

# Simulação do Algoritmo WSClock

Felipe Artur Macedo Lima

Guilherme Martinho Chumbinho de Andrade

## 1. Detalhamento da atividade

- Implementação em C/C++ ou Python
- Configuração Inicial:
  - Número de Frames de Memória a ser considerado
- Para cada Iteração, informar:
  - Tipo de Evento:
    1. (R) Referenciação de Página
      - Sistema pede o número da página referenciada;
    2. (S) Página a ser Substituída
      - Sistema pede o número da nova página carregada na memória;
    3. (W) Término do período de trabalho
      - Sistema pede o tempo do evento;
    4. (E) Exibição da Lista Circular com Atributos das Páginas
    5. (F) Fim da Simulação

## 2. Explicação do código

- BliBlio\_WSClock.h

```
5 #include <locale.h>
6 // definicao do working set (delta)
7 #define Delta 3
8
9 //Struct da pagina com seus parametros
10 struct pagina
11 {
12     unsigned int id;
13     unsigned int referencia;
14     bool R;
15 };
16 typedef struct pagina Pagina;
17 //Struct do no que vai servir de apoio para apontar durante o código
18 struct no
19 {
20     Pagina informacao;
21     struct no *prox;
22 };
23 typedef struct no No;
24
25 typedef No *Clock;
26
27 unsigned int tempo = 1;
28 // Função de inicializar Clock
29 void inicializar_Clock(Clock *c)
30 {
31     *c = NULL;
32 }
33
34 // gera um numero inteiro aleatorio entre 0 e o limite passado
35 unsigned int aleatorio(unsigned int x)
36 {
37     return rand() % x;
38 }
39
40 // retorna o achar_tempo atual e o incrementa por um numero aleatorio entre 1 e 4
41 unsigned int achar_tempo()
42 {
43     return tempo += aleatorio(5) + 1;
44 }
```

Figura 1

Como pode ser observado na figura 1, o código foi começado primeiro criando um struct para página e escrevendo seus parâmetro sendo eles id, referencia e R, e depois criamos outro struct chamado de nó, e ele vai servir de auxílio durante o código para apontar para valores na lista. Depois foi criado a variável global tempo que vai servir como tempo valor que vai ser colocado no tempo de referência da página ao longo do código, com essa variável criado criamos mais duas funções que uma gera um número inteiro aleatório entre 0 e limite passado para então ser usado na função de achar o tempo onde retornamos a soma do tempo com o retorno da função de achar um número aleatório somado com 1, e por fim criamos a função de inicializar a lista encadeada.

```
// Função de referenciar a pagina e quando a função terminar muda o bit R de 0 para 1
void referenciar_pag(Clock *c, unsigned int id)
{
    Clock aux = *c;
    // Utiliza o parametro id para encontrar a pagina desejada
    while (aux->informacao.id != id)
        aux = aux->prox;

    aux->informacao.R = true;
    aux->informacao.referencia = achar_tempo();
}

// Função de printar o Clock
void printar_Clock(Clock c)
{
    printf("PONTEIRO->(id: %d; ref: %d, R:%d)", c->informacao.id, c->informacao.referencia, c->informacao.R);
    Clock aux = c->prox;
    //Loop para passar por todas as paginas do clock
    while (aux != c && aux != NULL)
    {
        printf(", (id: %d; ref: %d, R:%d)", aux->informacao.id, aux->informacao.referencia, aux->informacao.R);
        aux = aux->prox;
    }
    printf("\n");
}
```

