

PP4 - Gerenciamento de memória virtual

1.Introdução.

1.1 Enunciado

O desenvolvedor de um sistema operacional deseja avaliar o impacto (a) do algoritmo de substituição de páginas, (b) do tamanho da página e (c) da velocidade do disco 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 sistema é composto pelas referências a páginas virtuais (que contém os dados referenciados), que são parte dos endereços virtuais gerados pelos processos em execução. O mecanismo de hardware de memória virtual deve verificar se a referência está na TLB (uma cache do mecanismo de paginação), cujo acesso leva 1ns e, se estiver, há um acerto na TLB e a página referenciada está na memória RAM. Neste caso, basta um único acesso à memória RAM (cujo acesso leva 10ns) e o dado é acessado. Se não houver acerto na TLB, o hardware deve acessar a tabela de páginas daquele processo na memória RAM. Se a tabela de páginas indicar que a página referenciada é válida, então ela está na memória RAM. Neste caso, é necessário atualizar a TLB (1ns) e fazer um novo acesso à memória RAM para acessar o dado. Porém, se a tabela de páginas do processo informar que a página não é válida, então ela deve ser buscada do disco. Neste caso, o processo deve ser bloqueado, ou seja, impedido de continuar executando, e a página faltante do disco deve ser transferida para a memória principal. O tempo de transferência é de 1us por KB transferido para discos lentos e de 500ns/KB para discos rápidos (níveis mínimo e máximo). Se houver páginas físicas (frames) livres na memória principal, então o sistema operacional escolhe uma frame livre qualquer, transfere a página do disco para aquela frame e, ao término da transferência, atualiza a tabela de páginas do processo, conclui o acesso ao dado faltante (um acesso à RAM) e desbloqueia o processo. 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. A memória física possui 256KB. As páginas podem variar de 4KB a 16KB (níveis mínimo e máximo). 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 referências a páginas virtuais são para a última página virtual referenciada pelo processo, e o restante para uma página virtual

aleatória qualquer. A quantidade de páginas virtuais depende do tamanho da página, sendo que os endereços virtuais possuem 16 bits. Além de avaliar o impacto dos fatores citados sobre a métrica de desempenho, 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. 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. Simule esse sistema por 1 hora, com tempo de aquecimento de 10%.

1.2 Objetivos (requisitos funcionais e não funcionais).

O presente trabalho tem como objetivo, desenvolver componentes necessários no simulador GenESYs, e a partir do modelo criado para o simulador arena no trabalho enunciado na seção 1.1, criar o mesmo modelo no GenESYs e obter resultados estatisticamente iguais.

RFs

Os requisitos funcionais deste projeto é composto por implementações e testes de componentes que são necessários no simulador. Abaixo estão listados esses componentes

- Componente "Station"
- Componente "Seize".
- Componente "Create"
- Componente "Delay"
- Componente "Search".
- Componente "Assign".
- Componente "Decide".
- Componente "Route"
- Componente "Hold"
- Componente "Separate"
- Componente "Dispose"
- Componente "While" e "Endwhile"

RNFs

Modelar no genesys modelos análogos aos que já existem no arena.

Realizar simulação do modelo que foi simulado no arena, no genesys.

2. Planejamento do projeto.

Para alcançar os objetivos, o projeto como um todo foi dividido em etapas, cada uma dessas etapas com um escopo bem definido do que deve ser realizado. No final de um conjunto de etapas, quando cabível, foram feitos testes pequenos, apenas para validações iniciais (modelos com poucos componentes e o uso do gdb, para avaliar o comportamento desses componentes).

1ª Etapa: Realizar simulação no arena.

Essa etapa inicial consiste em executar o modelo existente no Arena para obter uma maior familiarização, tanto com o modelo a ser reproduzido, quando com a ferramenta de simulação.

