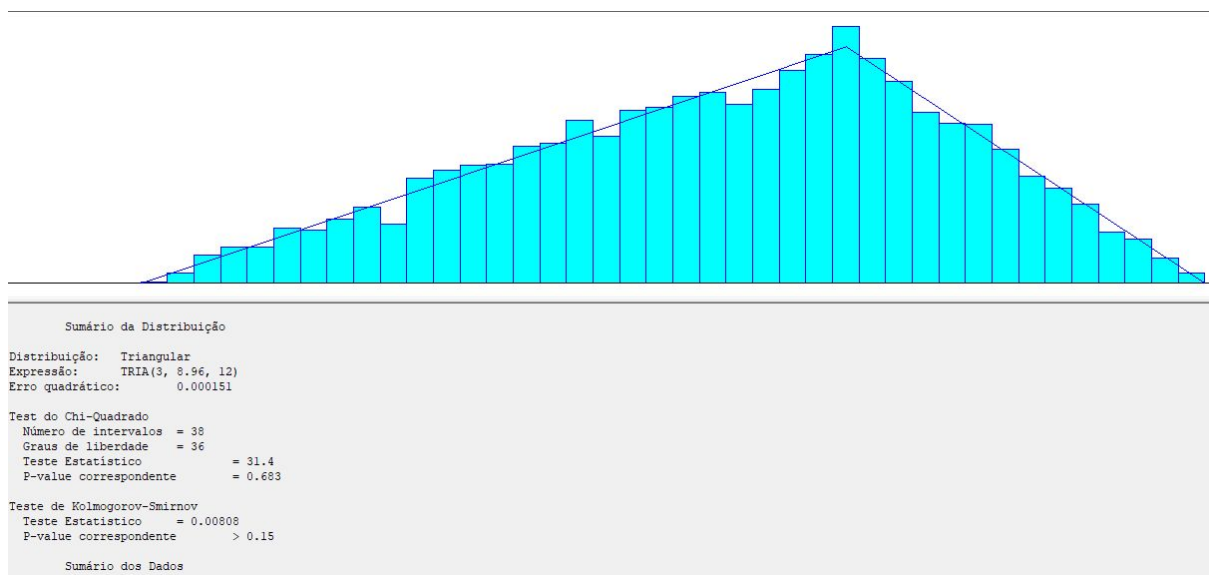
Os dados necessários a serem coletados para o desenvolvimento deste projeto é o tempo entre referências à memória feita pelos processos. Esses dados já são fornecidos para estudo, conforme o arquivo

### Tratamento de dados coletados

A partir da análise dos dados de coleta que foram oferecidos pelo professor no arquivo “tempo\_entre\_referencias\_memoria\_ns.txt” por meio da ferramenta Input Analyser, foi constatado que a melhor forma de distribuição cabível nos dados seria a distribuição triangular.



A distribuição triangular ocorre, principalmente, quando se desconhece a curva associada a uma variável aleatória, mas se tem boas estimativas dos seus limites: inferior (a) e superior (b), bem como de seu valor mais provável (m). Por exemplo, é conhecido o valor mínimo, mais provável e máximo associados ao tempo necessário para a realização de um teste de qualidade de um produto. Pelo fato de se estar estimando o comportamento de variáveis com uma informação a mais (o valor mais provável), implica, em geral, perspectivas de resultados mais aderentes à realidade do que aqueles com base apenas nos valores mínimo e máximo (com o emprego da distribuição uniforme) (FREITAS, 2018).



**Figura 3. Distribuição utilizada: Distribuição triangular - TRIA(a=3, m=8.96, b=12).**

Como podemos observar na figura 3, o Input Analyser encontrou uma distribuição triangular com limite inferior  $a=3$ , limite superior  $b=12$  e valor mais provável  $m=8,96$ . O Input Analyser também fez dois testes de aderência: qui-quadrado e Kolmogorov-Smirnov; e obteve valores baixos para o p-value em ambos testes, portanto podemos dizer que esta distribuição representa bem os dados obtidos experimentalmente. Este fato também pode ser confirmado visualmente pela figura gerada pelo Arena.

## 6. Tradução do modelo

Durante esta fase, certas precauções foram tomadas com o objetivo de garantir a legibilidade do modelo. Os nomes das variáveis, blocos e estatísticas foram cuidadosamente escolhidos de forma a garantir a clareza dos diferentes componentes do modelo.

O desenvolvedor de um sistema operacional deseja avaliar o impacto de:

- A. Algoritmo de substituição de páginas;
- B. Tamanho da página;
- C. Velocidade do disco.

Variable - Basic Process								
	Name	Comment	Rows	Columns	Data Type	Clear Option	File Name	Initial Values
1	MemorySize	Tamanho da memoria em bytes			Real	System		1 rows
2	<u>PageSize</u>	Tamanho da pagina em bytes			Real	System		1 rows
3	Process.Count	Numero de processos que existem atualmente			Real	System		0 rows
4	Process.MaxCount	Numero maximo de processos a ser criado			Real	System		1 rows
5 ▶	TLB	TLB dos processos, cada linha eh uma entrada e cada coluna eh	6	2	Real	System		0 rows
6	<u>DiskTransferRate</u>	Taxa de transferencia do disco em ns/Kb			Real	System		1 rows
7	TLB.I	Indice para busca na TLB			Real	System		0 rows
8	TLB.Min	Indice da linha a ser removida da TLB			Real	System		0 rows
9	PageTable1	Tabela de paginas do processo 1	64	2	Real	System		0 rows
10	PageTable2	Tabela de paginas do processo 2	64	2	Real	System		0 rows
11	MainMemory		64	2	Real	System		0 rows
12	Mem.ToRemove				Real	System		0 rows
13	Mem.I				Real	System		0 rows
14	Mem.Usage	O numero de paginas carregadas para a memoria principal			Real	System		1 rows
15	BreakLoop				Real	System		0 rows
16	Suspend				Real	System		0 rows
17	TLB.Hits				Real	System		0 rows

