CS 3460

Introduction to STL Parallel Execution

Parallel Execution Policy

- Introduced with the C++ 17 standard, tiny update with C++ 20
- A large number of `<algorithm>`s were updated to have parallel execution policies, including vectorized parallel execution
- Four execution policies, defined in `<execution>`
  - `std::exectuion::seq` : Sequential
  - `std::exectuion::par` : Parallel
  - `std::exectuion::par_unseq` : Parallel and vectorized
  - `std::exectuion::unseq` : Single thread vectorized
- Very important: No protection from race or deadlock conditions
  - Developer has responsibility to prevent

Specifying Execution Policy

- If no policy specified, it is historical sequential execution
  - `std::sort(myArray.begin(), myArray.end());`
- First parameter is execution policy
  - `std::sort(std::execution::par, myArray.begin(), myArray.end());`
- Yes, it is that simple!
- But remember, developer is responsible to ensure no race or deadlock conditions occur

Code Demonstration – Parallel Algorithms