2ª Etapa: Implementar componente *hold* e *signal*

Os componentes *hold* e *signal* costumam trabalhar em conjunto, portanto é plausível realizar o desenvolvimento dos dois em uma única etapa.

3ª Etapa: Testar os componentes *create* e *dispose*

Esses dois componentes comumente iniciam, e terminam uma simulação respectivamente, portanto para testar os demais componentes, é necessário que ambos funcionem.

4ª Etapa: Testar o componentes *station(enter)* e *route*,

Novamente, são dois componentes (3 se considerar o *station* como um componente também) que operam em conjunto em um modelo, portanto faz sentido realizar os testes deles juntos.

5ª Etapa: Implementar componentes *While* e *Endwhile*

Esses componentes são dois componentes que operam em conjunto, portanto é necessário que o desenvolvimento deles seja feito em conjunto.

6ª Etapa: testar componente *decide*, *assign*, *seize* e *delay*

Esses componentes foram os que sobraram, portanto nessa etapa, como o projeto já está mais avançado, faz sentido testar todos eles de uma vez, e já dar início à integração com o modelo final.

Etapa Final: Integrar componentes e corrigi-los se necessários.

Essa etapa será a etapa final, nessa etapa os componentes irão compor o modelo final e este será executado no genesys, caso necessário, nessa etapa serão feitas as correções necessárias

3. Desenvolvimentos

3.1 Hold e Signal

O desenvolvimento do hold e do signal foi feito em conjunto, no modelo, os holds existentes utilizam do signal para saber quando liberar as entidades.

Basicamente o que o hold faz é, quando o tipo dele é esperar por um sinal, ele guarda a entidade que está nele em uma fila.

```
else if (_type == Type::WaitForSignal) {
    Waiting* waiting = new Waiting(entity, this, _model->getSimulation()->getSimulatedTime());
    this->_queue->insertElement(waiting);
}
}
```

Essa entidade só será liberada quando uma entidade passar por um signal, e esse signal fará uma busca por holds que estão esperando por ele e chamará no hold uma função para liberar “limit” entidades da fila

```
template<typename Base, typename T>
inline bool instanceof(const T*) {
    return std::is_base_of<Base, T>::value;
}

void Signal::_execute(Entity* entity) {
    std::list<ModelComponent*>::iterator it = _model->getComponentManager()->begin();
    for (; it != _model->getComponentManager()->end(); it++) {
        auto component = *it;
        if (instanceof<Hold>(*it)) {
            Hold* h = ((Hold*)(*it));
            if (h->getWaitForValueExpr() == signalName) {
                h->release_signal(limit);
            }
        }
    }
}
```

```

void Hold::release_signal(int _limit) {
    for(int i = 0; i < _queue->size(); i++) {
        Waiting* waiting = _queue->getAtRank(i);
        _model->sendEntityToComponent(waiting->getEntity(), this->getNextComponents()->front(), 0.0);
        if (i >= _limit) {
            break;
        }
    }
}

```

3.2. While e Endwhile

O funcionamento do while com o endwhile é o seguinte:

- uma vez que a entidade entra no while, uma condição é testada
 - Caso essa condição seja verdadeira, a entidade passa para o próximo componente (ligado ao while).
 - Se essa condição for falsa, a entidade é enviada diretamente para o componente que ao qual o endwhile está conectado.
- E toda vez que uma entidade entra em um endwhile, ela é enviada para o while associado a esse endwhile.

