Лекция 1. Организация разработки программ на C++

История языка

- Появление
  - Bjarne Stroustrup (Bell Labs), начало 80-х
  - транслятор программ в язык C (cfront)

- Стандарты
  - базовые элементы для реализации компиляторами
  - 1998 / 2003 / 2005 Technical Report 1
  - C++11 (он же C++0x) / C++14 / C++17, …

Философия языка

- Максимальная обратная совместимость с C (на уровне компиляции и линковщика), но не полная
- Множество стилей
  - процедурное программирование
  - ООП
  - обобщенное программирование (STL алгоритмы)
  - метапрограммирование (boost.spirit)
- Не платить за то, что не используешь
- Избегать платформозависимых особенностей

Применимость. За!

- C++ - всего лишь инструмент, достаточно универсальный, поэтому не всегда удобный

- Программы с высокими требованиями к ресурсам компьютера: процессору и памяти
  - OS и сопутствующие программы (Windows, KDE)
  - драйверы/встроенные системы
  - научные программы
  - игры/симуляторы
  - сервера с высокой нагрузкой (google search)

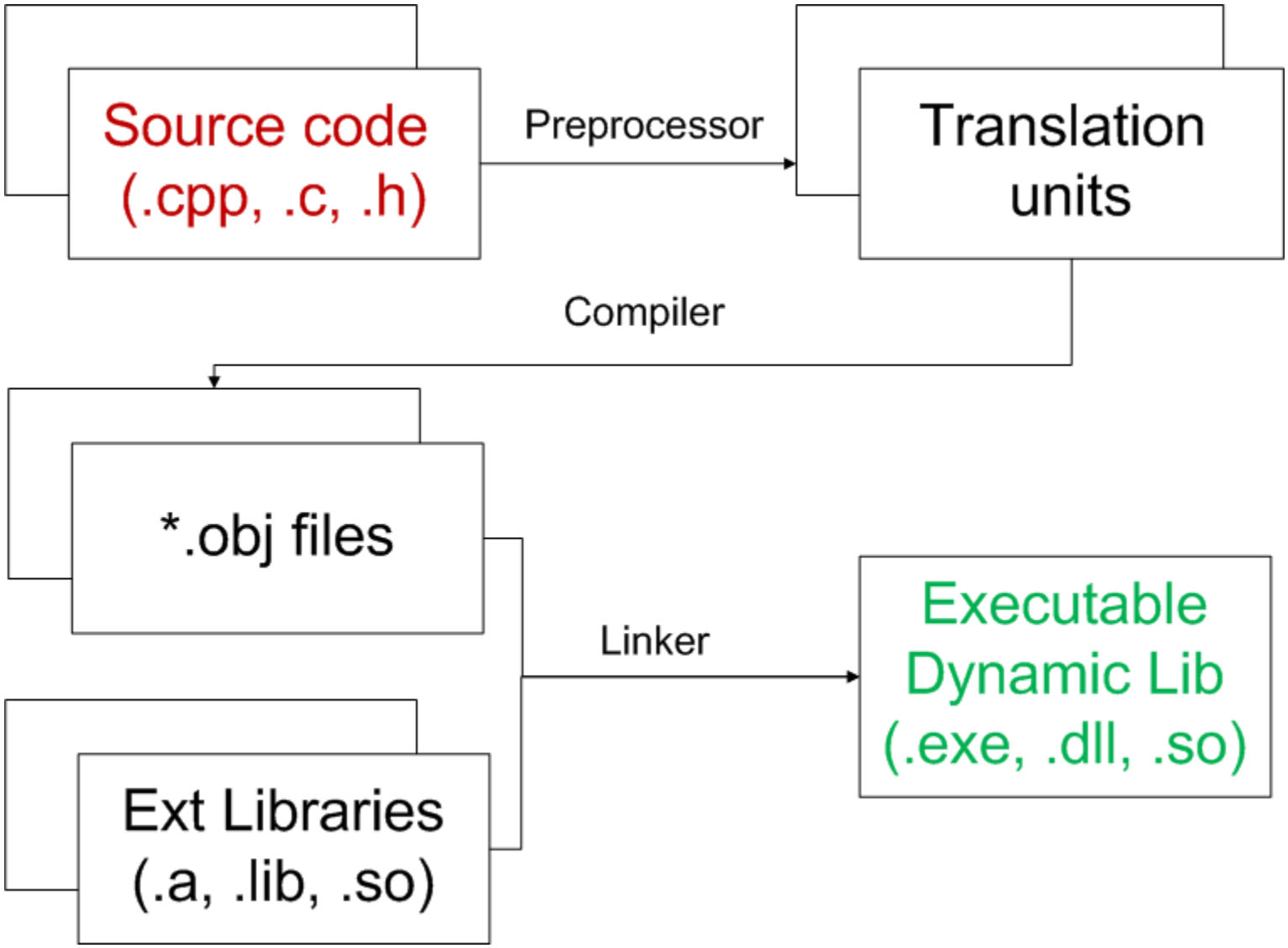
Применимость. Против!

