

Relatório Sistemas Operacionais II Exercício 1

Equipe: Gustavo Pinho Kretzer de Souza - 08232097
Rodrigo Boratti - 07132056
Ricardo Boschini Albuquerque Passarella – 09232045

Criamos uma fila de threads `_sleeping` na classe `Synchronizer_Common` para armazenar as thread que estarão dormindo.

`Thread::Queue _sleeping;`

O método `sleep()` coloca a thread para dormir, para fazer isso coloca a mesma em estado suspenso e a guarda numa fila de thread que estão dormindo. Esse método tem que ser atômico, para evitar problemas na execução caso o for escalonado outra thread durante a execução do método `sleep()`. Por exemplo, se antes de ser evocado o `Thread::running()` o escalonador trocar de thread, e ao invés de utilizarmos a thread na qual queremos colocar para dormir, pega a que está em execução no momento após a troca do escalonador. Então iremos suspender a thread errada, possivelmente criando um dead lock, ou algum outro estado inconsistente no sistema.

```
void sleep(){
    begin_atomic();
    Thread* running = Thread::running();
    _sleeping.insert(&running->_link);
    running->suspend();
    end_atomic();
}
```

Em `wakeup()` acorda uma thread que esteja dormindo, para acordar remove-se uma thread dormindo da fila de `sleeping` e troca-se o estado da thread com o método `resume`. Caso tiver algum thread na fila de `sleeping` ele acorda o primeiro da fila. Esse método é atômico, pois caso pode acontecer de ser testado se contem algum elemento na fila (`!_sleeping.empty()`), e o escalonador trocar a thread de execução e a fila se tornar vazia, e quando retornar a execução no bloco do `if` irá tentar remover um elemento que não existe, causando assim um erro.

```
void wakeup() {
    begin_atomic();
    if(!_sleeping.empty())
        _sleeping.remove()->object()->resume(); // end_atomic
    else
        end_atomic();
}
```

Em `wakeup_all()` acorda todas as thread que estão dormindo, ou seja todas que estão na fila de `sleeping`. Para acordar, até a fila de `sleeping` esteja vazia, remove a thread desta fila e a acorda chamando o `resume()`. Da mesma maneira que em `wakeup()` sempre antes de testar se a fila ainda contem elementos e a sua remoção devem ser operações atômicas.

```
void wakeup_all() {
    begin_atomic();
    while(!_sleeping.empty()) {
        _sleeping.remove()->object()->resume(); // end_atomic
        begin_atomic();
    }
    end_atomic();
}
```