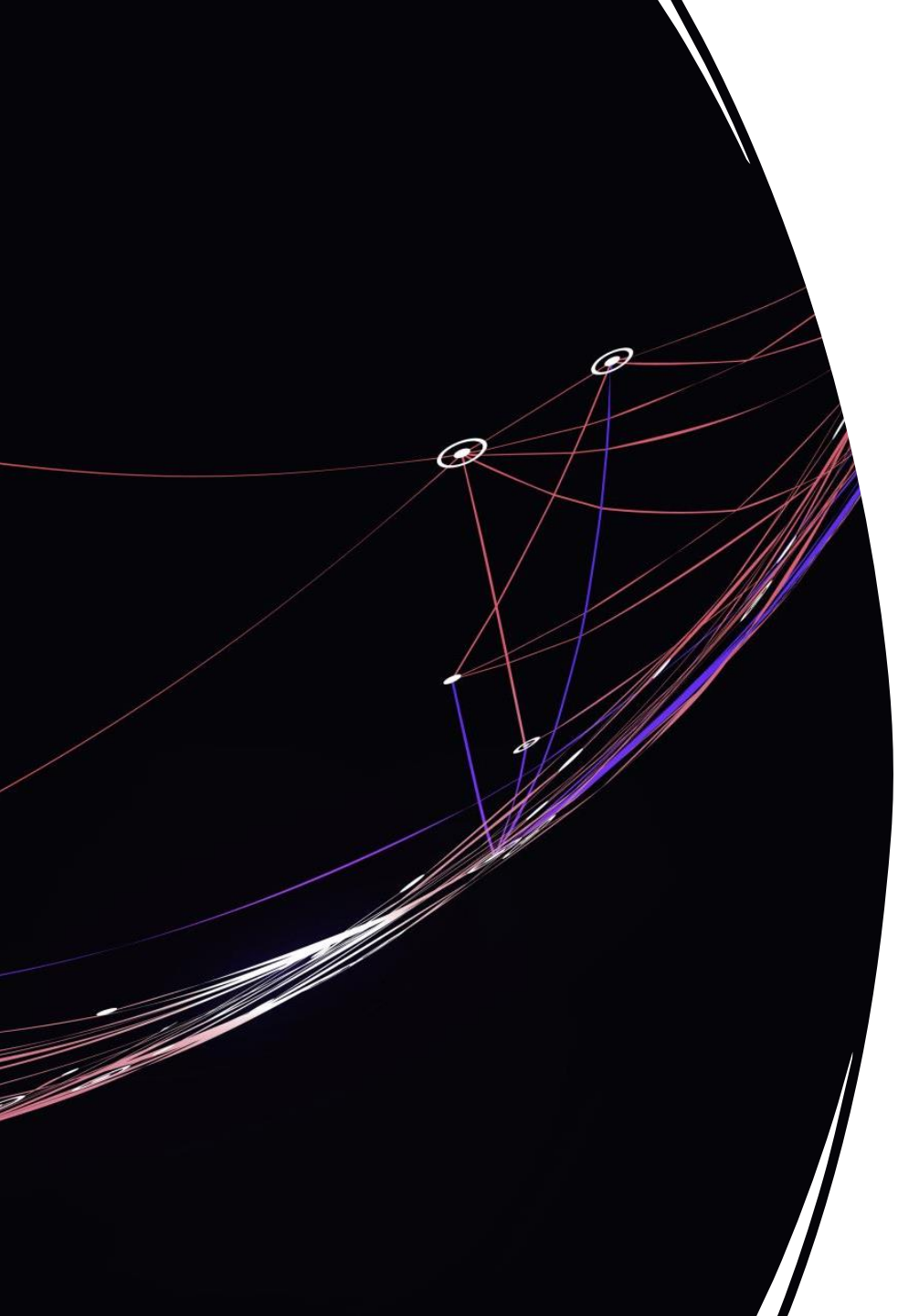


인트아이 C++ 심화 스터디

VCPKG, BOOST(ASIO, BEAST)

