Cooperative and Forced Termination

See also

Asynchronous Exceptions in Haskell
Marlow, Peyton Jones, Moran and Reppy
PLDI, 2001

Cooperative vs. Forced Termination

Cooperative termination: ask a process to clean itself up and exit

- `Thread.interrupt()`

- Ctl-C or `kill`

- `break-thread`

Forced termination: stop a process immediately

- `Thread.stop()`

- `kill -9`

- `kill-thread`

- `custodian-shutdown-all`

Cooperative Termination in Java

- When a Java thread is interrupted with **Thread.interrupt()**, then **Object.wait()**, **Thread.join()**, or **Thread.sleep()** raises **InterruptedException**

```
class Worker implements Runnable {
  public void run() {
    try {
      while(...) {
        ... Thread.currentThread().sleep()
        ...
      }
    } catch (InterruptedException e) { ... }
  }
}
```

Is **Thread.sleep()** (or one of the other methods) called often enough?

Cooperative Termination in MzScheme

`break-thread` immediately raises an asychronous exception in the target thread — unless breaks are disabled

`parameterize-break` enables/disables breaks

Breaks are implicitly disabled in certain clean-up contexts

- Queue example: works fine due to NACKs

- Swap-channel example: need to use `wrap-evt` instead of `handle-evt`

- Initial setup: tedious, but easily abstracted to `thread/breakable`

- Code that allocates a resource: tedious!

Cooperative Termination Successes

Using CtI-C to interrupt compilation

• Cleans up half-written files before exiting

Using CtI-C to interrupt scripts

• Default handler prints stack trace

Cooperative Termination Failures

Interrupting DrScheme's Help browser:

• Initially used cooperative termination

• Quickly found to be buggy (after a few days)

• Somewhat quickly found to be still buggy (after a few months)

• Slowly found to be actually quite buggy, still (after a few years)

• Slowly found to be hopelessly buggy (after a few years more)

• **Finally** converted forced termination

  ○ *and we've never encountered a bug since*

Interrupting a package build:

• Initially written using cooperative termination

• Buggy code **immediately** converted forced termination