**Figura 4. Variáveis do modelo de simulação no Arena.**

Na figura 4, são apresentadas as variáveis do modelo de simulação traduzido para o Arena, inclusive com as variáveis que representam o tamanho de uma página (*PageSize*) e velocidade de transferência do disco (*DiskTransferRate*). O Algoritmo de paginação foi feito como dois arquivos de simulação separados, um para LRU e outro para LFU.

Sobre o desempenho do sistema de gerenciamento de memória virtual por paginação em um nível, que é medido como o **tempo médio de acesso a um dado referenciado na memória**. O desenvolvedor do sistema operacional deseja obter **estatísticas sobre o tempo em que os processos ficaram bloqueados, o tempo médio de transferência de uma página** e ainda sobre a **taxa de ocupação da memória física**.

Statistic - Advanced Process					
	Name	Type	Expression	Report Label	Output File
1	BlockedTime	Output	TAVG(Hold.suspended.Queue.WaitingTime)	BlockedTime	blocked_time.dat
2	AccessTime	Output	TAVG(MemoryAccess.OtherTime)	AccessTime	access_time.dat
3	MemoryUsage	Output	(Mem.Usage / PageCount) * 100	MemoryUsage	memory_usage.dat

**Figura 5. Estatísticas coletadas pelo Arena.**

Na figura 5, as estatísticas cadastradas para coleta no Arena.

Como conversado com o cliente, o tempo médio de transferência é considerado constante em cada cenário é dado pela fórmula abaixo:

$$T = R \times \frac{S}{1024}$$

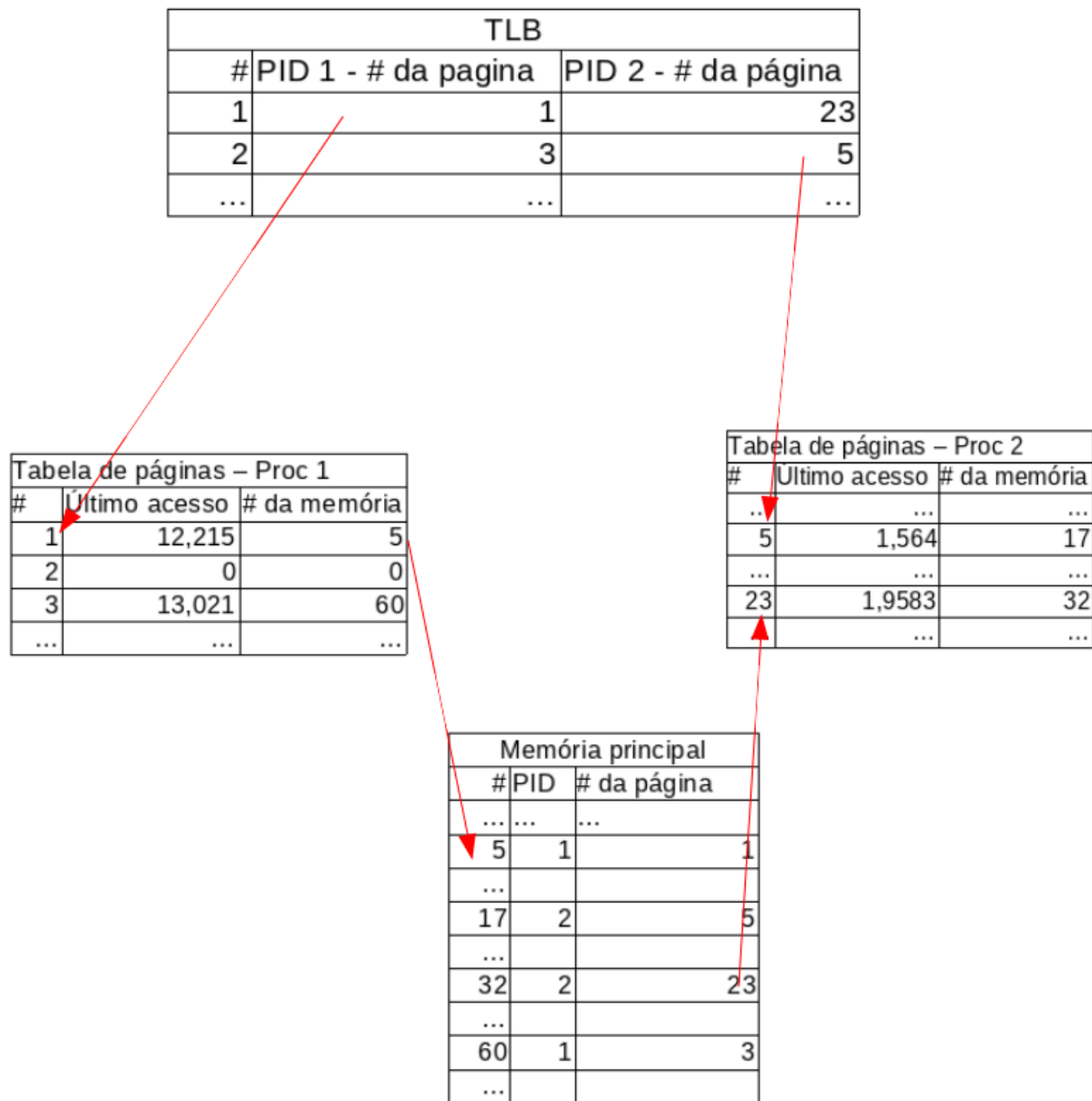
Onde:

T : Tempo de transferência (em Nanosegundos)

R : Velocidade de transferência do disco (em ns/KB );

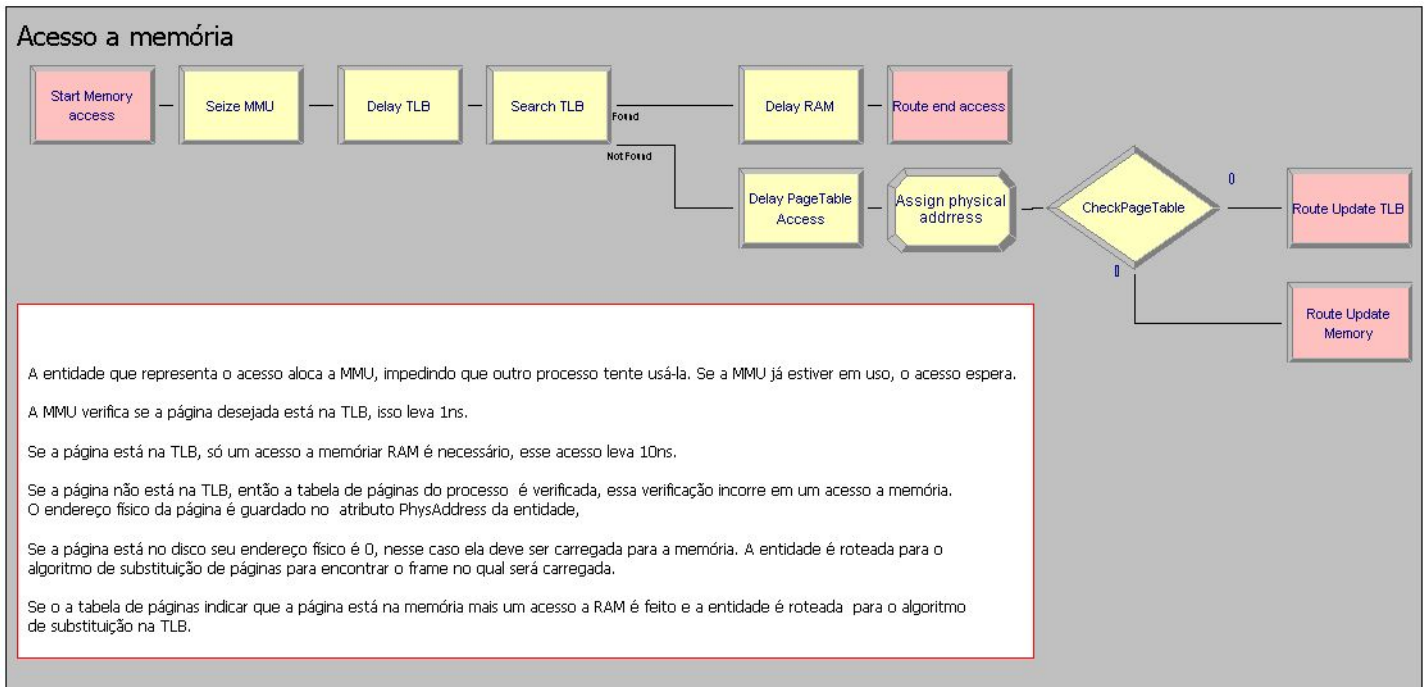
S : Tamanho da página (em Bytes).

O sistema é composto pelas referências a páginas virtuais (que contém os dados referenciados), que são parte dos endereços virtuais gerados pelos processos em execução.



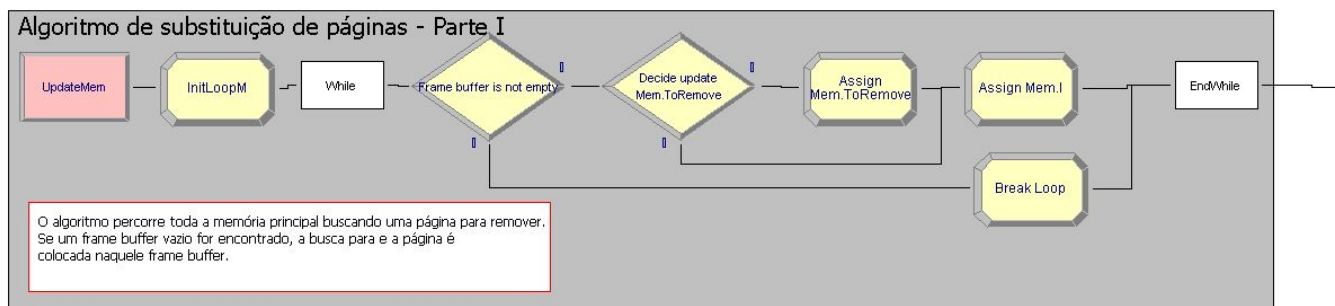
**Figura 6.** Interação entre os componentes do sistema.