인하대학교 인트아이

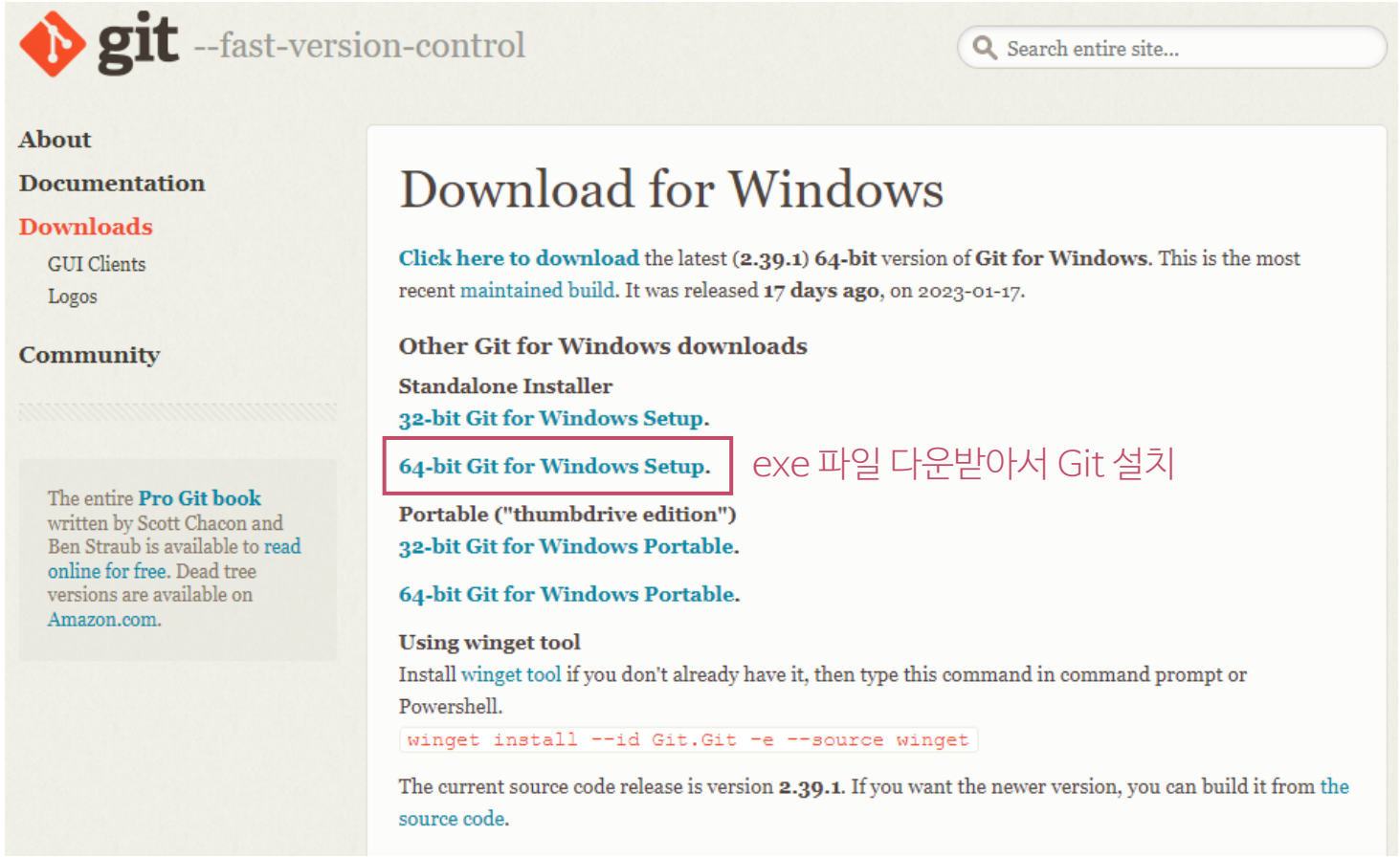
6회차
퀴즈 정답

```
template<typename T>  T가 print 메소드를 가지고 있다면 { } 안의 내용이 문제 없이 컴파일 됨
concept Printable = requires(T obj) { obj.print(); };

template<Printable P>  print 메소드를 가지고 있는 T만 받는 템플릿
void print_obj(const P &obj) {
    obj.print();
}
```