- Возможно, не лучший выбор для:
  - клиентских частей web-приложений
  - небольших программ-сценариев (лучше скрипты)
  - приложений, нетребовательных к ресурсам
- Универсальность порождает сложность
  - более высокий порог вхождения чем Java, C#, тем более Python
- Выход: хладнокровно выбирать язык; иногда комбинировать с другими языками программирования (например, с Python или Lua). Но лучше не в одном процессе.

Библиотеки

- Стандартные: в основном CRT, STL
  - наиболее используемые структуры данных и алгоритмы
  - работа с примитивами OS (файлы, потоки)
- Работа с OS (WinAPI, POSIX)
- Общего назначения (Boost, Qt)

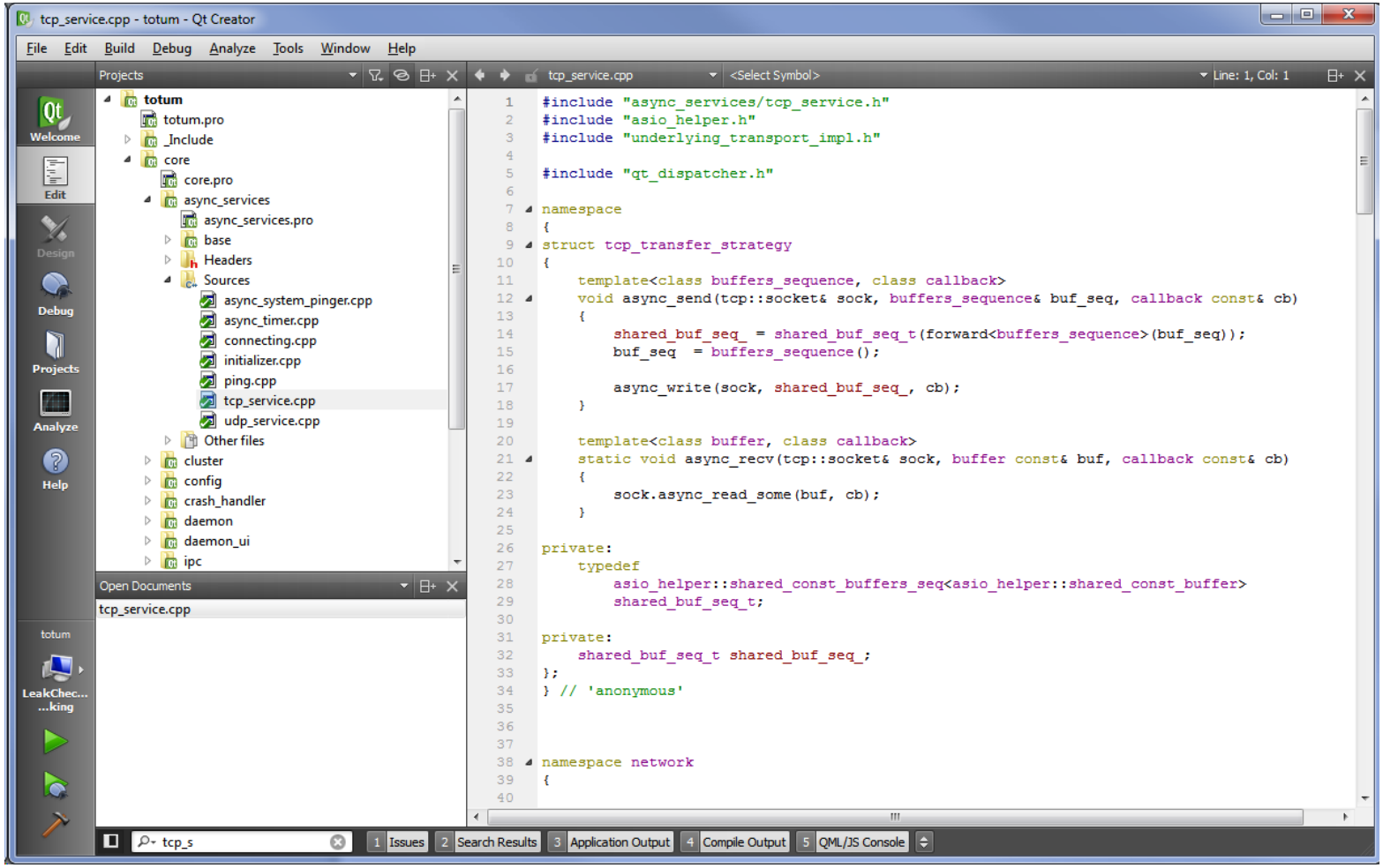
Компиляция


Source code
(.cpp, .c, .h)
Preprocessor
Translation
units
Compiler
*.obj files
Linker
Executable
Dynamic Lib
(.exe, .dll, .so)
Ext Libraries
(.a, .lib, .so)

Средства разработки

- Компиляторы: g++, msvc, clang, intel
- Автоматизация сборки: make (Makefile), cmake, autotools, SCons, qmake/qbs
- Редактор кода: notepad++, vim/emacs
- Отладчик: msvs, gdb
- Refactoring: Visual Assist, ReSharper
- Тестирование: boost.test, google.test, CUTE
- Системы контроля версий: CVS, SVN, git, hg
- Integrated development environment (IDE): MSVS, QtCreator, Eclipse CDT, NetBeans, Kdevelop, CLion
- Online compilers: ideone.com
- Статический анализ кода: PVS Studio, cppcheck, CLang

Пример IDE – Qt Creator



Выполнение программы

- Программа – последовательное выполнение инструкций
- «Точка входа» в C++ -программу:

```
1. int main()
2. {
3.     return 0;
4. }
```

Та самая программа

```
1.
2.  #include <iostream>
3.
