

Sistemas Operacionais: Editor de Texto

...

Cristiano Lima - Ilem Santos

Implementar um simples editor de texto que execute a seguinte funcionalidade

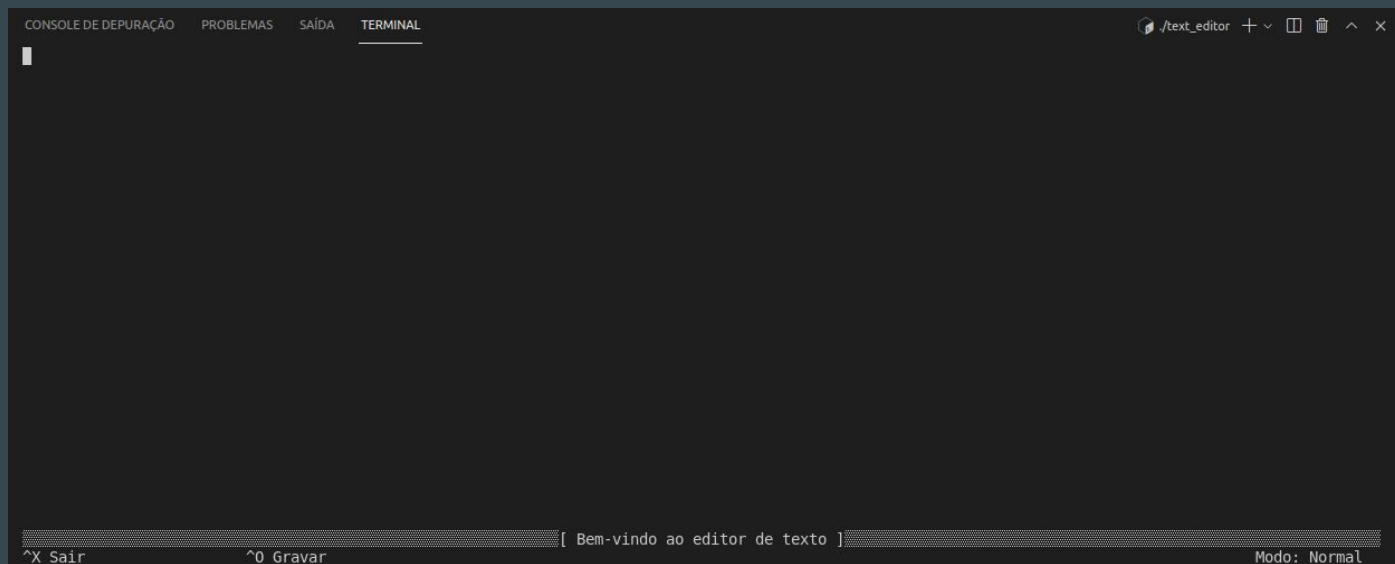
- Salvar somente números;
- Salvar somente letras;
- Salvar somente caracteres especiais;
- Salvar o número de linhas já escritos.

Utilize threads para identificar os itens a serem salvos e crie 4

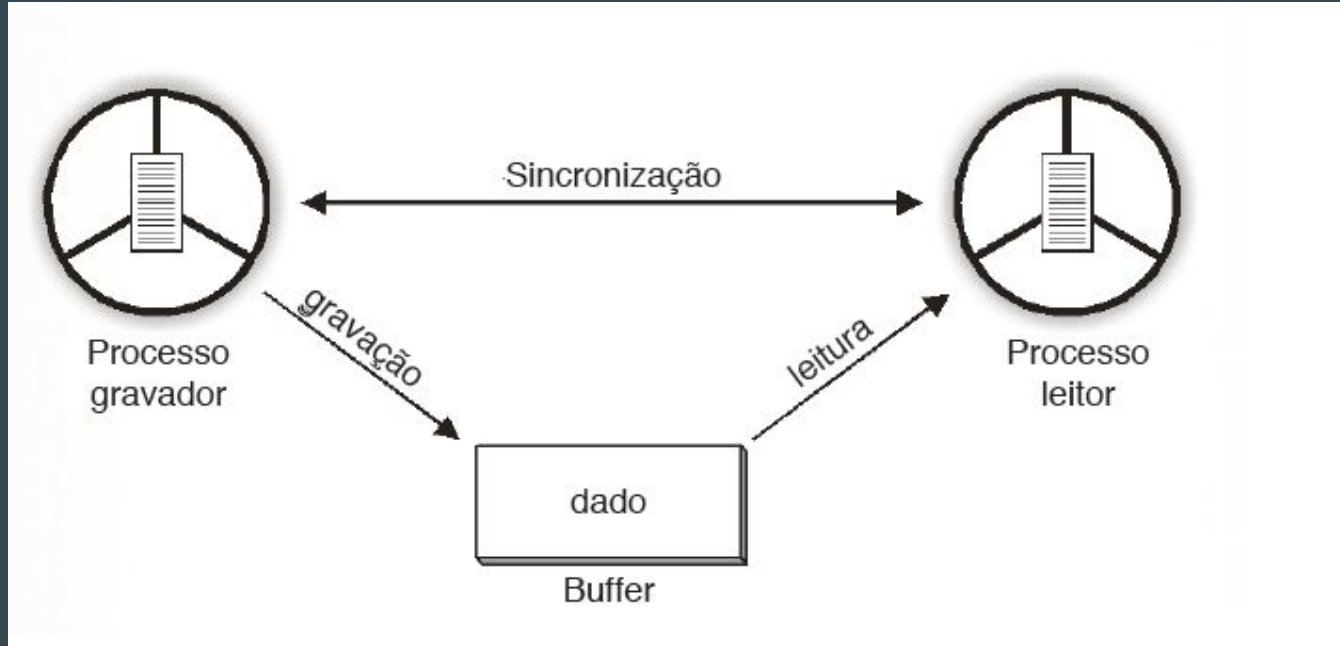
threads para alterar cada arquivo simultaneamente.

