

Interruptions



Interruptions

Interrupting a fiber:

- is an effect
- (semantically) blocks until interruption is completed
- can also be forked!

Can run effects when interrupted

- resource cleanup

Child fibers

- new fiber is a child of the fiber that forked it
- child fiber cannot outlive parent, will be interrupted
- can spawn fibers as children of "main", via `.forkDaemon`

Race

- two effects are run on separate fibers
- winner (first to finish) dictates result, loser is interrupted

```
val interruption = for {  
  fib <- zioWithTime.fork  
  _ <- ... *> fib.interrupt  
  result <- fib.join  
} yield result
```

```
val zioWithTime =  
  (...).onInterrupt(ZIO.succeed("..."))
```

```
val parentEffect =  
  ZIO.succeed("spawning fiber") *>  
  zioWithTime.forkDaemon *> // < this  
  ZIO.sleep(1.second) *>  
  ZIO.succeed("parent successful")
```

```
val aRace = slowEffect.race(fastEffect)
```

ZIO rocks

