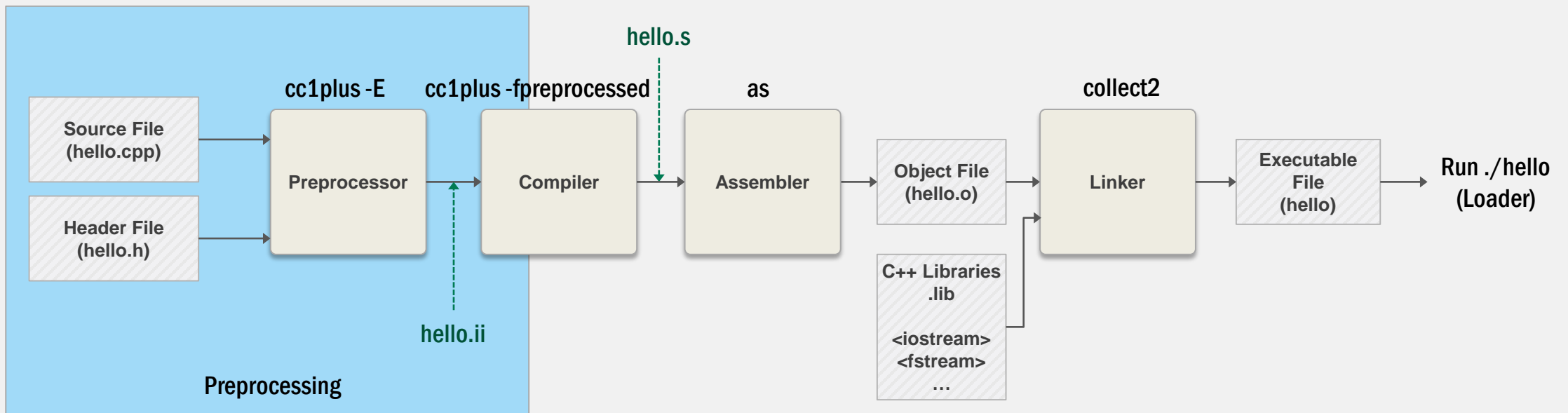


## 02. Preprocessors

국민대학교 소프트웨어학부

# Preprocessing

- C++ 전처리 과정
  - Headers, Macros 와 관련된 수행을 처리



- 수행과정
  - 1. 헤더파일의 삽입
  - 2. 매크로 치환 및 적용 과정

- The C++ compilation process -

# Preprocessing (1) – Include headers

- C++ 전처리 과정 (1)

- 예제: hello.cpp

```
#include <iostream>

int main(int argc, char *argv[])
{
    std::cout << "Hello World" << std::endl;

    return 0;
}
```