As imagens a seguir mostram esse comportamento implementado nos dois componentes.

```

void While::_execute(Entity* entity) {
    auto value = _model->parseExpression(_condition);
    if (value) {
        _model->sendEntityToComponent(entity, this->getNextComponents()->front(), 0.0);
    } else {
        _model->sendEntityToComponent(entity, _after_endwhile, 0.0);
    }
}

void Endwhile::_execute(Entity* entity) {
    _model->sendEntityToComponent(entity, _attached_while, 0.0);
}

```

3.3 Alterações no Assign

No assign, foram feitas alterações para permitir que sejam atribuídos valores a variáveis de duas dimensões, e para que a entidade que esteja passando pelo assign troque de tipo.

```

/* TODO: +- an enum is not a good idea. Should b
enum class DestinationType : int {
    Attribute=0, Variable=1, Entity=2
};

Assignment(DestinationType destinationType, std::string destination, std::string expression, std::string row, std::string c
    this->_destinationType = destinationType;
    this->_destination = destination;
    this->_expression = expression;
    this->_row = row;
    this->_column = column;
};

```

Foi criado esse novo construtor de assignment para passar os parâmetros de linha e coluna da variável de duas dimensões.

```

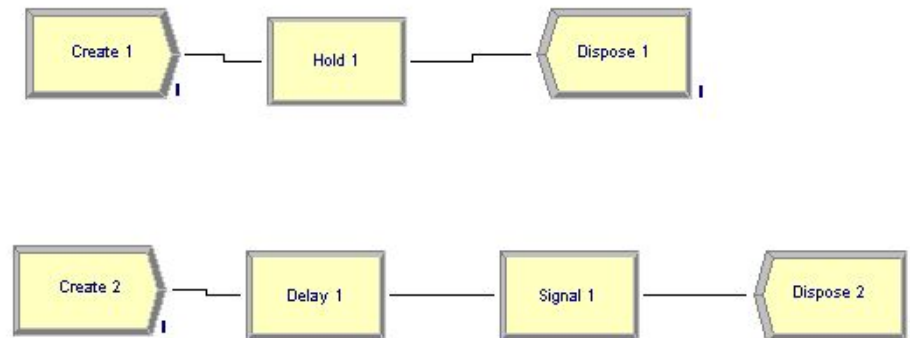
if (let->getDestinationType() == DestinationType::Variable) {
    Variable* myvar = (Variable*) this->_model->getElementManager()->getElement(Util::TypeOf<Variable>(), let->getDestinati
    if (let->getRow() != "" && let->getColumn() != "") {
        std::string index = std::to_string(_model->parseExpression(let->getRow())) + std::to_string(_model->parseExpression
        value = myvar->getValue(index);
        myvar->setValue(index, value);
    } else {
        myvar->setValue(value);
    }
}

```

4. Testes

Como já dito na seção 2, foram realizados alguns testes para uma validação inicial dos componentes implementados. Esses testes são descritos a seguir.

O primeiro teste consiste de um modelo simples, composto de delay, hold e signal, creates e disposes.



Esse teste, no arena, apresentou os seguintes resultados:

ARENA Simulation Results					
gustavo.borgesfr@gmail.com					
Resumo de resultados para 3 Replicações					
Projeto: Unnamed Project			Data da execução :11/ 6/2019		
Analista: gustavo.borgesfr@gmail.com			Data de revisão do modelo:11/ 6/2019		
RESULTADOS					
Identificador	Média	Meia-largura	Mínima	Maximo #	de Replicações
Entity 1.NumberIn	30001.	.00000	30001.	30001.	3
Entity 1.NumberOut	30000.	.00000	30000.	30000.	3
System.NumberOut	30000.	.00000	30000.	30000.	3
Tempo executando a simulação: 0.00 minutos.					
Simulação completada.					