Vcpkg

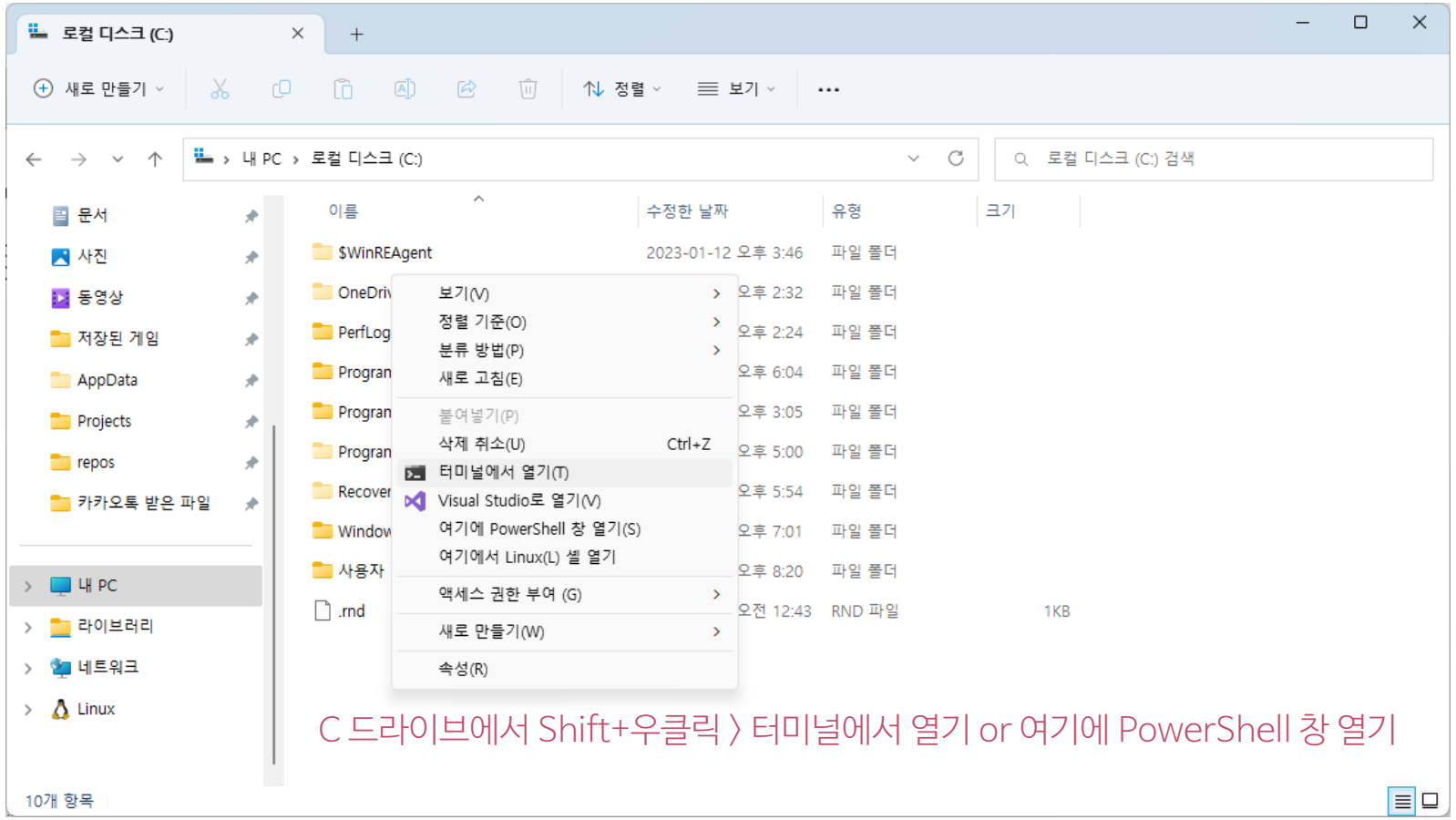
- 마이크로소프트에서 만든 C/C++ 라이브러리 관리자
- 명령어 한 번으로 Visual Studio에서 복잡한 외부 라이브러리 설정 과정을 생략할 수 있음
- Vcpkg는 Visual C++ Package Manager의 약자
- Vcpkg를 설치하기 위해서는 Git이 필요함(소스코드 다운받는 용도)

