

# Relatório Final: Exemplo 1 - Leitores e Escritores

---

**De: Gabriel Henrique da Silva**

**RA:22020864**

Essa parte do projeto foi feita em markdown

## Introdução

---

Este código resolve o problema de **Leitores e Escritores**, garantindo que:

- Múltiplos leitores possam acessar o recurso ao mesmo tempo.
- Apenas um escritor tenha acesso exclusivo ao recurso ao modificar.

## Código Explicado

---

### Variáveis e Mutexes

```
int data = 0;  
int read_count = 0;  
pthread_mutex_t mutex;  
pthread_mutex_t rw_mutex;
```

- `data` : Recurso compartilhado.
- `mutex` e `rw_mutex` : Controlam o acesso seguro.

### Função Escritor

```
void *writer(void *arg) {
    pthread_mutex_lock(&rw_mutex);
    data++;
    printf("Escritor alterou o valor para %d\n", data);
    pthread_mutex_unlock(&rw_mutex);
    return NULL;
}
```

- **Exclusivo:** Escritor bloqueia `rw_mutex` para garantir acesso exclusivo.

## Função Leitora

```
void *reader(void *arg) {
    pthread_mutex_lock(&mutex);
    read_count++;
    if (read_count == 1) pthread_mutex_lock(&rw_mutex);
    pthread_mutex_unlock(&mutex);

    printf("Leitor leu o valor %d\n", data);

    pthread_mutex_lock(&mutex);
    read_count--;
    if (read_count == 0) pthread_mutex_unlock(&rw_mutex);
    pthread_mutex_unlock(&mutex);
    return NULL;
}
```

- **Leitura Simultânea:** Vários leitores acessam `data` simultaneamente, controlados por `read_count` e `rw_mutex`.

## Função Principal

```
int main() {
    pthread_t readers[5], writers[3];
    pthread_mutex_init(&mutex, NULL);
    pthread_mutex_init(&rw_mutex, NULL);

    for (int i = 0; i < 5; i++) pthread_create(&readers[i], NULL, reader,
```

```
    for (int i = 0; i < 3; i++) pthread_create(&writers[i], NULL, writer,  
  
    for (int i = 0; i < 5; i++) pthread_join(readers[i], NULL);  
    for (int i = 0; i < 3; i++) pthread_join(writers[i], NULL);  
  
    pthread_mutex_destroy(&mutex);  
    pthread_mutex_destroy(&rw_mutex);  
    return 0;  
}
```

- **Inicialização e Execução:** Cria as threads leitoras e escritoras e aguarda sua conclusão.

## Conclusão

O código garante que vários leitores possam acessar simultaneamente e que escritores tenham acesso exclusivo ao recurso.