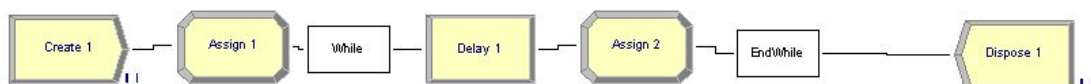
E já no GenESYs os resultados foram os seguintes:

```

Begin of Report for Simulation (based on 3 replications)
  Statistic for Create:
    Create 1:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stdddev ..... varCoef ..... confInterv ..... confLevel .....
      Count number in ..... 3 ..... 9600.000000 9600.000000 9600.000000 0.000000 0.000000 0.000000 0.950000
    Create 2:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stdddev ..... varCoef ..... confInterv ..... confLevel .....
      Count number in ..... 3 ..... 2401.000000 2401.000000 2401.000000 0.000000 0.000000 0.000000 0.950000
  Statistic for Dispose:
    Dispose 1:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stdddev ..... varCoef ..... confInterv ..... confLevel .....
      Dispose 1.Count number out ..... 3 ..... 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.950000
    Dispose 2:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stdddev ..... varCoef ..... confInterv ..... confLevel .....
      Dispose 2.Count number out ..... 3 ..... 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.950000
  Statistic for EntityType:
    Entity:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stdddev ..... varCoef ..... confInterv ..... confLevel .....
      Entity.Total Time ..... 3 ..... 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.950000
      Delay 1.Waiting Time ..... 3 ..... 0.006944 0.006944 0.006944 0.000000 0.000000 0.000000 0.000000 0.950000
  Statistic for Queue:
    Queue HOLD 1:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stdddev ..... varCoef ..... confInterv ..... confLevel .....
      Queue HOLD 1.Number In Queue ..... 3 ..... 4800.500000 4800.500000 4800.500000 0.000000 0.000000 0.000000 0.000000 0.950000
      Queue HOLD 1.Time In Queue ..... 3 ..... 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.950000
End of Report for Simulation
Simulation of model "Model 1" has finished.

```

Outro teste feito foi um teste com whiles assigns e delays, além do create e dispose.



Este teste, por sua vez, no Arena apresentou os seguintes resultados:

RESULTADOS					
Identificador	Média	Meia-largura	Mínima	Maximo	# de Replicações
Entity 1.NumberIn	61.000	.00000	61.000	61.000	3
Entity 1.NumberOut	60.000	.00000	60.000	60.000	3
System.NumberOut	60.000	.00000	60.000	60.000	3

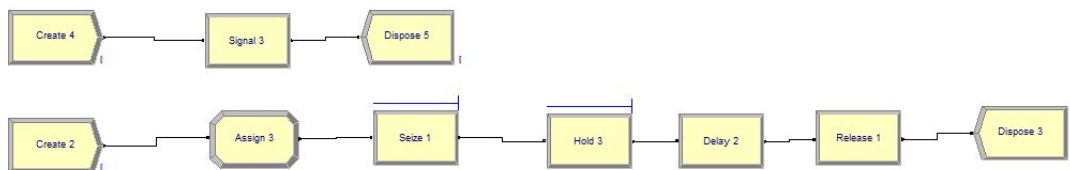
E no GenESYs os resultados:

```

Begin of Report for Simulation (based on 3 replications)
Statistic for Create:
  Create 1:
    name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
    Count number in ..... 3 ..... 25.000000 ..... 25.000000 ..... 25.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
Statistic for Dispose:
  Dispose 1:
    name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
    Dispose 1.Count number out ..... 3 ..... 24.000000 ..... 24.000000 ..... 24.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
Statistic for EntityType:
  Entity:
    name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
    Delay 1.Waiting Time ..... 3 ..... 0.010417 ..... 0.010417 ..... 0.010417 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
    Entity.Total Time ..... 3 ..... 0.114583 ..... 0.114583 ..... 0.114583 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
End of Report for Simulation
Simulation of model "Model 1" has finished.

```

O último teste feito foi com Creates, hold e signal, seize, delay, release e disposes.