Git 설치



https://git-scm.com/download/win

Vcpkg 설치

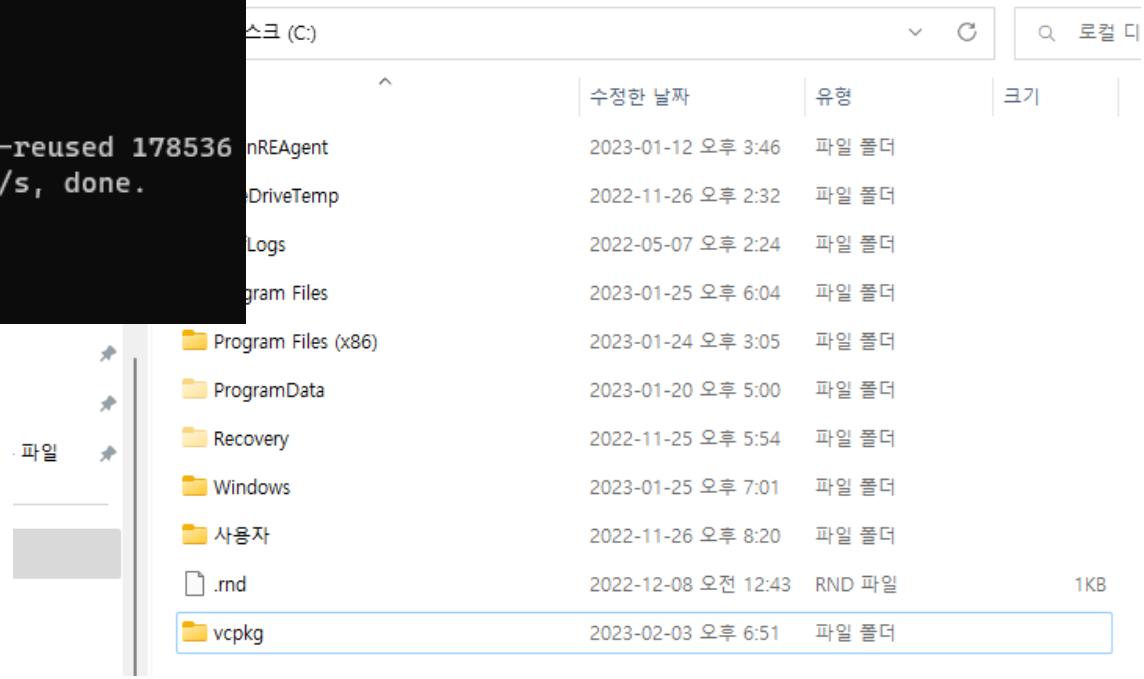


C 드라이브에서 Shift+우클릭 〉 터미널에서 열기 or 여기에 PowerShell 창 열기

Vcpkg 설치

```
PS C:\> git clone https://github.com/Microsoft/vcpkg.git
Cloning into 'vcpkg'...
remote: Enumerating objects: 178853, done.
remote: Counting objects: 100% (317/317), done.
remote: Compressing objects: 100% (180/180), done.
remote: Total 178853 (delta 193), reused 202 (delta 137), pack-reused 178536
Receiving objects: 100% (178853/178853), 69.77 MiB | 10.06 MiB/s, done.
Resolving deltas: 100% (112876/112876), done.
Updating files: 100% (9814/9814), done.
PS C:\>
```

git clone https://github.com/Microsoft/vcpkg.git 입력



C 드라이브에 vcpkg 폴더가 생겼음

Vcpkg 설치

```
PS C:\> .\vcpkg\bootstrap-vcpkg.bat
Downloading https://github.com/microsoft/vcpkg-tool/releases/download/2023-01-24/vcpkg.exe -> C:\vcpkg\vcpkg.exe... done
.
Validating signature... done.

Telemetry
---------
vcpkg collects usage data in order to help us improve your experience.
The data collected by Microsoft is anonymous.
You can opt-out of telemetry by re-running the bootstrap-vcpkg script with -disableMetrics,
passing --disable-metrics to vcpkg on the command line,
or by setting the VCPKG_DISABLE_METRICS environment variable.

Read more about vcpkg telemetry at docs/about/privacy.md
PS C:\>
```

.₩vcpkg₩bootstrap-vcpkg.bat 입력

Boost

- 편의기능을 제공하는 함수들이 포함된 C++ 라이브러리
- Boost 라이브러리 개발자의 대부분이 C++ 표준 위원회 소속 → Boost의 기능이 C++ 신버전의 표준으로 들어오기도 함 (ex. 람다 문법)
- C++ 신버전에서만 제공하는 기능을 이전버전에서도 사용할 수 있게 함 (ex. std::thread는 C++11부터 지원하기 때문에, C++03에서 Thread가 필요하면 boost::thread를 이용)
- C++ 표준에는 없는 기능을 제공해주기도 함 (ex. 문자열 split)
- C++로 프로젝트를 진행한다면 거의 필수인 라이브러리

Boost

https://www.boost.org/doc/libs/

Accumulators

Framework for incremental calculation, and collection of statistical accumulators.

Author(s)..... Eric Niebler
First Release.....1.36.0
C++ Standard Minimum Level.....03
Categories..... Math and numerics

Algorithm

A collection of useful generic algorithms.

Author(s)..... Marshall Clow
First Release.....1.50.0
C++ Standard Minimum Level.....03
Categories..... Algorithms

Align

Memory alignment functions, allocators, traits.

Author(s)..... Glen Fernandes
First Release.....1.56.0
C++ Standard Minimum Level.....03
Categories..... Memory

Any

Safe, generic container for single values of different value types.

Author(s)..... Kevlin Henney
First Release.....1.23.0
C++ Standard Minimum Level.....03
Categories..... Data structures

Array

STL compliant container wrapper for arrays of constant size.

Author(s)..... Nicolai Josuttis
First Release.....1.17.0
C++ Standard Minimum Level.....03
Categories..... Containers

Asio

Portable networking and other low-level I/O, including sockets, timers, hostname resolution, socket iostreams, serial ports, file descriptors and Windows HANDLEs.

Author(s)..... Chris Kohlhoff
First Release.....1.35.0
C++ Standard Minimum Level.....03
Categories..... Concurrent Programming, Input/Output

Assert

Customizable assert macros.

