### Plano de aula 1 – Funcionamento do objeto de aprendizagem amnesia

**Público alvo**

Alunos sem um conhecimento prévio do Objeto de Aprendizagem (OA) Amnesia.

**Motivação do Plano de Aula**

O aluno não tem como utilizar o OA Amnesia adequadamente, sem saber como ele funciona. Esta aula é motivada pela necessidade de transmitir aos alunos as principais características do OA e suas funcionalidades.

**Objetivo Geral**

Ensinar aos alunos o OA Amnesia.

**Objetivos Específicos**

O aluno deverá ser capaz de:

1. Iniciar a execução do OA, identificar os seus principais componentes (outros módulos – processador/rastro, cache, memória virtual - menus, janelas para interação, arquivos de entrada/saída, modos de execução).
2. Conhecer o fluxo básico de execução, identificando, inclusive entradas inválidas para a simulação.
3. Identificar as principais informações inseridas no *log* de saída, contendo os acessos à memória.
4. Identificar as métricas usadas nas informações relativas ao desempenho.
5. Executar simulações simples no OA Amnesia.
6. Criar os seus próprios arquivos de arquitetura em XML e arquivos *de rastro.*

**Desenvolvimento**

A aula será desenvolvida com atividades práticas usando o OA Amnesia em um computador. Cada atividade descrita a seguir aborda um ou mais objetivos a serem atingidos.

1. Operações básicas no Amnesia

[Atividades\Operações Basicas.docx](Atividades/Operações%20Basicas.docx)

1. Geração do arquivo de Arquitetura em XML

(a) **Motivação**: Essa atividade apresenta quais as informações devem estar presentes no arquivo XML para cada componente, fazendo que informações corretas gerem uma arquitetura válida para ser aceita pelo OA Amnesia.

(b) **Objetivo**: Fazer com que os alunos entendam como a arquitetura é descrita no arquivo XML, para que futuramente eles possam criar as suas próprias arquiteturas.

(c) **Detalhamento**: Será aberto um arquivo de arquitetura (02-2) em um editor de texto, onde serão apresentadas as seguintes informações:

* + 1. Cabeçalho do arquivo XML, que deve ser igual em todos os arquivos XML carregados pelo OA Amnesia

<?xml version="1.0" encoding="ISO-8859-1"?>

<!DOCTYPE AmnesiaConfiguration SYSTEM "Configuration/amnesia.dtd">

<?xml-stylesheet type="text/css" href="teste.css"?>

* + 1. Início da descrição da arquitetura com a tag <AmnesiaConfiguration>
    2. Apresentar as características do processador que estaria acessando a hierarquia de memória. Tais acessos podem ser feitos por um arquivo de rastro, resultante da execução em um processador. Mesmo usando tal arquivo de rastro, as características do processador precisam ser fornecidas. Tais características devem ser mantidas no arquivo XML.

<Processor>

<processorContains>0</processorContains>

<createTraceFile>0</createTraceFile>

</Processor>

* + 1. Informações de tamanho da palavra em BYTES. Os valores devem ser iguais nas duas tags <wordSize>.

<Trace>

<wordSize>4</wordSize>

</Trace>

<CPU>

<wordSize>4</wordSize>

</CPU>

* + 1. Informações da memória principal: Informações de tamanho de bloco, tamanho da memória, tempo de ciclos entre outros.

<MainMemory>

<blockSize>1</blockSize>

<memorySize>16</memorySize>

<ciclesPerAccessRead>1</ciclesPerAccessRead>

<ciclesPerAccessWrite>2</ciclesPerAccessWrite>

<timeCicle>10</timeCicle>

</MainMemory>

* + 1. Informações da memória virtual que são: tamanho de páginas em quantidade de palavras, tamanho do disco em quantidade de palavras e tempo para acessos à leitura e escrita.