Figura 2

Aqui criamos duas funções, uma sendo a de referenciar a página selecionada que recebe como parâmetro o id da pagina, que é utilizada para encontrar a página solicitada e quando ela é achado o seu bit R tem seu valor trocado para true(1) e seu tempo é atualizado. A segunda função acima serve para printar a lista circular, nela somente precisa passar qual a lista que deseja printar que ela percorre essa lista do começo ao final printando todos os valores como o id, referência e o bit R em ordem do começo ao final da lista.

```

// Função de substituir pagina do WSClock
void substituir_pagina(Clock *c)
{
    unsigned int tempo = achar_tempo();
    unsigned int diferenca = tempo - Delta;
    Clock aux = *c;
    //Procura o frame que vai ser substituido
    do
    {
        if (aux->informacao.R)
            aux->informacao.R = false;
        else if (aux->informacao.referencia < diferenca)
        {
            //encontrou ele
            aux->informacao.referencia = tempo;
            aux->informacao.R = true;
            printf("pagina %d trocada\n", aux->informacao.id);
            *c = aux->prox;
            return;
        }
        aux = aux->prox;
    } while (aux != *c);

    // Não substituiu nenhum frame então ele substituiu o primeiro frame
    aux->informacao.referencia = tempo;
    aux->informacao.R = true;
    *c = aux->prox;
    printf("pagina %d trocada\n", aux->informacao.id);
}

```

Figura 3

Na figura 3 temos a função de substituir a página do WSClock, nela precisamos saber apenas qual a lista, e então criamos duas variáveis sendo uma delas o tempo atual e outra a diferença entre o tempo atual e o delta, ao saber esses dois valores o ponteiro percorre a lista até achar a frame que foi solicitado para substituir e caso não substitua nenhum frame ele volta para o primeiro frame da lista e o substitui trocando o valor de seu bit R para true(1) e atualizando seu tempo.

```

107 // Função de inserir uma pagina no clock
108 void inserir_pagina(Clock *c, unsigned int id, unsigned int Referencia)
109 {
110     No *aux_no = (No *)malloc(sizeof(No));
111     //Parametro que ele recebe e inseri dados
112     aux_no->informacao.id = id;
113     aux_no->informacao.referencia = Referencia;
114     aux_no->informacao.R = false;
115
116     // Clock estah vazio
117     if (*c == NULL)
118     {
119         aux_no->prox = NULL;
120         *c = aux_no;
121     }
122     // Clock tem apenas um elemento
123     else if ((*c)->prox == NULL)
124     {
125         // Insere no final da lista
126         aux_no->prox = *c;
127         (*c)->prox = aux_no;
128     }
129     // Clock tem mais de 1 elemento
130     else
131     {
132         Clock aux = (*c)->prox;
133         // Vai ate o ultimo elemento da lista
134         while (aux->prox != *c)
135             aux = aux->prox;
136
137         // Insere no final da lista
138         aux_no->prox = *c;
139         aux->prox = aux_no;
140     }
141 }

```

Figura 4

A função de inserir a página mostrada acima funciona da maneira que recebe o id e a referência digitada pelo usuário e os adiciona na lista circular junto com seu bit R que sempre começa com o valor de falso(0). Para saber onde inserir isso na lista circular ele percorre a lista para ver se ela está vazia ou já está populada, caso esteja vazia ele insere esses valores na primeira posição, mas caso já possua valores inseridos ele percorre a lista até achar o último membro e insere o novo valor depois dele ou seja no final da lista.

## ○ Main.cpp

```
1  #include "BliBlio_WSClock.h"
2  //Equipe: Equipe: Guilherme Marinho, Felipe Lima
3  //funções para printa das escolhas, tela inicial, e tela final
4  void printar_opcao() {
5      printf("Escolha uma opcao:\n(R) Referenciação de Página \n(S) Página a ser Substituída \n(W) Término do periodo de trabalho\n");
6  }
7  void printar_tela_final() {
8      printf("Trabalho feito\n");
9  }
10 void print_tela_inicial() {
11     printf("Trabalho de WSClock\nEquipe: Guilherme Martinho e Felipe Lima\n");
12 }
```

Figura 5.a

```
1
2
3
4
5  do periodo de trabalho\n(E) Exibição da Lista Circular com Atributos das Páginas\n(F) Fim da simulacao\n");
6
7
8
9
10
11
12
```

