Week 8

Exercise 66

../66/main.cc

```
#include "main.ih"

int main(int argc, char const **argv)
try
{
    if (argc == 3)
    {
        size_t Promnr = stoul(argv[1]);
        size_t stopAt = stoul(argv[2]);

        threadstarter(Promnr, stopAt);
      }
      else
      {
        cout << "Please enter nr of desired threads and number of threads "
             << "that have to be finished for the program to end\n";
      }
}
catch(...)
{
  cout << "haHA caught one!\n";
}
```

../66/main.ih

```
#define ERR(msg) printf("%s : %d", (msg), __LINE__)

#include <thread>
#include <chrono>
#include <iostream>
#include <future>
#include <vector>
#include <algorithm>

using namespace std;

string threadFun(size_t idx, promise<bool> &prom);
void threadstarter(size_t const &Promnr, size_t const &stopAt);
void run(size_t const &Promnr
                    , vector<future<bool>> &futures, size_t const &stop);
```

../66/run.cc

```
#include "main.ih"

void run(size_t const &Promnr
                      , vector<future<bool>> &futures, size_t const &stopAt)
{
  future_status status[Promnr];
  vector<future_status> vstatus(status,
     status + sizeof(status) / sizeof(status[0]) );

  size_t count = 0;
  size_t end = 0;

  vector<bool> done (Promnr, false); //zodat er niet wordt gewacht op futures
  // die al ready zijn leek me niet nodig maar dit was ook
  //niet waar de fout zit

```

```
17    while (count < 10)
18    {
19        this_thread::sleep_for(chrono::seconds(1));
20
21        for (size_t idx = 0; idx < Promnr; ++idx)
22        {
23          if (done[idx] == false)
24          {
25            if (futures[idx].wait_for(chrono::seconds(0)) == future_status::ready)
26            {
27              //cout << idx << "\t is done!\n";
28              done[idx] = true;
29              if (++end == stopAt)
30                break;
31            }//else
32            //cout << idx << " is not done!\n";
33          }
34        }
35        //cout << "end counter: " << end << '\n';
36
37        if (end == stopAt) //stoppen na stopAt ready futures statusses
38          break;
39
40        cerr << "inspecting: " << ++count << '\n';
41    }
42  }
```

../66/threadFun.cc

```
1   #include "main.ih"
2
3   string threadFun(size_t idx, promise<bool> &prom)
4   {
5       // het nut van idx hier is om te kijken welke thread het is maar het
6       // werd me niet duidelijker waar het probleem lag hierdoor
7       cerr << "entry\n";
8
9       size_t wtime = rand() % 5;
10
11      this_thread::sleep_for(chrono::seconds(wtime));
12      cerr << "first cerr\n";
13
14      this_thread::sleep_for(chrono::seconds(wtime));
15      cerr << idx << " second cerr\n";
16
17      prom.set_value(true); //bij future errors gaat het geloof ik hier fout
18      //bij seg faults wordt de message hieronder wel geprint
19
20      cerr << "set value didnt cause error here!\n";
21
22      return "end the program";
23  }
```

../66/threadstarter.cc

```
1   #include "main.ih"
2
3   void threadstarter(size_t const &Promnr, size_t const &stopAt)
4   {
5     vector<promise<bool>> promises;
6     vector<future<bool>> futures;
7
8     thread threads[Promnr];
9
10    for (size_t idx = 0; idx < Promnr; ++idx)
```

```
11    {
12      promises.emplace_back();
13      futures.emplace_back(promises[idx].get_future());
14
15      threads[idx] = thread
16      {
17        [&, idx]()
18        {
19          threadFun(idx, promises[idx]);
20        }
21      };
22    }
23
24    run(Promnr, futures, stopAt);
25
26    for (auto &it: threads)
27      it.join();
28
29  }
```