Resultados no Arena:

```

Resumo de resultados para 3 Replicações

Projeto: Unnamed Project ..... Data da execução :11/ 3/2019
Analista: gustavo.borgesfr@gmail.com ..... Data de revisão do modelo:11/ 3/2019

.....|RESULTADOS

Identificador ..... Média ..... Meia-largura ..... Mínima ..... Maximo # de Replicações .....
-----|-----
Entity 1.NumberIn ..... 201.00 ..... .000000 ..... 201.00 ..... 201.00 ..... 3
Entity 1.NumberOut ..... 121.00 ..... .000000 ..... 121.00 ..... 121.00 ..... 3
recurso_de_teste.NumberSeized ..... 41.000 ..... .000000 ..... 41.000 ..... 41.000 ..... 3
recurso_de_teste.ScheduledUtilization ..... 1.0000 ..... .000000 ..... 1.0000 ..... 1.0000 ..... 3
System.NumberOut ..... 121.00 ..... .000000 ..... 121.00 ..... 121.00 ..... 3

```

Genesys:

```

Begin of Report for Simulation (based on 3 replications)
  Statistic for Create:
    Create 1:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
      Count number in ..... 3 ..... 48.000000 ..... 48.000000 ..... 48.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
    Create 2:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
      Count number in ..... 3 ..... 33.000000 ..... 33.000000 ..... 33.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
  Statistic for Dispose:
    Dispose 1:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
      Dispose 1.Count_number_out ..... 3 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
    Dispose 2:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
      Dispose 2.Count_number_out ..... 3 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
  Statistic for EntityType:
    Entity:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
      Entity.Total Time ..... 3 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
      Delay 1.Waiting Time ..... 3 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
  Statistic for Queue:
    Queue_HOLD 1:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
      Queue_HOLD 1.Number_In_Queue ..... 3 ..... 1.000000 ..... 1.000000 ..... 1.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
      Queue_HOLD 1.Time_In_Queue ..... 3 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
    Queue_SEIZE 1:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
      Queue_SEIZE 1.Number_In_Queue ..... 3 ..... 24.000000 ..... 24.000000 ..... 24.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
      Queue_SEIZE 1.Time_In_Queue ..... 3 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
  Statistic for Resource:
    test res:
      name ..... elems ..... min ..... max ..... average ..... variance ..... stddev ..... varCoef ..... confInterv ..... confLevel .....
      test res.Time Seized ..... 3 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
      test res.Seizes ..... 3 ..... 1.000000 ..... 1.000000 ..... 1.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
      test res.Releases ..... 3 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.000000 ..... 0.950000 .....
  End of Report for Simulation
Simulation of model "Model 1" has finished.

```

```

| Statistic for Create:
| | Create_1:
| | | name | elems | min | max | average | variance | stddev | varCoef | confInterv | confLevel
| | | Count number in..... | 30 | 126.000000 | 126.000000 | 126.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| Statistic for Dispose:
| | Dispose_1:
| | | name | elems | min | max | average | variance | stddev | varCoef | confInterv | confLevel
| | | Dispose_1.Count_number_out..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| Statistic for EntityType:
| | Init:
| | | name | elems | min | max | average | variance | stddev | varCoef | confInterv | confLevel
| | | Delay_1.Waiting_Time..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| | | Delay_2.Waiting_Time..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| | | Delay_3.Waiting_Time..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| | | Delay_4.Waiting_Time..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| | | Init.Total_Time..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| | | Delay_5.Waiting_Time..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| | | Delay_6.Waiting_Time..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| | | Delay_7.Waiting_Time..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| | | Transfer Time..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000
| | | Time in Stations..... | 30 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.950000