O mecanismo de hardware de memória virtual deve verificar **se a referência está na TLB, na memória RAM ou no disco.**

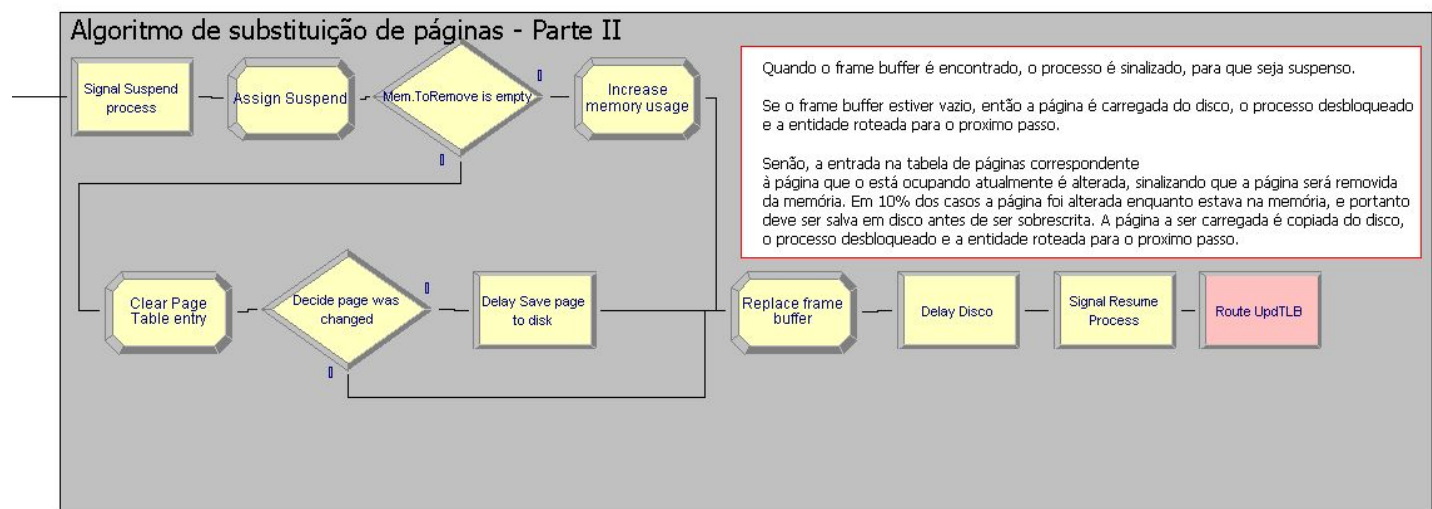


**Figura 7. Acesso a memória modelado no Arena.**

Se a memória física tiver todos seus frames alocados, então uma página física (frame) deve ser escolhida para ser removido da memória (frame vítima), conforme um algoritmo de substituição de páginas. A política de substituição de páginas é local. Se a página física tiver sido alterada, ela primeiro deve ser transferida inteiramente para o disco antes de receber a nova página virtual. Os algoritmos de substituição de páginas a serem avaliados são o LRU (*Least Recently Used*), que substitui a página referenciada a mais tempo no passado e o LFU (*Least Frequently Used*), que substitui a página menos frequentemente usada.



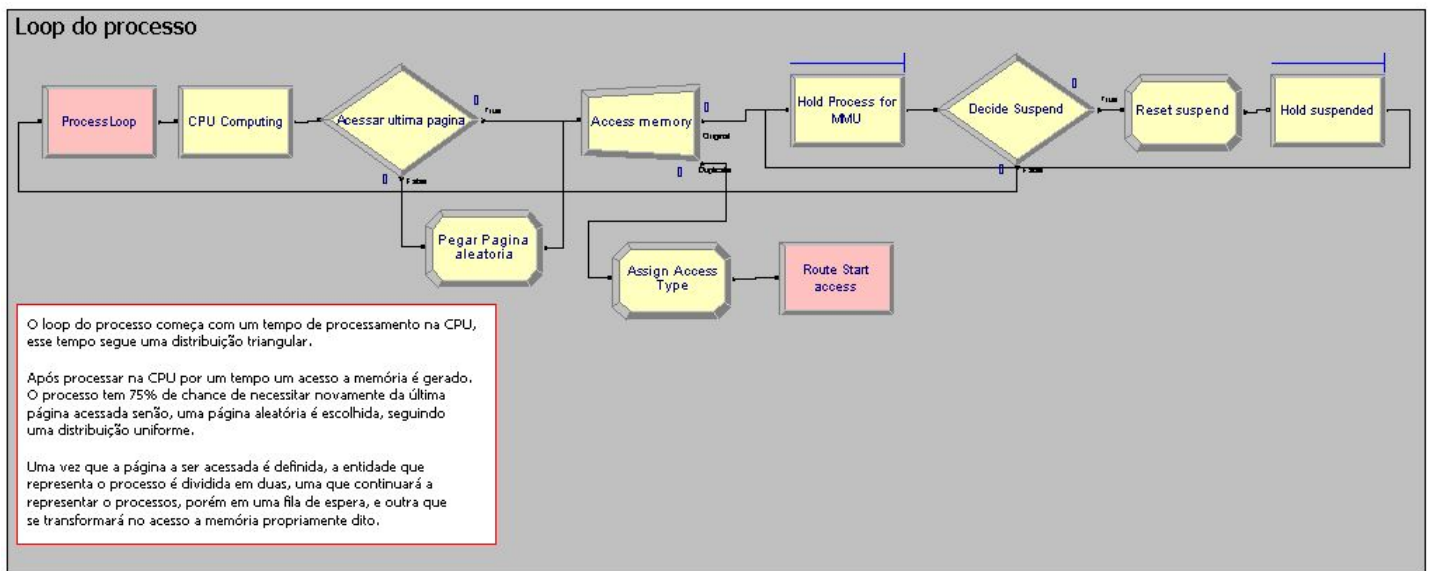
**Figura 8. Primeira parte do algoritmo de substituição de páginas.**