4.  int main(int argc, char* argv[])
5.  {
6.      using namespace std;
7.      cout << "Hello, " << argv[1] << endl;
8.
9.      return 0;
10. }
```

```
1.  C:\test.exe World!
2.  Hello, World!
```

Объявление переменных

```
1.  #include <string>
2.
3.  int main()
4.  {
5.      using namespace std;
6.
7.      int    answer = 42;
8.      double pi     = 3.14;
9.
10.     string language = "C++";
11.
12.     string foo;
13.     foo = "bar";
14.
15.     return 0;
16. }
```

Условия, циклы

```
1.
2.  #include <iostream>
3.
4.  int main()
5.  {
6.      int arr [5] = {1, 2, 3, 4, 5};
7.      int sum = 0;
8.
9.      // for (initialization; continue condition; modification)
10.     for(int i = 0; i < 5; ++i)
11.         sum = sum + arr[i];
12.
13.     if (sum == 15)
14.         std::cout << "sum is correct" << std::endl;
15.
16.     return 0;
17. }
```

Функции

```
 1. #include <iostream>
 2.
 3. int factorial(int n)
 4. {
 5.     int result = 1;
 6.     while(n > 1)
 7.     {
 8.         result *= n;
 9.         --n;
10.     }
11.
12.     return result;
13. }
14.
15. int main()
16. {
17.     int n = 6;
18.     std::cout << factorial(n) << std::endl;
19.
20.     return 0;
21. }
```

Простейший ввод/вывод

```
1. pi 3.14
```

```
1.  #include <fstream>
2.  #include <string>
3.
4.  int main()
5.  {
6.      using namespace std;
7.
8.      ifstream in ("in.txt");
9.      ofstream out("out.txt");
10.
11.     string const_name;
12.     double const_value;
13.
14.     in >> const_name >> const_value;
15.     out << const_name << " " << const_value;
16.
17.     return 0;
18. }
```

Вопросы?