```


Memory Access:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
Delay_1.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Delay_2.Waiting_Time.....	30	0.000000	0.000000	0.000000	-0.000000	-nan	-nan	-nan	0.950000	
Delay_3.Waiting_Time.....	30	0.000000	0.000000	0.000000	-0.000000	-nan	-nan	-nan	0.950000	
Delay_4.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Memory Access.Total_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Delay_5.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Delay_6.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Delay_7.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Transfer Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Time in Stations.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Process:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
Delay_1.Waiting_Time.....	30	0.000000	0.000000	0.000000	-0.000000	-nan	-nan	-nan	0.950000	
Delay_2.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Delay_3.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Delay_4.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Process.Total_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Delay_5.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Delay_6.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Delay_7.Waiting_Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Transfer Time.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Time in Stations.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Statistic for Queue:										
HoldProcess\WUQueue:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
HoldProcess\WUQueue.Number_In_Queue....	30	31.500000	31.500000	31.500000	0.000000	0.000000	0.000000	0.000000	0.950000	
HoldProcess\WUQueue.Time_In_Queue.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
HoldSuspendedQueue:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
HoldSuspendedQueue.Number_In_Queue....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
HoldSuspendedQueue.Time_In_Queue.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Seize\WUQueue:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
Seize\WUQueue.Number_In_Queue.....	30	10.921053	10.921053	10.921053	-0.000000	-nan	-nan	-nan	0.950000	
Seize\WUQueue.Time_In_Queue.....	30	0.000000	0.000000	0.000000	-0.000000	-nan	-nan	-nan	0.950000	
Statistic for Resource:										
MMU:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
MMU.Time_Seized.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
MMU.Seizes.....	30	62.000000	62.000000	62.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
MMU.Releases.....	30	62.000000	62.000000	62.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Statistic for Station:										
ProcLoop:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
ProcLoop.Number_In_Station.....	30	63.500000	3717.500000	1890.500000	1189377.00001090	585622	0.576877	390.261053	0.950000	
ProcLoop.Time_In_Station.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
StartMemAccessStation:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
StartMemAccessStation.Number_In_Station	30	31.500000	1829.500000	930.500000	287979.66666536	637370	0.576719	192.033217	0.950000	
StartMemAccessStation.Time_In_Station..	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Station_3:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
Station_3.Number_In_Station.....	30	31.500000	1829.500000	930.500000	287979.66666536	637370	0.576719	192.033217	0.950000	
Station_3.Time_In_Station.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Station_4:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
Station_4.Number_In_Station.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Station_4.Time_In_Station.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Station_5:										
name	elems	min	max	average	variance	stddev	varCoef	confInterv	confLevel	
Station_5.Number_In_Station.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	
Station_5.Time_In_Station.....	30	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.950000	

Resultados do Arena:

RESULTADOS					
Identificador	Média	Meia-largura	Mínima	Maximo	# de Replicações
MemoryUsage	100.00	.00000	100.00	100.00	30
BlockedTime	4.3989E-06	2.0655E-09	4.3883E-06	4.4090E-06	30
AccessTime	5.4186E-08	8.8103E-11	5.3598E-08	5.4740E-08	30
Process.NumberIn	4.6128E+06	7510.6	4.5660E+06	4.6634E+06	30
Process.NumberOut	4.6128E+06	7510.6	4.5660E+06	4.6634E+06	30
Init.NumberIn	2.0000	.00000	2.0000	2.0000	30
Init.NumberOut	2.0000	.00000	2.0000	2.0000	30
MemoryAccess.NumberIn	4.6128E+06	7510.6	4.5660E+06	4.6634E+06	30
MemoryAccess.NumberOut	4.6128E+06	7510.6	4.5660E+06	4.6634E+06	30
MMU.NumberSeized	4.6128E+06	7510.6	4.5660E+06	4.6634E+06	30
MMU.ScheduledUtilization	.99978	3.9289E-07	.99978	.99979	30
System.NumberOut	4.6128E+06	7510.6	4.5660E+06	4.6634E+06	30
Tempo executando a simulação: 23.40 minutos.					
Simulação completada.					

6.Conclusões

Os objetivos não foram atingidos, os resultados do modelo implementado no GenESYs não são estatisticamente iguais aos do mesmo modelo no Arena, ficaram faltando algumas coisas a serem corrigidas, como algumas expressões no modelo do GenESYs, e também a correção do componente Search. Os componentes implementados provavelmente não estão 100% mas acredito que estejam em um bom estado.