**Figura 9. Segunda parte do algoritmo de substituição de páginas.**

Há 2 processos executando. Cada processo em execução (não bloqueado) gera uma referência à memória apenas após sua última referência ter sido atendida, após um atraso que foi medido muitas vezes (em ns) e que está no arquivo "tempo\_entre\_referencias\_memoria\_ns.txt". **98% das referência a páginas virtuais são para a última página virtual referenciada pelo processo**, e o restante para uma página virtual aleatória qualquer.

Como conversado com o cliente, a chance de uma referência ser para a última página acessada foi alterada para **75%**.



**Figura 10. Loop do processo.**

A quantidade de páginas virtuais depende do tamanho da página, sendo que os endereços virtuais possuem 16 bits. A memória física possui 256KB. As páginas podem variar de 4 KB a 16 KB (níveis mínimo e máximo).

Ele também gostaria de saber se pode afirmar, com 90% de certeza, que o tempo médio que processos ficam esperando (bloqueados) pelo gerenciador de memória é diferente para cada algoritmo avaliado, e se páginas de 4KB levam a menos tempo de bloqueio dos processos que páginas de 16KB. Essas afirmações serão verificadas em fases posteriores.

Simule esse sistema por **1 hora**, com tempo de aquecimento de **10%**. Como conversado com o cliente, a simulação por um tempo de 1 hora demanda muito poder computacional, portanto o tempo foi alterado para **0,25 segundos**, com tempo de aquecimento de **10%**.

## 7. Verificação e validação

De acordo com Freitas (2008), tanto a qualidade quanto a validade de um modelo de simulação estão diretamente relacionados com a proximidade entre os resultados do modelo e os do sistema real. A etapa de validação consiste em saber se o modelo se comporta de forma similar ao sistema real, e a etapa de verificação consiste em analisar se o modelo se comporta da maneira esperada, identificando e corrigindo eventuais erros no modelo.

Na etapa de verificação, o modelo foi depurado por meio do *debugger* do Arena utilizando *breakpoints*, e também executando a simulação passo a passo com animações, observando o caminho que cada entidade percorre, assim como observando a quantidade de entidades que passam por cada blocos e verificando se condiz com o esperado. A barra de *runtime elements*, que contém os atributos das entidades, também foi de grande auxílio para esta etapa.

A validação do sistema foi feita ao comparar o modelo produzido com a descrição do enunciado, tendo em vista que o sistema real proposto pelo enunciado não existe. Também foram consultados livros sobre organização de computadores (PATTERSON, 2010), para compreender corretamente o projeto a ser implementado.

## 8. Projeto experimental final

Nesta etapa do projeto é feito o planejamento dos cenários a serem cobertos pela modelagem do problema. Ao variar cada um dos três fatores considerados (algoritmo de substituição de páginas, tamanho da página e taxa de transferência), observando que cada um dos fatores pode assumir dois níveis, obtivemos 8 cenários que devem ser simulados, conforme ilustrado na tabela 1.



Cenários			
Algoritmo de Substituição	Tamanho da página	Taxa transf. disco/mem	Número do cenário
LRU	4KB	500 ns / KB	1
		1 $\mu$ s / KB	2
	16KB	500 ns / KB	3
		1 $\mu$ s / KB	4
LFU	4KB	500 ns / KB	5
		1 $\mu$ s / KB	6
	16KB	500 ns / KB	7
		1 $\mu$ s / KB	8