NCURSES

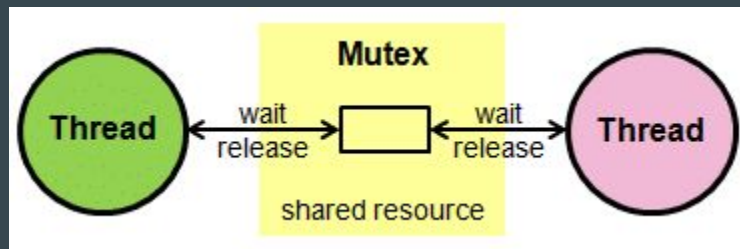
É uma biblioteca que provê uma API para o desenvolvimento de interfaces em modo texto.



Conflitos



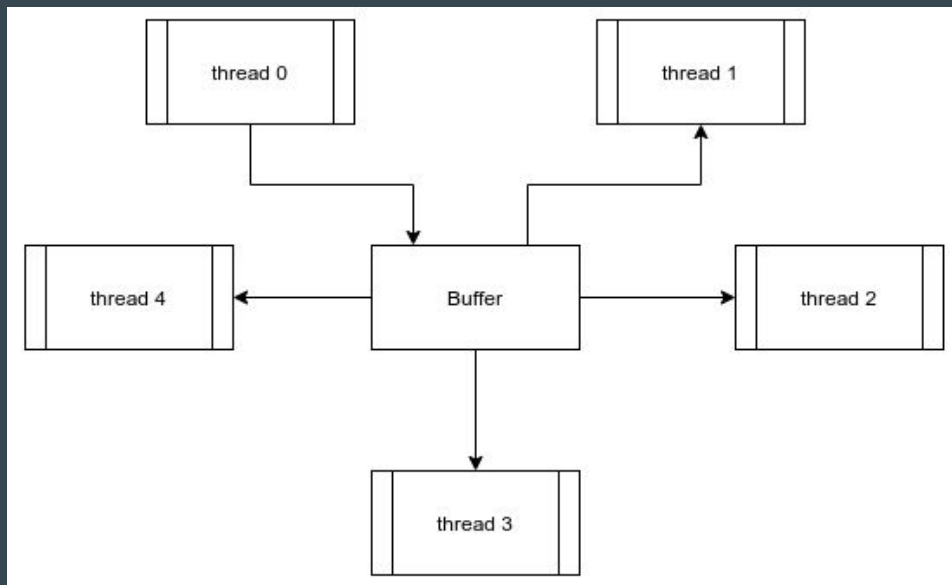
Exclusão mútua



Técnica utilizada em programação concorrente com a finalidade de evitar que dois processos tenham acesso simultaneamente a um recurso compartilhado, Chamada de seção crítica.

Semáforo

Variável utilizada para controlar o acesso a recursos compartilhados



```
typedef struct _rwlock_t {  
    sem_t lock_1;           /* trava binaria */  
    sem_t lock_2;           /* trava binaria */  
    sem_t wrt;              /* permite 1 escritor */  
    int readCount;          /* n de leitores */  
    pthread_mutex_t mutex;  
    pthread_cond_t cond_mutex;  
} rwlock_t;
```

```

1 void texteditor(texteditor_t *p) {
2
3     while (!exitflag) {
4
5         sem_wait(&p->ptr_semaphore->lock_2);
6         sem_wait(&p->ptr_semaphore->wrt);
7
8         /* [ Seção Crítica ] */
9
10        sem_post(&p->ptr_semaphore->wrt);
11        sem_post(&p->ptr_semaphore->lock_2);
12
13        pthread_mutex_lock(&p->ptr_semaphore->mutex);
14        pthread_cond_wait(&p->ptr_semaphore->cond_mutex, &p->ptr_semaphore->mutex);
15        pthread_mutex_unlock(&p->ptr_semaphore->mutex);
16    }
17
18    return 0;
19 }

```

```

1 void my_thread(texteditor_t *rw) {
2
3     while (state_condition) {
4         sem_wait(&rw->ptr_semaphore->lock_2);
5         sem_wait(&rw->ptr_semaphore->lock_1);
6         rw->ptr_semaphore->readCount++;
7         if (rw->ptr_semaphore->readCount == 1)
8             sem_wait(&rw->ptr_semaphore->wrt);
9         sem_post(&rw->ptr_semaphore->lock_1);
10        sem_post(&rw->ptr_semaphore->lock_2);
11
12        /* [ Seção Crítica ] */
13
14        sem_wait(&rw->ptr_semaphore->lock_1);
15        rw->ptr_semaphore->readCount--;
16        if (rw->ptr_semaphore->readCount == 0)
17            sem_post(&rw->ptr_semaphore->wrt);
18        sem_post(&rw->ptr_semaphore->lock_1);
19        pthread_cond_signal(&rw->ptr_semaphore->cond_mutex);
20    }
21 }

```

Implementação