Figura 5.b

As imagens acima apresentam três funções que foram criadas para a facilitação de entendimento para o usuário. A Figura 5.b é a continuação horizontal da Figura 5.a.

```
14 int main() {
15     //função para poder escrever acentos da lingua portuguesa
16     setlocale(LC_ALL, "Portuguese");
17
18
19     Clock c;
20     //Criação de variáveis que vou utilizar na main
21     unsigned int x, numero_paginas, tempo_Ref;
22     //Inicilizei a lista encadeada CLock
23     inicializar_Clock(&c);
24     //Printando a tela inicial
25     print_tela_inicial();
26     //Funções de pausar a tela e limpar ela dps de ler a mensagem inicial
27     system("pause");
28     system("cls");
29     //Pede a quantidade de frames que vai fazer para passar como parametro para função de inserir_pagina
30     printf("Escreva a quantidade de frames que voce deseja(A quantidade preceisa ser maior do que 1): ");
31     scanf("%d", &numero_paginas);
32     //loop para repetir a inserção de frames na pagina ate a quantidade de frames escolhida seja cumprida
33     for (x = 0; x < numero_paginas; x++) {
34
35         printf("Digite o tempo desejado:");
36         scanf("%d", &tempo_Ref);
37         inserir_pagina(&c, x, tempo_Ref);
38     }
```

Figura 6

Primeiramente, o Clock foi alocado como “c” e as variáveis iniciais foram declaradas. A partir disso, a lista encadeada CLock foi inicializada e a tela principal será apresentada ao usuário, em seguida, a quantidade de frames que será passada como parâmetro para função de inserir\_pagina é coletada. Por fim, um loop foi feito

para repetir a inserção de frames na página ate que a quantidade de frames escolhido seja cumprida.

```
39 //variaveis auxiliares para o uso do switch case
40 unsigned int auxiliar1;
41 char opcao;
42 //Switch case da tela de opção onde recebe a opção escolhida
43 while (true)
44 {
45     //Chama a função de printar opções
46     printar_opcoes();
47     scanf(" %c", &opcao);
48     //Se a opção digitada for R ele entra nesse if e realizar a referencia de pagina
49     if (opcao == 'R')
50     {
51         printf("Referenciação de Página:");
52         printf("Digite a pagina (numero de 0 a %d): ", numero_paginas - 1);
53         scanf(" %d", &auxiliar1);
54         referenciar_pag(&c, auxiliar1);
55         printf("Pagina %d referenciada\n", auxiliar1);
56     }
57     //Se a opção digitada for S ele entra nesse else if e realizar a Substituição de pagina
58     else if (opcao == 'S')
59     {
60         printf("Página a ser Substituída:");
61         substituir_pagina(&c);
62     }
```

Figura 7

Primeiramente, uma variável inicial para o uso do switch case foi alocada, em seguida, as opções são apresentadas para o usuário. A partir da escolha das letras: R, S, W, E e F ações diferentes ocorrem. No tocante às seleções:

- **R:** O código entra em um if e realiza o processo de referenciação da página;
- **S:** código entra em um else if e realiza a Substituição de página;

```
63 //Se a opção digitada for W ele entra nesse else if ele mostra o termino do periodo de trabalho
64 else if (opcao == 'W')
65 {
66     printf("Término do período de trabalho:");
67     printf("achar_tempo atual: %d\n", tempo);
68 }
69 //Se a opção digitada for E ele entra nesse else if ele printa a lista circular com todos os atributos da pagina
70 else if (opcao == 'E')
71 {
72     printf("Exibicao da lista circular com atributos das paginas ");
73     printar_Clock(c);
74 }
75 //Se a opção digitada for F ele entra nesse else if ele finaliza o codigo e sai do switch case
76 else if (opcao == 'F')
77 {
78     break;
79 }
80 //Caso a pessoa digite uma opção incorreta ele exibe essa mensagem de erro e volta pra tela de escolher a opção
81 printf("Erro\Opção digitada não existe!!!\n");
82 }
83 system("pause");
84 printf("*****\n");
85 }
86 //Limpa a tela e printa a tela final
87 system("cls");
88 printar_tela_final();
89 return 0;
90 }
```

Figura 8

- **W:** O código entra em um else if e realiza o processo de termino do período de trabalho;

- **E:** O código entra em um else if e imprime a lista circular com todos os atributos da página;
- **F:** O código entra em um else if e finaliza o código e sai do switch case;
- **Outro:** Caso a pessoa digite uma opção incorreta ele exibe uma mensagem de erro e volta pra tela de escolher a opção;

Por fim, a função `printar_tela_final()` é executada e o programa é finalizado.