Ruby 101

Advanced Ruby: Mutex

Advanced Ruby:

Mutex:

Provides a locking mechanism
Allows only one thread to lock

Allows data to be modified by the locking code block,
but keeps other code blocks locked out

Advanced Ruby:

Example program:

```
my_array = [0,0]
count = 0
my_string = ""
write_thread = Thread.new do
    while true do
        my_array[0] = count
        doubled = count * 2
        trippled = count * 3
        squared = count * count
        if doubled - trippled * squared < -1000
            print my_string
        end
        my_array[1] = doubled
        count += 1
    end
end
```

Advanced Ruby:

Example program:

```
...
get_thread = Thread.new do
    100000.times do
        elet_1 = my_array[0]
        elet_2 = my_array[1]
        if elet_1 * 2 != elet_2
            puts count
            raise "Threads out of sync"
        end
    end
end
get_thread.join
```

Advanced Ruby:

```
require 'thread'
my_mutex = Mutex.new
my_array = [0,0]
count = 0
my_string = ""
write_thread = Thread.new do
    while true do
        my_mutex.synchronize do
            my_array[0] = count
            doubled = count * 2
            trippled = count * 3
            squared = count * count
            if doubled - trippled * squared < -1000
                print my_string
            end
            my_array[1] = doubled
            count += 1
        end
    end
end
```

Advanced Ruby:

```
require 'thread'
my_mutex = Mutex.new
my_array = [0,0]
count = 0
my_string = ""
write_thread = Thread.new do
    while true do
        my_mutex.synchronize do
            my_array[0] = count
            doubled = count * 2
            trippled = count * 3
            squared = count * count
            if doubled - trippled * squared < -1000
                print my_string
            end
            my_array[1] = doubled
            count += 1
        end
    end
end
```

Ruby 101

Advanced Ruby:

```
...
get_thread = Thread.new do
    100000.times do
        my_mutex.synchronize do
            elet_1 = my_array[0]
            elet_2 = my_array[1]
            if elet_1 * 2 != elet_2
                puts count
                raise "Threads out of sync"
            end
        end
    end
end
get_thread.join
```