

pdfpages

# Simulação e algoritmos de semáforo, para o problema “Leitores e escritores”

Victor Emanuel Almeida      Levi Cícero Arcanjo

13 de outubro de 2020

## 1 Algoritmo

Segue abaixo a implementação de um algoritmo para a resolução:

```
#include <iostream>
using namespace std;

typedef int semaphore;

int n_readers = 0;

semaphore A = 1;
semaphore B = 1;
```

Figura 1: Criação das variáveis e inicializações

Nesse primeiro trecho de código, temos a criação e inicialização das variáveis, tanto a variável que armazena o número de leitores ativos, quanto os semáforos A e B, que tem função de proteger a variável “n\_readers” e impedir a escrita enquanto alguém estiver lendo ou escrevendo respectivamente.

```
void down(int &_semaphore){
    while(_semaphore == 0);
    _semaphore = _semaphore - 1;
}

void up(int &_semaphore){
    if(_semaphore == 0)
        _semaphore = _semaphore + 1;
}
```

Figura 2: Implementação das funções up e down

- Função down: Mantém o processo em uma espera ocupada enquanto a variável do semáforo é igual a zero, caso contrário a decrementa.
- Função up: incrementa o valor do semáforo caso seja possível.

```
void writer(){  
    down(B);  
    write();  
    up(B);  
}
```

Figura 3: Implementação do escritor

Como podemos ver na figura acima o escritor espera até ser possível realizar o `down()`, quando concluído ele pode realizar a escrita, pois não existe nenhum outro processo lendo nem escrevendo.

```
void reader(){  
    down(A);  
    n_readers++;  
    if(n_readers == 1){  
        down(B);  
    }  
    up(A);  
  
    read();  
  
    down(A);  
    n_readers--;  
    if(n_readers == 0){  
        up(B);  
    }  
    up(A);  
}
```

Figura 4: Implementação do leitor

Por fim temos a implementação do leitor, o qual utiliza dois semáforos, sendo o A para modificar a variável “`n_readers`”, e o B para leitura.

## 2 Simulação

Agora vamos realizar a simulação do algoritmo supracitado, com os seguintes dados:

- Tempo de criação:
  - leitor 1 = momento 10
  - leitor 2 = momento 4
  - leitor 3 = momento 1
  - escritor 1 = momento 14
  - escritor 2 = momento 9
- Tempo de execução:
  - leitor 1 = momento 8
  - leitor 2 = momento 7
  - leitor 3 = momento 15
  - escritor 1 = momento 6
  - escritor 2 = momento 5