<VirtualMemory>

<pageSize>4</pageSize>

<diskMemorySize>24</diskMemorySize>

<diskCiclesPerAccessRead>1</diskCiclesPerAccessRead>

<diskCiclesPerAccessWrite>2</diskCiclesPerAccessWrite>

<timeCicle>100</timeCicle>

<pageTableReplacementAlgorithm>FIFO</pageTableReplacementAlgorithm>

<TLBType>none</TLBType>

</VirtualMemory>

* + 1. Finalizar a configuração com a tag:

</AmnesiaConfiguration>

* + 1. Importante salientar que para todas as arquiteturas, as configurações dos parâmetros *ciclesPerAccessRead*, *ciclesPerAccessWrite* e *timeCicle* serão constantes conforme a seguir:

|  |  |  |  |
| --- | --- | --- | --- |
| **Memória / Configuração** | **ciclesPerAccessRead** | **ciclesPerAccessWrite** | **timeCicle** |
| **Cache** | 1 | 2 | **1** |
| **Principal** | 1 | 2 | **10** |
| **Disco** | 1 | 2 | **100** |
| **TLB** | 1 | 2 | **1** |

* + 1. Aplicar mudanças na arquitetura para que o aluno crie seu próprio arquivo de arquitetura. Sugestões de mudanças são:
       1. Mudar o tamanho da memória principal de 16 para 32.
       2. Mudar o tamanho da página de 4 para 8.
       3. Mudar o tamanho do disco de 24 para 16.
    2. Salvar arquivo com as mudanças realizadas dar um nome para a nova arquitetura.
    3. Carregar nova arquitetura e observar mudanças na interface do OA.

(d) **Arquivos**

(i) **Arquitetura**: Architecture-02-2-MM-16-VM(PS-4-DM-4-RA-FIFO)-TLB(none)

(e) **Pontos a destacar:**

(i) É esperado que o aluno entenda como descrever um arquivo de arquitetura XML, além disso, incentivar que o aluno busque informações de como descrever as informações de outras memórias que podem ser descritas no OA Amnesia como Cache e TLB.