**Tabela 1. Cenários a serem simulados.**

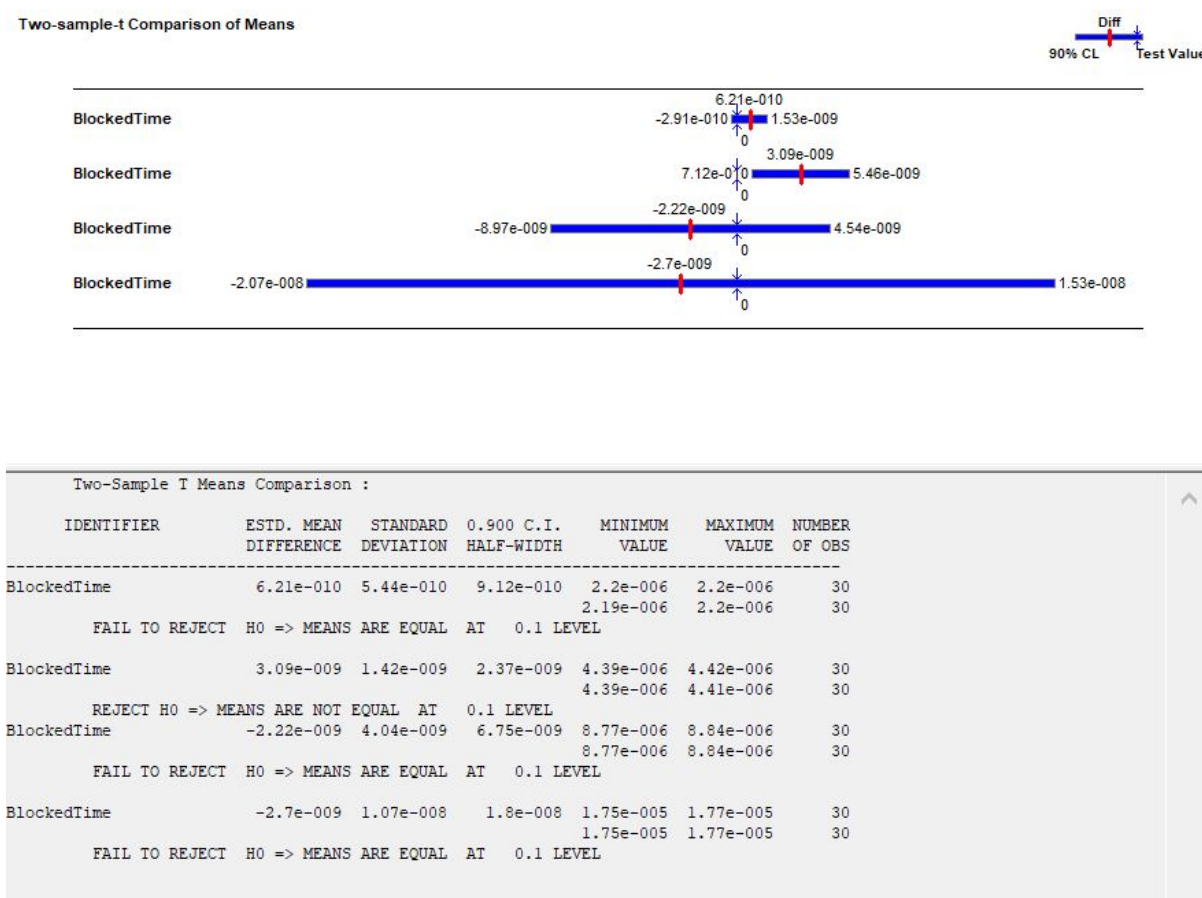
Considerando que a quantidade de cenários é baixa ao variar um único fator por vez, são necessários poucos experimentos e, por tanto, é viável utilizar o projeto experimental simples. Pelo mesmo motivo, neste caso, não é necessário executar um projeto fatorial completo, fracionário ou utilizar outra técnica que permita reduzir a quantidade de experimentos.

## 9. Experimentação

Para cada um dos cenários foram obtidas 30 replicações, e com base nos resultados das mesmas, na próxima etapa poderá ser calculada a quantidade de replicações necessárias para satisfazer os objetivos com a precisão requerida.

## Resultados obtidos:

- Hipótese 1



**Figura 11.** Resultados obtidos através do output analyzer

A figura 11 representa os resultados a fim de obter conclusões sobre a Hipótese 1, ou seja, se o tempo médio que os processos ficam bloqueados pelo gerenciador de memória é diferente para os algoritmos LRU e LFU.

E com esses resultados pode-se concluir que o tempo médio que os processos ficam bloqueados é sim diferente para cada algoritmo de substituição de páginas, pois para o cenário onde o tamanho das páginas = 4KB e a taxa de transferência do disco é 1000ns/KB, rejeita-se com 90% de certeza a hipótese nula em que o tempo médio que

os processos ficam esperando (bloqueados) no caso do LRU é igual ao tempo médio dos processos que os processos ficam esperando (bloqueados) no caso do LFU.

Para os demais casos (págs. 4KB e taxa transf. 500ns; págs. 16KB e taxa transf. 500us; págs. 16KB e taxa transf. 1000ns) não foi possível rejeitar a hipótese nula, e portanto aceita-se a hipótese nula de que o tempo médio que os processos ficam esperando (bloqueados) sob a influência dos dois diferentes algoritmos de substituição de página é igual. Entretanto, como para o cenário págs. 4KB e taxa transf. 1000ns, aceitamos a hipótese alternativa, podemos afirmar, com 90% de certeza, que o tempo médio que processos ficam esperando (bloqueados) pelo gerenciador de memória é diferente para cada algoritmo avaliado.

