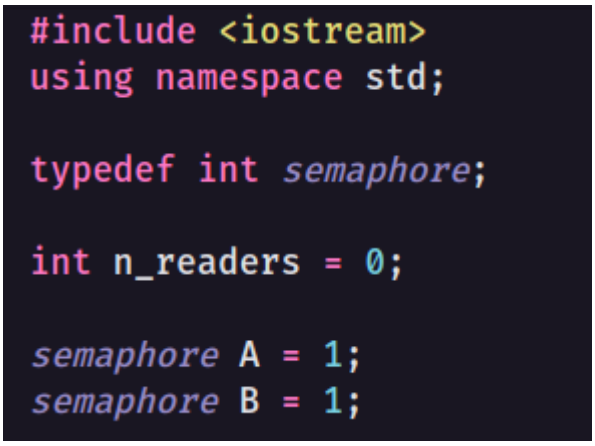
Simulação e algorítimos de semáforo, para o problema “Leitores e escritores”

Victor Emanuel Almeida Levi Cícero Arcanjo

13 de outubro de 2020

1 Algorítimo

Segue abaixo a implementação de um algorítimo 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 algorítimo 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

	leitor 1	leitor 2	leitor 3	escritor 1	escritor 2	semáforo A	semáforo B	n_leitores
momento 0						1	1	0
momento 1			down(A)			0	1	0
momento 2			n_leitores++			0	1	1
momento 3			up(A)			1	1	1
momento 4		down(A)	down(B)			0	0	1
momento 5		n_leitores++				0	0	2
momento 6		up(A)				1	0	2
momento 7						1	0	2
momento 8						1	0	2
momento 9					down(B)	1	0	2
momento 10	down(A)	read()				0	0	2
momento 11	n_leitores++					0	0	3
momento 12	up(A)		read()			1	0	3
momento 13						1	0	3
momento 14		down(A)		down(B)		0	0	3
momento 15		n_leitores--				0	0	2
momento 16	read()	up(A)				1	0	2
momento 17						1	0	2
momento 18					wait()	1	0	2
momento 19						1	0	2
momento 20			down(A)			0	0	2
momento 21	down(A)		n_leitores--			0	0	1
momento 22	wait()		up(A)			1	0	1
momento 23	down(A)			wait()		0	0	1
momento 24	n_leitores--					0	0	0
momento 25	up(B)					0	1	0
momento 26	up(A)				down(B)	1	0	0
momento 27						1	0	0
momento 28					write()	1	0	0
momento 29						1	0	0
momento 30						1	0	0
momento 31						1	0	0
momento 32					up(B)	1	1	0
momento 33				down(B)		1	0	0
momento 34						1	0	0
momento 35						1	0	0
momento 36				write()		1	0	0
momento 37						1	0	0
momento 38						1	0	0
momento 39						1	0	0
momento 40				up(B)		1	1	0

Figura 5: Simulação do algorítimo