



Universidade do Vale do Itajaí

Curso: Bacharelado em Ciência da Computação

Professor: Felipe Viel

Disciplina: Sistemas Operacionais

Avaliação – Escalonamento e Memória

Acadêmicos: Alexandre Machado de Azevedo,

Matheus Passold Carelli,

Vinícius dos Santos Moreira.

Itajaí, Setembro de 2023

Projeto

Suponha que um sistema tenha um endereço virtual de tamanho entre 16 bits à 32 bits com deslocamento na página de 256 b à 4 Kb. Escreva um programa que receba um endereço virtual (em decimal) na linha de comando ou leitura do arquivo addresses.txt faça com que ele produza o número da página e o deslocamento do endereço fornecido, sendo que essa posição indica qual a posição que será lido do arquivo data_memory.txt. Você irá encontrar esses arquivos no github da disciplina, mais especificamente na pasta Memory (link repositório).

Por exemplo, seu programa seria executado da seguinte forma:

```
./virtual_memory_translate.exe 19986
```

ou

```
./virtual_memory_translate.exe addresses.txt
```

Seu programa produzirá:

- O endereço 19986 contém:
 - o número da página = 4
 - o deslocamento = 3602
 - o Valor lido: 50 (exemplo)

No caso, o número em binário é 0100 1110 0001 0010, sendo que 0100 diz respeito à página e 1110 0001 0010 diz respeito ao deslocamento na página. Para manipular os números em nível de bit, é recomendado usar os operadores bitwise (bit-a-bit) da linguagem escolhida.

No caso o exemplo apresentado é para 16 bits. No caso de 32 bits, haveriam mais 16 bits a esquerda (mais significativo) referentes ao número de páginas.

Escrever este programa exigirá o uso do tipo de dados apropriado para armazenar 16 à 32 bits (short ou int). É recomendado que você também use tipos de dados sem sinal. Além disso, para endereços de 32 bits deve ser possível usar paginação hierárquica de 2 níveis mantendo 4 Kb, com cada nível tendo 10 bits de tamanho.

Resolução:

O código foi implementado em JavaScript, sendo criado diversas funções com o intuito de ser possível que seja retornado o número de páginas, o deslocamento e o valor correspondentes ao endereço de memória passado na execução do programa. Foram criados para transformar números decimais em binários e ser possível identificar o local de memória correspondente ao número decimal informado. Foi criado também uma função para a leitura de arquivos txt com valores de endereço de memória, além de funções para transformação e leitura dos binários a partir do número decimal para ser possível obter o valor da página, do deslocamento e do valor.

Resultados:

Endereço de 16bits:

```
xandeturf@Speed:~/sistemas_operacionais/trabalhos/m2/javascript$ node script.js endereco=19986
[
  {
    "enderecoSolicitado": "19986",
    "numero_paginas": 4,
    "numero_deslocamento": 3602,
    "numero": "65"
  }
]
```

Endereço de 32bits:

```
xandeturf@Speed:~/sistemas_operacionais/trabalhos/m2/javascript$ node script.js endereco=429986
[
  {
    "enderecoSolicitado": "429986",
    "numero_paginas": 104,
    "numero_deslocamento": 4002,
    "numero": "82"
  }
]
```