- Hipótese 2

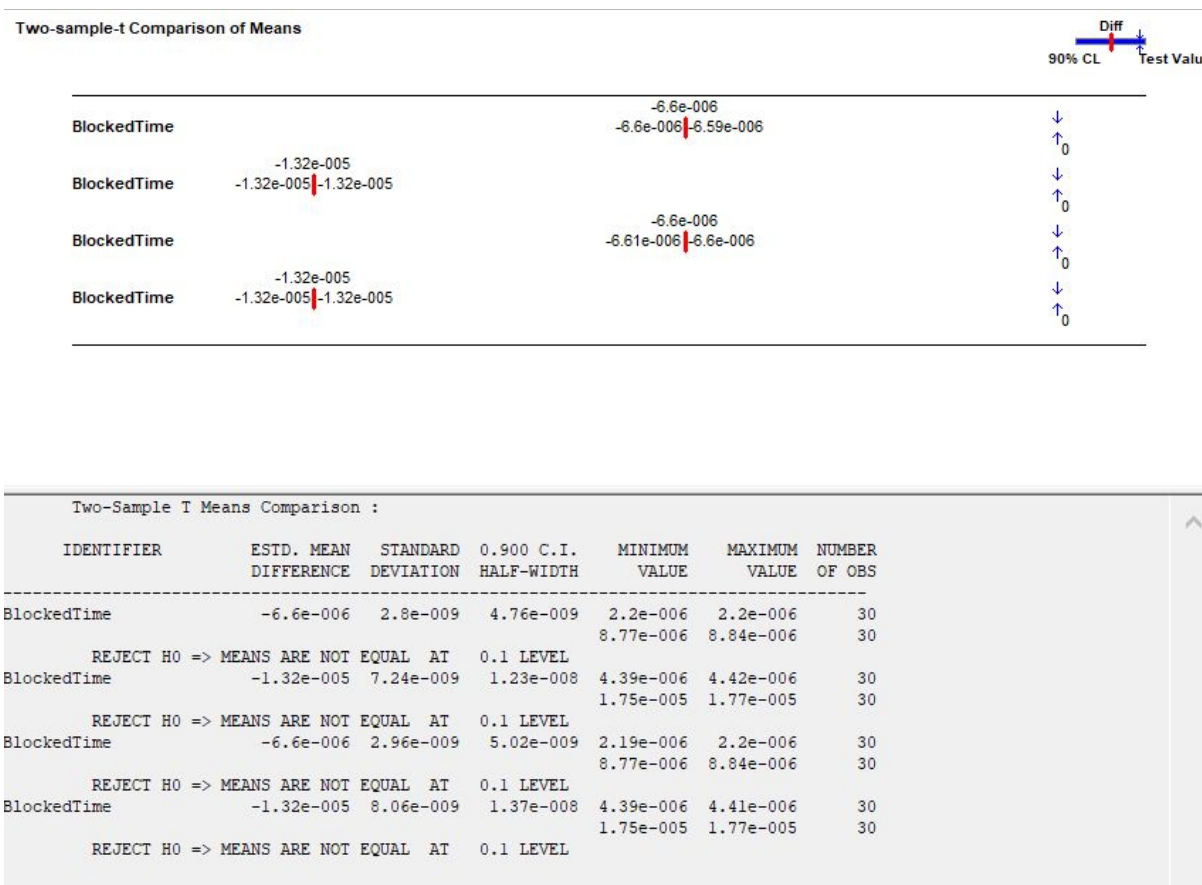


Figura 12 - Resultados obtidos através do output analyzer

Na figura 12 pode-se observar os resultados para a seguinte hipótese: se o tempo de bloqueio para páginas de tamanho de 4KB é menor que o tempo de bloqueio para páginas de tamanho de 16KB. De acordo com tais observações, pode-se concluir que o tempo de bloqueio de processos com páginas de tamanho 4KB e 16KB são diferentes em todos os cenários (conforme descritos na tabela 1). Portanto, conforme as figuras acima pode-se rejeitar a hipótese nula para esta questão, ou seja, aceitamos a hipótese alternativa de que o tempo de bloqueio de processos com páginas de tamanho de 4KB é menor que o tempo de bloqueio de processos com páginas de tamanho 16KB.

Portanto, com a comparação das médias através dos cenários nas figuras abaixo, conclui-se que ambos algoritmos (LRU e LFU) com páginas de tamanho 4KB é melhor, ou seja mais rápido, que páginas com tamanho de 16KB.

#### Cenário 1 - LRU - Páginas 4KB e Taxa Transferência do Disco 500ns

Classical C.I. Intervals Summary						
IDENTIFIER	AVERAGE	STANDARD DEVIATION	0.900 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
BlockedTime	2.2e-006	1.58e-009	4.89e-010	2.2e-006	2.2e-006	30

#### Cenário 2 - LRU - Páginas 4KB e Taxa Transferência do Disco 1000ns

Classical C.I. Intervals Summary						
IDENTIFIER	AVERAGE	STANDARD DEVIATION	0.900 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
BlockedTime	4.4e-006	4.65e-009	1.44e-009	4.39e-006	4.42e-006	30

#### Cenário 3 - LRU - Páginas 16KB e Taxa Transferência do Disco 500ns

Classical C.I. Intervals Summary						
IDENTIFIER	AVERAGE	STANDARD DEVIATION	0.900 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
BlockedTime	8.8e-006	1.53e-008	4.74e-009	8.77e-006	8.84e-006	30

Cenário 4 - LRU - Páginas 16KB e Taxa Transferência do Disco 1000ns

Classical C.I. Intervals Summary						
IDENTIFIER	AVERAGE	STANDARD DEVIATION	0.900 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
BlockedTime	1.76e-005	3.94e-008	1.22e-008	1.75e-005	1.77e-005	30

Cenário 5 - LFU - Páginas 4KB e Taxa Transferência do Disco 500ns

Classical C.I. Intervals Summary						
IDENTIFIER	AVERAGE	STANDARD DEVIATION	0.900 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
BlockedTime	2.2e-006	2.53e-009	7.85e-010	2.19e-006	2.2e-006	30

Cenário 6 - LFU - Páginas 4KB e Taxa Transferência do Disco 1000ns

Classical C.I. Intervals Summary						
IDENTIFIER	AVERAGE	STANDARD DEVIATION	0.900 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
BlockedTime	4.4e-006	6.23e-009	1.93e-009	4.39e-006	4.41e-006	30

Cenário 7 - LFU - Páginas 16KB e Taxa Transferência do Disco 500ns

Classical C.I. Intervals Summary						
IDENTIFIER	AVERAGE	STANDARD DEVIATION	0.900 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
BlockedTime	8.8e-006	1.6e-008	4.96e-009	8.77e-006	8.84e-006	30