Author(s)..... Peter Dimov
First Release.....1.27.0
C++ Standard Minimum Level.....03
Categories..... Correctness and testing, Error handling and recovery

Boost 설치

```
PS C:\> .\vcpkg\vcpkg search boost
asio[coroutine]                         Boost.Coroutine (optional) if you use spawn() to launch coroutines
asio[regex]                             Boost.Regex (optional) if you use any of the read_until() or async_read_un...
asio-grpc[boost-container]              Use Boost.Container instead of <memory_resource>
autobahn                  20.8.1#1      WAMP for C++ in Boost/Asio
azmq                      1.0.3         Boost Asio style bindings for ZeroMQ
beast                     0#1           HTTP/1 and WebSocket, header-only using Boost.Asio and C++11
bext-wintls               0.9.5         Native Windows TLS stream wrapper for use with boost::asio
boost                     1.81.0#1      Peer-reviewed portable C++ source libraries
boost[mpi]                              Build with MPI support
boost-accumulators        1.81.0#1      Boost accumulators module
boost-algorithm           1.81.0#1      Boost algorithm module
boost-align               1.81.0#1      Boost align module
boost-any                 1.81.0#1      Boost any module
boost-array               1.81.0#1      Boost array module
boost-asio                1.81.0#1      Boost asio module
boost-asio[ssl]                         Build with SSL support
```

.₩vcpkg₩vcpkg search boost 로 설치 가능한 Boost 라이브러리가 있는지 찾아보기

Boost 설치

```
PS C:\> .\vcpkg\vcpkg install boost:x64-windows
Computing installation plan...
The following packages will be built and installed:
    boost[core]:x64-windows -> 1.81.0#1
  * boost-accumulators[core]:x64-windows -> 1.81.0#1
  * boost-algorithm[core]:x64-windows -> 1.81.0#1
  * boost-align[core]:x64-windows -> 1.81.0#1
  * boost-any[core]:x64-windows -> 1.81.0#1
  * boost-array[core]:x64-windows -> 1.81.0#1
  * boost-asio[core]:x64-windows -> 1.81.0#1
  * boost-assert[core]:x64-windows -> 1.81.0#1
  * boost-assign[core]:x64-windows -> 1.81.0#1
  * boost-atomic[core]:x64-windows -> 1.81.0#1
  * boost-beast[core]:x64-windows -> 1.81.0#1
  * boost-bimap[core]:x64-windows -> 1.81.0#1
  * boost-bind[core]:x64-windows -> 1.81.0#1
  * boost-build[core]:x64-windows -> 1.81.0#1
  * boost-callable-traits[core]:x64-windows -> 1.81.0#1
  * boost-chrono[core]:x64-windows -> 1.81.0#1
  * boost-circular-buffer[core]:x64-windows -> 1.81.0#1
  * boost-compatibility[core]:x64-windows -> 1.81.0#1
  * boost-compute[core]:x64-windows -> 1.81.0#1
  * boost-concept-check[core]:x64-windows -> 1.81.0#1
  * boost-config[core]:x64-windows -> 1.81.0#1
  * boost-container[core]:x64-windows -> 1.81.0#1
```

.₩vcpkg₩vcpkg install boost:x64-windows 로

Boost 라이브러리 설치(시간 오래 걸림)

라이브러리 이름 뒤에 :x64-windows를 붙여야

64비트 윈도우 라이브러리가 설치됨

설치가 끝나면, .₩vcpkg₩vcpkg integrate install 로

Vcpkg를 Visual Studio와 연동

알아두면 좋은 꿀팁

```
PS C:\Users\PSJ> vcpkg search freeglut
freeglut                3.4.0           A free OpenGL utility toolkit, the open-sourced alternative to the GLUT li...
```

Vcpkg를 이용해 박인규 교수님의 컴퓨터그래픽스설계에서 사용하는 FreeGLUT 라이브러리도 설치 가능

이름있는 C++ 라이브러리는 전부 Vcpkg를 이용해 설치 가능

```
PS C:\Users\PSJ> vcpkg search sdl SDL = 게임 프로그래밍용 라이브러리
ffmpeg[sdl2]                            Sdl2 support
imgui[sdl2-binding]                     Make available SDL2 binding
imgui[sdl2-renderer-binding]            Make available SDL2 Renderer binding
libtcod[sdl]                            Support for SDL2 windows and events with the libtcod context.
libwebp[vwebp-sdl]                      Build the vwebp viewer tool.
magnum[sdl2application]                 Sdl2Application library
osg[sdl1]                               Build SDL 1 plugin, and enable SDL 1 app features
sdl1                    1.2.15#17       Simple DirectMedia Layer is a cross-platform development library designed ...
sdl1-net                1.2.8#4         Networking library for SDL
sdl2                    2.26.2          Simple DirectMedia Layer is a cross-platform development library designed ...
sdl2[base]                              Base functionality for SDL
sdl2[samplerate]                        Use libsamplerate for audio rate conversion
sdl2[vulkan]                            Vulkan functionality for SDL
sdl2[wayland]                           Build with Wayland support
sdl2[x11]                               Build with X11 support
```