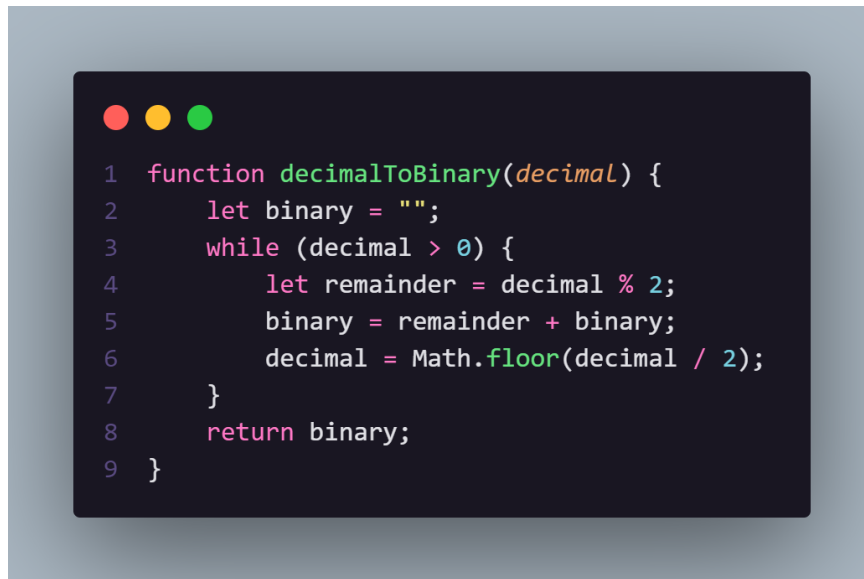
Endereço de 32bits com hierarquia de 2 níveis:

```
xandeturf@Speed:~/sistemas_operacionais/trabalhos/m2/javascript$ node script.js endereco=5194304 hierarquia=true
hierarquiaAtivada!
[
  {
    "enderecoSolicitado": "5194304",
    "numero_paginas_nivel1": 1,
    "numero_paginas_nivel2": 244,
    "numero_deslocamento": 576,
    "valorLido": "27"
  }
]
```

Códigos:

```
1 function localizarEndereco(enderecoSolicitado, tamanhoDaPagina, eh32bits) {
2   let enderecoBinario = decimalToBinary(enderecoSolicitado).padStart(eh32bits ? 32 : 16, 0);
3
4   let numero_paginas = 0;
5   let numero_deslocamento = 0;
6   let tratamentoBinarioDeslocamento = Math.log2(tamanhoDaPagina);
7
8   for (const numero of numeros_decimais) {
9     var binario = '';
10    if (eh32bits) {
11      binario = decimalToBinary(numero_paginas).padStart(32 - tratamentoBinarioDeslocamento, 0);
12    } else {
13      binario = decimalToBinary(numero_paginas).padStart(16 - tratamentoBinarioDeslocamento, 0);
14    }
15    binario += decimalToBinary(numero_deslocamento).padStart(tratamentoBinarioDeslocamento, 0);
16
17    if (enderecoBinario == binario) {
18      return {
19        enderecoSolicitado,
20        numero_paginas,
21        numero_deslocamento,
22        numero
23      };
24    }
25    numero_deslocamento++;
26    if (numero_deslocamento >= tamanhoDaPagina) {
27      numero_paginas++;
28      numero_deslocamento = 0;
29    }
30  }
31  return false;
32 }
```

Nessa função ele realiza as validações de entrada do endereço que foi requisitado, para a partir desse momento decidir utilizar a função de conversão de decimal para binário e então validar os valores respectivos a cada informação necessária.



```
1 function decimalToBinary(decimal) {  
2     let binary = "";  
3     while (decimal > 0) {  
4         let remainder = decimal % 2;  
5         binary = remainder + binary;  
6         decimal = Math.floor(decimal / 2);  
7     }  
8     return binary;  
9 }
```

Essa função se trata da conversão do endereço que foi passado por parâmetro na execução do código que está na forma decimal, para sua representação em binário. Sendo assim possível de realizar os cálculos necessários para obter as páginas, o deslocamento e o valor necessário.

Resultados finais:

Pode se observar que o código escrito teve resultados esperados para a resolução do problema. Foi possível visualizar a página, o deslocamento e o valor do endereço corretos, utilizando diversos parâmetros que deveriam ser possíveis de serem alterados. Dentro os parâmetros possíveis, o tamanho da página poderia ser de 256b a 4Kb e em todas as execuções os valores foram calculados de maneira correta. Além disso, o valor do endereço de 16 bits a 32 bits também era possível de ser modificado e ainda sim o código foi executado corretamente. As conversões de decimal para binário e vice-versa funcionaram corretamente e permitiram a precisão necessária para a pesquisa. O objetivo principal do projeto foi alcançado.