1. Geração do arquivo de rastro
   1. **Motivação**: Os arquivos de rastro contêm a relação cronológica de acessos à memória em uma execução anterior. Tais arquivos são usados para determinar a ordem os acessos devem ocorrer, simulando o uso da hierarquia de memória no Amnesia. Os alunos precisam saber como os arquivos de rastro são organizados de modo a usá-los adequadamente.
   2. **Objetivo**: Fazer com que os alunos entendam como os acessos à memória são descritos no arquivo de rastro, para que futuramente eles possam criar os seus próprios arquivos de rastro.
   3. **Detalhamento**: Será aberto o arquivo de rastro (6) em um editor de texto, onde serão apresentadas as seguintes informações:
      1. Cada linha do arquivo de rastro é referente a 1 acesso à memória.
      2. Cada linha é composta por dois números (tipo de acesso e endereço a ser acessado) o que vem após isso e ignorado como comentário.
      3. **(se não apresentado na primeira atividade)** Os tipos de acessos possíveis são:
         1. “0”: Leitura de dados;
         2. “1”: Gravação de dados;
         3. “2”: Busca de instrução;
      4. Os endereços estão em hexadecimal.
      5. Verificar que o acesso ao endereço mais alto, representa o tamanho do processo.
         1. Supondo que há um acesso de leitura ao endereço f (2 f).
         2. O processo terá 16 palavras (f em decimal é 15. O processo é de 0 até 15 dados)
         3. O tamanho da memória deve ser no mínimo do tamanho do processo (ex: 16 palavras).
         4. O tamanho total de memória é o tamanho total da RAM mais o tamanho total do disco.
   4. **Arquivos:**
      1. Rastro: TR\_6\_read\_and\_write\_30\_rand\_PS\_24
   5. **Pontos a destacar:**
      1. Espera-se que o aluno entenda como descrever um arquivo de rastro.
2. Arquitetura com entradas inválidas
   1. **Motivação:** verificar que há restrições em relação os valores de tamanho de cada componente. Deve-se evitar valores que possam causar erro na construção arquitetura.
   2. **Objetivo:** Com essa tarefa é possível notar que há algumas restrições de valores para descrever as memórias presentes em cada arquitetura, além de observar que algumas combinações de valores geram erros.
   3. **Detalhamento:** O aluno irá carregar uma arquitetura (**with-errors**) com erros implantados propositalmente, para no momento que os alunos tentarem carregar a arquitetura os erros são apresentados e corrigidos. As sequências de erros são:
      1. Tamanho de palavra nas tags <Trace> e <CPU> são diferentes
      2. Tamanho da memória principal não é múltipla do tamanho de páginas
      3. O Tamanho do disco não é múltiplo do Tamanho da página
      4. Política de substituição não é uma entrada válida
      5. Tamanho da página não é base 2. Após essa correção indicar que há outro erro e solicitar que os alunos identifiquem esse erro. O erro em questão é que o tamanho da memória principal não será múltiplo do tamanho da página.
      6. Após essas correções a arquitetura irá carregar corretamente.
      7. (Opcional) Abrir novamente a arquitetura e deixar os alunos injetarem erros na arquitetura e verificar quais os resultados obtidos com o carregamento da arquitetura.
   4. Arquivos:
      1. Arquiteturas: ***Architecture-with-errors*.**
   5. **Pontos a destacar:**
      1. É esperado com essa atividade que os alunos entendam como construir as suas próprias arquiteturas, evitando os valores que possam causar erros na arquitetura.
      2. Não é necessário realizar uma execução na arquitetura.
3. Arquivo de Rastro com entradas inválidas
   1. **Motivação:** Verificar que há algumas restrições em relação aos valores de tamanho de cada componente. Tentar evitar valores que possam causar erro na construção da arquitetura.
   2. **Objetivo:** Ressaltar que há restrições de valores para os tipos de acesso, além de observar os valores inválidos para o arquivo.
   3. **Detalhamento:** O aluno irá abrir um arquivo de rastro (***with-errors***) com erros implantados propositalmente, para no momento que os alunos tentarem carregar o arquivo de rastro, os erros sejam apresentados e corrigidos. As sequências de erros são:
      1. Um comentário em uma posição errada do arquivo de rastro
      2. Um valor de **tipo de acesso** inválido
      3. Endereço com valor negativo
      4. Um endereço hexadecimal inexistente (g)
      5. Endereço maior que tamanho total da memória.
      6. Após essas correções o arquivo de rastro deve carregar corretamente.
      7. (Opcional) Abrir novamente o arquivo de rastro e deixar os alunos injetarem erros no arquivo e verificar quais os resultados obtidos com o carregamento do arquivo de rastro.
      8. Após as correções realizar uma execução direta e verificar o resultado.
   4. Arquivos
      1. **Arquiteturas:** Architecture-with-errors - com as correções realizadas.
      2. **Rastro:** Trace\_with\_error
   5. **Pontos a destacar:**
      1. É esperado com essa atividade que os alunos entendam como construir os seus próprios arquivos de rastro, evitando os valores que podem causar erros.
      2. Realizar a execução somente para verificar que os erros foram removidos.
4. **Exercício**: Desenvolver um arquivo de arquitetura e um arquivo de rastro para realizar uma simulação no OA Amnesia, após finalizar a simulação realizar uma coleta dos tempos das memórias presentes na arquitetura.
   1. O arquivo de arquitetura deve ter:
      1. Tamanho de palavra 4
      2. Memória Principal de tamanho 16 palavras
      3. Tamanho de bloco 1 palavra.
      4. Memória Virtual com tamanho de página 16
      5. Tamanho de disco 32 palavras.
      6. Com Política de substituição LRU.
      7. Sem TLB.
   2. O arquivo Trace deve conter:
      1. Acessos de Leitura aos endereços pares de 0 a 1C.
      2. Acesso de Escrita aos endereços impares de 0 a 1f.
   3. Realizar a simulação direta e coletar os dados de:
      1. Read Time, Write Time e Total Time nas estatísticas: Memória RAM, Disk e page Table.
      2. Total Time da simulação.
      3. Escrever situação final da tabela de páginas.
   4. **Objetivo**: Fazer com que os alunos fixem o o funcionamento do AO Amnesia para que nas próximas aulas não restem dúvidas de como operar a ferramenta, criar arquivos de arquitetura e arquivos de rastro.

**Síntese**

Essa aula apresentou a interface do OA Amnesia com o usuário, onde são realizadas as simulações de hierarquia de memória e memória virtual. Também foi apresentado como descrever uma arquitetura no arquivo XML e como é descrito um arquivo de rastro. Além disso, foram apresentados possíveis defeitos nos arquivos de arquitetura e nos arquivos de rastro e como solucionar esses defeitos.

**Recursos**

Os recursos necessários para esta aula são computadores com o OA Amnesia e um projetor multimídia.

**Avaliação**

Não será realizada nenhuma avaliação após a aula.

**Referências Bibliográficas**

Hill, M. D. (1989). dinero - cache simulator, version III (Enhanced Version). http://www.ece.cmu.edu/~ece548/tools/dinero/src/doc.h, [accessed on May 25].