Boost 설치 확인

```
#include <boost/algorithm/cxx11/is_sorted.hpp>
#include <iostream>

int main() {
    int A[5] = { 0, 1, 3, 4, 7 };
    int B[4] = { 5, 2, 6, 1 };
    std::cout << std::boolalpha
              << boost::algorithm::is_increasing(A, A + 5) << '\n'
              << boost::algorithm::is_increasing(B, B + 4) << '\n';
    return 0;
}
```

cxx11은 C++11에서 새로 추가된 기능임을 의미 → C++11 미만 버전(ex. C++03)에서도 is_sorted를 사용할 수 있게 함

std::is_sorted

Defined in header <algorithm>

```
template< class ForwardIt >
bool is_sorted( ForwardIt first, ForwardIt last );
```

(since C++11)
(until C++20)

```
true
false
```

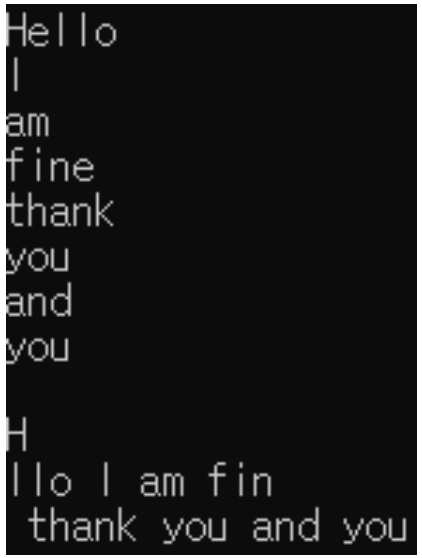
boost::algorithm::is_increasing은 주어진 배열이 오름차순으로 정렬되었는지 확인하는 함수

Boost 라이브러리는 C++ 개발에 유용한 새로운 함수들도 제공

is_increasing은 STL에는 아예 존재하지 않고 Boost에서만 존재하는 함수

Boost를 이용한 String Split

```cpp
#include <boost/algorithm/string.hpp>
#include <iostream>
#include <string>
#include <vector>

int main() {
    std::string s = "Hello I am fine thank you and you";
    std::vector<std::string> v;

    boost::algorithm::split(v, s, boost::algorithm::is_any_of(" "));
    for (const auto &e : v) {
        std::cout << e << '\n';
    }
    std::cout << '\n';

    boost::algorithm::split(v, s, boost::algorithm::is_any_of("e"));
    for (const auto &e : v) {
        std::cout << e << '\n';
    }
    return 0;
}
```



Boost Asio

- Asio = Asynchronous I/O (비동기 입출력)
- 비동기 = 함수의 결과가 바로 나오지 않음 (ex. readFile 함수로 파일을 읽을 때 파일 내용을 반환할 때까지 기다리지 않음)
- 함수의 반환 값을 기다리지 않기 때문에, Thread와 함께 사용하면 동시에 여러 작업을 처리할 수 있음
- 주로 실행시간이 오래 걸리는 파일, 네트워크(인터넷) 입출력에 이용
- Asio 안에는 비동기 관련 기능만 들어있는 것이 아니라, 입출력 전체에 관한 기능이 전부 들어있음(ex. TCP, UDP, IO context 등)
- boost::asio 네임스페이스

Boost Beast

- Boost에서 제공하는 HTTP 라이브러리
- HTTP를 이용해 인터넷에서 데이터를 가져오는 작업은 오래 걸리므로, 일반적으로는 Asio를 이용해 비동기로 진행
- 스터디에서 비동기까지 다루면 난이도가 너무 올라가기 때문에, 동기 방식으로 진행
- boost::beast 네임스페이스

Clients

These HTTP clients submit a GET request to a server specified on the command line, and prints the resulting response. The crawl client asynchronously fetches the document root of the 10,000 top ranked domains, this may be used to evaluate robustness. All asynchronous clients support timeouts.

Description	Source File	Source File (using SSL)
HTTP, synchronous	http_client_sync.cpp	http_client_sync_ssl.cpp
HTTP, asynchronous	http_client_async.cpp	http_client_async_ssl.cpp
HTTP, asynchronous using `net::system_executor`		http_client_async_ssl_system_executor.cpp
HTTP, coroutine	http_client_coro.cpp	http_client_coro_ssl.cpp
WebSocket, C++20 coroutine	http_client_awaitable.cpp	
HTTP crawl (asynchronous)	http_crawl.cpp	
HTTP json_body (synchronous)	example/http/client/body/json_client.cpp	

https://www.boost.org/doc/libs/1_81_0/libs/beast/doc/html/beast/examples.html#beast.examples.clients