Cenário 8 - LFU - Páginas 16KB e Taxa Transferência do Disco 1000ns

Classical C.I. Intervals Summary						
IDENTIFIER	AVERAGE	STANDARD DEVIATION	0.900 C.I. HALF-WIDTH	MINIMUM VALUE	MAXIMUM VALUE	NUMBER OF OBS.
BlockedTime	1.76e-005	4.37e-008	1.36e-008	1.75e-005	1.77e-005	30

## 10. Interpretação e análise dos resultados

Ao medir a taxa de ocupação da memória física, observamos que os algoritmos de substituição de páginas utilizados mantêm a memória sempre 100% ocupada, pois as páginas somente são removidas quando são substituídas por outras, pela natureza destes algoritmos. O tempo de *warmup* requerido no enunciado foi suficiente, e inclusive poderia ser menor. Este sistema foi considerado como terminal.

### Replicações

A fim de calcular o número de replicações necessárias foi testado com o cenário 1, descrito na tabela 1. Inicialmente com 10 replicações, com 30, e com 60. Procuramos diferenças entre as médias e variâncias para as métricas calculadas nestes cenários e concluímos que o melhor número de replicações seria 30.

Para cálculo de replicações a fórmula utilizada foi a seguinte:

$$n \geq (t_{crit}^2 * S^2) / E^2$$

Que resultou em  $n = 0,00208$ . Consideramos então as diferenças como explicado anteriormente.

## 11. Comparação e identificação das melhores soluções

Para encontrar a melhor solução, os oito cenários foram comparados entre si. A principal métrica para escolher a melhor alternativa é o tempo médio de acesso a memória, onde quanto menor o nível deste fator, melhor é o cenário relacionado. Estes resultados são avaliados sob um nível de significância de 90%, conforme especificado no enunciado.

$$\begin{aligned}
 \text{Tempo médio de acesso a memória} &= +1.629\text{E-}007 \\
 &+3.750\text{E-}008 * A \\
 &-6.863\text{E-}008 * B \\
 &-1.140\text{E-}007 * C \\
 &-3.563\text{E-}008 * A * B \\
 &-5.068\text{E-}008 * A * C \\
 &+7.405\text{E-}008 * B * C \\
 &+1.170\text{E-}008 * A * B * C
 \end{aligned}$$

Onde:

- A. Velocidade do disco;
- B. Tamanho de página;
- C. Algoritmo de substituição de páginas.

Sendo que no algoritmo LRU a equação é:

$$\begin{aligned}
 \text{Tempo médio de acesso a memória} &= \\
 &+1.34667\text{E-}008 \\
 &+6.68200\text{E-}010 * \text{Velocidade do disco} \\
 &-1.16667\text{E-}010 * \text{Tamanho da página} \\
 &-3.15500\text{E-}011 * \text{Velocidade do disco} * \text{Tamanho da página}
 \end{aligned}$$

No algoritmo LFU é:

$$\begin{aligned}
 \text{Tempo médio de acesso a memória} &= \\
 &-4.02667\text{E-}008 \\
 &+1.06800\text{E-}010 * \text{Velocidade do disco} \\
 &+1.28667\text{E-}008 * \text{Tamanho da página} \\
 &-1.59500\text{E-}011 * \text{Velocidade do disco} * \text{Tamanho da página}
 \end{aligned}$$

## 12. Documentação

Para a documentação de este projeto, foi confeccionado o presente relatório. Também procurou-se documentar de forma breve e clara o modelo nos arquivos do Arena.

## 13. Apresentação dos resultados e implementação

Através dos resultados obtidos (especificados na seção 10 de interpretação e análise de resultados) foi possível chegar às conclusões sobre os objetivos especificados na formulação do problema.

De acordo com Hasslinger (2018), em média, o algoritmo LFU é mais eficiente; porém, é necessário destacar que a eficiência relativa dos algoritmos LFU e LRU é altamente dependente do programa em execução e de outros parâmetros, como quantidade de memória, velocidade do disco e outros.

Por meio da simulação, chegamos às mesmas conclusões que Hasslinger: encontramos que o tempo de espera entre os dois algoritmos avaliados é diferente, conforme esperado na teoria; e encontramos também que páginas de 4KB levam menos tempo de bloqueio dos processos que páginas de 16KB, conforme esperado teoricamente.



## 14. Referências bibliográficas

**FREITAS FILHO**, P. J. Introdução à Modelagem e Simulação de Sistemas, 2ª Ed, Visual Books, 2008

**PATTERSON**, DAVID A; **HENNESSY**, JOHN L.; Computer Organization and Design: The Hardware/Software Interface. 4rd edition, Morgan Kaufmann Publishers, San Francisco, California, USA, 2010.

**CASSETTARI**, H. H. Análise da Localidade de Programas e Desenvolvimento de Algoritmos Adaptativos para Substituição de Páginas. Qualificação de Mestrado. Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais. 2003.

**HASSLINGER**, GERHARD; **HEIKKINEN**, JUHO; **NTOUGIAS**, KONSTANTINOS; **HASSLINGER**, FRANK; **HOHLFELD**, OLIVER. Optimum Caching versus LRU and LFU: Comparison and Combined Limited Look-Ahead Strategies. Shanghai, China. 24 de maio de 2018. INSPEC Accession Number: 17789104, DOI: 10.23919/WIOPT.2018.8362880, Publisher: IEEE. Disponível em <<https://ieeexplore.ieee.org/document/8362880/>>. Acesso em 9 de junho de 2019, 18:24:17.