Subject: Coroutines package for U++
Posted by Oblivion on Sat, 05 Nov 2022 11:19:16 GMT
View Forum Message <> Reply to Message

Hi,

I have added an experimental CoRoutines package to both UppHub and upp-components repo.
It provides a simple and basic interface for building coroutines.

Main github repo (for pull request etc.):

There are two types of CoRoutine in the package:

CoRoutine -> A regular coroutine that can return any value (or void)
CoGenerator -> A regular generator type function.

- The classes provided with the package expose "traditional" Upp-like interface (Do(), Get(), Pick() and Next() -methods, depending on their type.)

- Allows exception handling and propagation.

- Also an example code is included.

Example:

```
#include <Core/Core.h>
#include "CoRoutine.h"

using namespace Upp;

CoRoutine<Vector<int>> CoMakeIota(int begin, int end, int step)
{
 Vector<int> v;
 for(int i = begin; i < end; i += step) {
  v.Add(i);
  RLOG(__func__);
  co_await CoSuspend();
 }
 co_return v;
}

CoGenerator<int> CoGenerateNumbers()
{
 int i = 0;
 for(;;) {
  RLOG(__func__);
  co_yield i++;
```

```
 }
}

CONSOLE_APP_MAIN
{
 StdLogSetup(LOG_COUT|LOG_FILE);

 auto co1 = CoMakeIota(0, 10, 2);
 auto co2 = CoGenerateNumbers();

 try {
  while(co1.Do())
   RLOG(__func__); // NOP;
  auto v = co1.Pick();
  RDUMP(v);

  for(int i = 0; i < 10; i = co2.Next())
   RDUMP(i);
 }
    catch(...) {
        RLOG("Unhandled exception");
    }

}
```

Suggestion, pull requests, reviews, etc. are welcome.

Best regards,
Oblivion