HTTP

Naver, Google 등의 대부분의 사이트에서 사용하는 HTTPS는 port 번호가 443

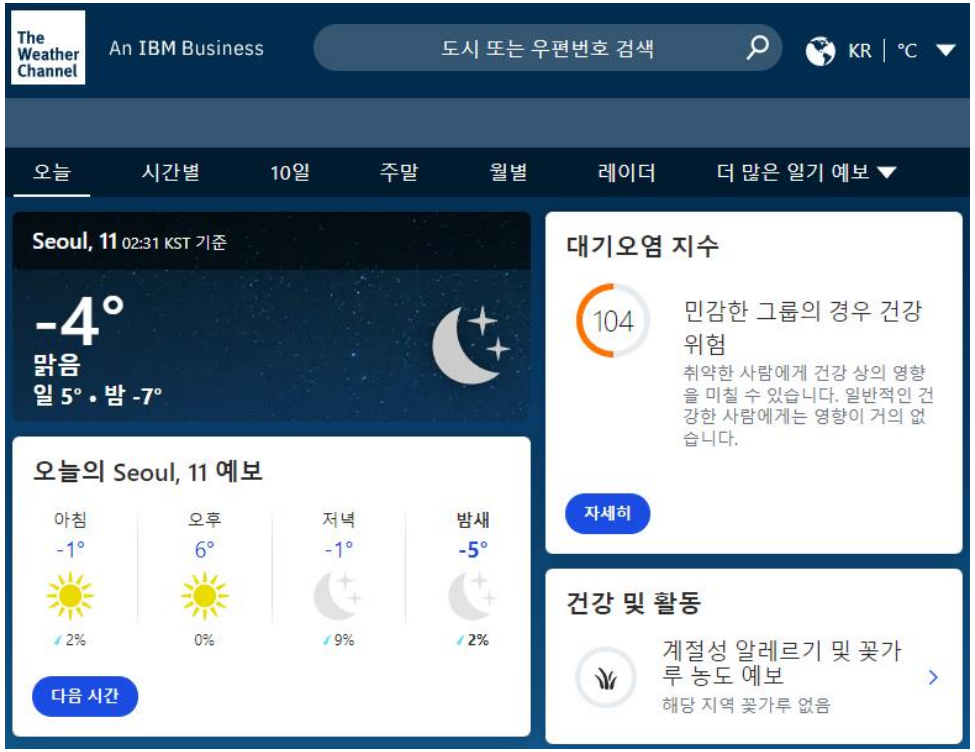
HTTP는 port 번호가 80

HTTP는 1.0과 1.1이 있는데, 일반적으로 HTTP 1.1을 사용

Fetch weather.com

```
#include <boost/asio/connect.hpp>
#include <boost/asio/ip/tcp.hpp>
#include <boost/asio/ssl/error.hpp>
#include <boost/asio/ssl/stream.hpp>
#include <boost/beast/core.hpp>
#include <boost/beast/http.hpp>
#include <boost/beast/ssl.hpp>
#include <boost/beast/version.hpp>
#include <cstdlib>
#include <exception>
#include <iostream>
#include <string>
#include <vector>

namespace asio = boost::asio;
namespace beast = boost::beast;
```

실습에 필요한 헤더파일 include, 네임스페이스 정의



https://weather.com/ko-KR/weather/today/l/KSXX0037

Fetch weather.com

```
int main() {
    try {
        const char *const host = "weather.com";
        const char *const port = "443";
        const char *const target = "/ko-KR/weather/today/l/KSXX0037";
        int version = 11;

        // The io_context is required for all I/O
        asio::io_context ioc;

        // The SSL context is required, and holds certificates
        asio::ssl::context ctx(asio::ssl::context::tls_client);

        // Verify the remote server's certificate
        ctx.set_verify_mode(asio::ssl::verify_none);

        // These objects perform our I/O
        asio::ip::tcp::resolver resolver(ioc);
        beast::ssl_stream<beast::tcp_stream> stream(ioc, ctx);

        // Set SNI Hostname (many hosts need this to handshake successfully)
        if (!SSL_set_tlsext_host_name(stream.native_handle(), host)) {
            beast::error_code ec{ static_cast<int>(::ERR_get_error()), asio::error::get_ssl_category() };
            throw beast::system_error{ ec };
        }
```

https://weather.com/ko-KR/weather/today/l/KSXX0037

TLS(Transport Layer Security) = 인터넷에서의 정보를 암호화해서 송수신하는 프로토콜
SSL(Secure Sockets Layer)에 기반한 기술

