Ruby - Bug #21360

Inconsistent Support for `Exception#cause` in `Fiber#raise` and `Thread#raise`

05/22/2025 09:39 AM - ioquatix (Samuel Williams)

Status:	Open		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:		**Backport:**	3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN

Description

The raise method supports setting the cause of an exception using the cause: keyword, but this behavior does not work as expected when calling Fiber#raise or Thread#raise, resulting in a TypeError. This breaks consistency with Kernel#raise and makes it difficult to attach causal chains to exceptions raised from other execution contexts.

The Problem

The following code behaves correctly when using Kernel#raise, correctly setting the cause:

```
cause = RuntimeError.new("cause")

begin
  raise RuntimeError, "boom", cause: cause
rescue => error
  pp error: error, cause: error.cause
end
```

Produces:

```
{error: #<RuntimeError: boom>, cause: #<RuntimeError: cause>}
```

However, using Fiber.current.raise or Thread.current.raise with the same arguments produces a TypeError:

```
begin
  Fiber.current.raise RuntimeError, "boom", cause: cause
rescue => error
  pp error: error, cause: error.cause
end
```

Results in:

```
{error: 
#<TypeError: backtrace must be an Array of String or an Array of Thread::Backtrace::Location>, cau
se: nil}
```

This occurs because the third argument is incorrectly interpreted as a backtrace, not as keyword arguments. A similar issue occurs with Thread#raise.

Proposed Solution

Update Fiber#raise and Thread#raise to accept and correctly interpret keyword arguments, including cause:, in the same manner as Kernel#raise. This would restore consistency across all raise implementations and allow causal exception chaining regardless of context. In other words, Fiber#raise and Thread#raise would be defined as the same interface as Kernel#raise.

Related issues:

Related to Ruby - Feature #21359: Introduce `Exception#cause=` for Post-Initi... **Assigned**

History

#1 - 05/22/2025 01:16 PM - ioquatix (Samuel Williams)

- Description updated

#2 - 05/22/2025 07:55 PM - Eregon (Benoit Daloze)

- Related to Feature #21359: Introduce `Exception#cause=` for Post-Initialization Assignment added