Stream = 정해진 포맷을 사용하여 문자 또는 바이트 형식으로 송수신되는 데이터 항목의 연속적인 흐름

SNI = Server Name Indication

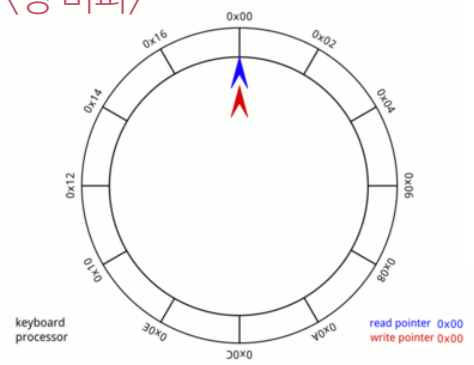
Fetch weather.com

```
// Look up the domain name
const auto results = resolver.resolve(host, port);

// Make the connection on the IP address we get from a lookup
beast::get_lowest_layer(stream).connect(results);

// Perform the SSL handshake
stream.handshake(asio::ssl::stream_base::client);

// Set up an HTTP GET request message
beast::http::request<beast::http::string_body> req{ beast::http::verb::get, target, version };
req.set(beast::http::field::host, host);

// Send the HTTP request to the remote host
beast::http::write(stream, req);

// This buffer is used for reading and must be persisted
beast::flat_buffer buffer;

// Receive the HTTP response
beast::http::response<beast::http::string_body> res;
beast::http::read(stream, buffer, res);

// Write the message to standard out
auto body = res.body();
```

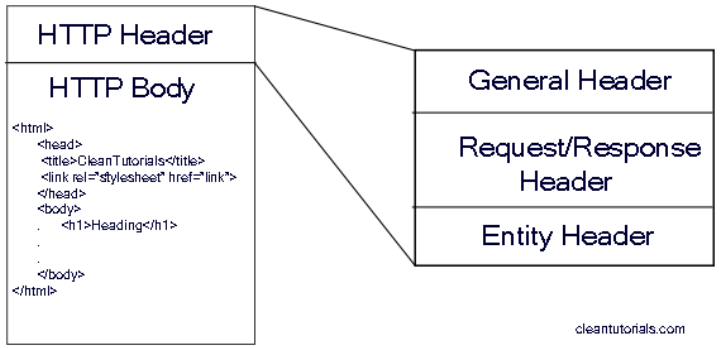
선형 동적 버퍼 (가변 길이 배열, 링 형태 x)

1. **GET 방식** - 어떠한 정보를 가져와 조회하기 위해서 사용하는 방식
   - URL에 변수(데이터)를 포함시켜 요청한다.
   - 데이터를 Header(헤더)에 포함하여 전송한다.
   - URL에 데이터가 노출되어 보안에 취약하다.
   - 전송하는 길이에 제한이 있다.
   - 캐싱(Caching, 한번 접근 후 또 요청할 시에 빠르게 접근하기위해 레지스터에 데이터를 저장시켜 놓는 것)할 수 있다.

〈링 버퍼〉



HTTP Request/response



Fetch weather.com

```
        // Print response body
        for (int i = 0; i < body.size(); i += 1000) {
            // std::cout << body.substr(i, 1000) << std::endl; // becouse of large content
        }

        body = body.substr(body.find("<span data-testid=\"TemperatureValue\" class=\"CurrentConditions--tempValue--MHmYY\">") + 81);
        body = body.substr(0, body.find("</span>") - 1);
        std::cout << "Seoul Temperature: " << body << std::endl;
    } catch (const std::exception &e) {
        std::cerr << "Error: " << e.what() << std::endl;
        return EXIT_FAILURE;
    }
    return EXIT_SUCCESS;
}
```

```
Seoul Temperature: -4
```

```
class="CurrentConditions--columns--30npQ"><div
class="CurrentConditions--primary--2DOqs"><span data-testid="TemperatureValue"
class="CurrentConditions--tempValue--MHmYY">-4°</span><div data-testid="wxPhrase"
class="CurrentConditions--phraseValue--mZC_p">맑음</div><div
class="CurrentConditions--tempHiLoValue--3T1DG">일<!-- --><span
data-testid="TemperatureValue">5°</span> <!-- -->•<!-- --> <!-- -->밤<!--
--><span data-testid="TemperatureValue">-7°</span></div></div><div
```

std::basic_string<CharT,Traits,Allocator>::find

```
size_type find( const basic_string& str, size_type pos = 0 ) const;                    (until C++11)
size_type find( const basic_string& str, size_type pos = 0 ) const noexcept;           (since C++11)
                                                                                (1)    (until C++20)
constexpr size_type find( const basic_string& str,
                          size_type pos = 0 ) const noexcept;                          (since C++20)
```

std::basic_string<CharT,Traits,Allocator>::substr

```
basic_string substr( size_type pos = 0, size_type count = npos ) const;                 (until C++20)
constexpr basic_string substr( size_type pos = 0, size_type count = npos ) const;       (since C++20)
```

Quiz

서울 날씨(온도)를
가져오는 코드를 수정해,
일본 도쿄 날씨(온도)를
출력하시오.

C++ 코드를 건드리면 어려우니,
사이트 주소만 건드리기