- 헤더파일

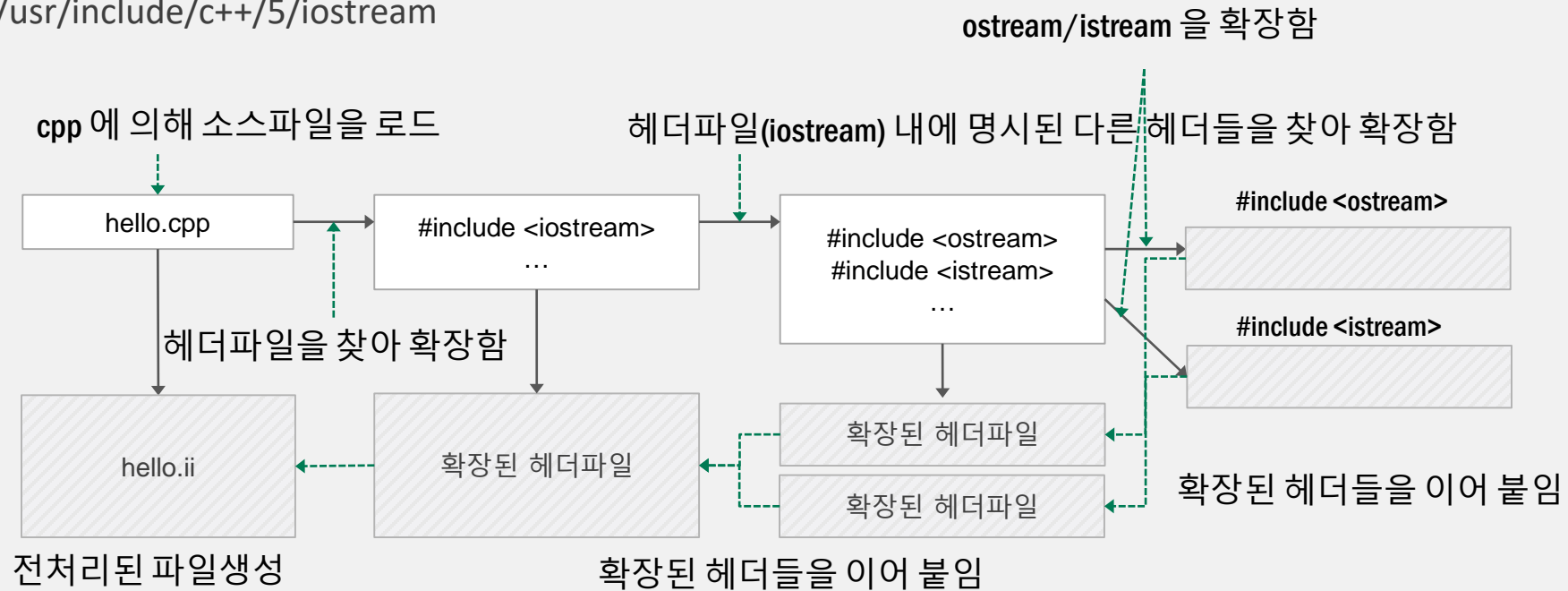
- #include <header file>

- Pound symbol (#) 을 서두에 붙여 표현
      - “;” 사용하지 않음

## Preprocessing (2) – Include headers

- C++ 전처리 과정 (2)

- **cpp** 에 의해 수행
- hello.ii 를 생산
  - 명시된 모든 헤더파일들을 확장하므로 사이즈가 커짐
  - `#include <iostream>`의 `iostream` 파일 위치
    - `/usr/include/c++/5/iostream`



### - Preprocessing procedure -

# Preprocessing (3) – Include headers

- C++ 전처리 과정 (3)

- #include <iostream> 에 대하여 확장
- 위치: /usr/include/c++/5/iostream

-v 를 했을 때, COMPILER\_PATH 안에 있는 폴더 중 <> 안에 있는 파일을 찾아 복불함. 만약에 찾아도 없으면, compile error 발생

```
#include <bits/c++config.h>
#include <ostream>
#include <istream>
```

헤더파일(iostream) 내에 명시된 다른 헤더들을 찾아 확장함

...

```
//@{
```

```
extern istream cin;          /// Linked to standard input
extern ostream cout;        /// Linked to standard output
extern ostream cerr;        /// Linked to standard error (unbuffered)
extern ostream clog;        /// Linked to standard error (buffered)
```

```
#ifdef _GLIBCXX_USE_WCHAR_T
```

```
extern wistream wcin;        /// Linked to standard input
extern wostream wcout;      /// Linked to standard output
extern wostream wcerr;      /// Linked to standard error (unbuffered)
extern wostream wclog;      /// Linked to standard error (buffered)
```

```
#endif
```

...

# Preprocessing (4) – Include headers

- C++ 전처리 과정 (3)

- hello.ii

```
...  
# 935 "/usr/include/c++/5/istream" 2 3  
# 41 "/usr/include/c++/5/iostream" 2 3  
...  
# 60 "/usr/include/c++/5/iostream" 3  
extern istream cin;  
extern ostream cout;  
extern ostream cerr;  
extern ostream clog;  
  
extern wistream wcin;  
extern wostream wcout;  
extern wostream wcerr;  
extern wostream wclog;  
...  
# 2 "hello.cpp" 2  
# 3 "hello.cpp"
```

명시된 다른 헤더들이 확장된 부분  
- 즉 명시된 헤더파일들을 을 읽어 붙여 넣어진 부분

```
int main(int argc, char *argv[])  
{  
    std::cout << "Hello World" << std::endl;  
  
    return 0;  
}
```

# Preprocessing (5) – Macros

- C++ 전처리 과정 (1)

- hello.cpp 를 수정

define 코드 : 앞으로 소스코드를 만날 때 MESSAGE를 만나면 MESSAGE를 지우고 "Hello~ "를 복붙해라.

```
#include <iostream>

#define MESSAGE "Hello World using Macro"

int main(int argc, char *argv[])
{
    std::cout << MESSAGE << std::endl;

    return 0;
}
```

- 빌드

```
kmucs@localhost:~/cpp$ g++ -v --save-temps -o hello hello.cpp
```

- 실행

```
kmucs@localhost:~/cpp$ ./hello
Hello World using Macro
```

# Preprocessing (6) – Macros

- C++ 전처리 과정 (2)

- hello.ii

```
...  
}  
# 2 "hello.cpp" 2  
...  
# 5 "hello.cpp"  
int main(int argc, char *argv[])  
{  
    std::cout << "Hello World using Macro" << std::endl;  
  
    return 0;  
}
```

정의된 문자열 "Hello World using Macro" 부분이  
MESSAGE 자리 확장됨 - cat hello.ii -

```
#include <iostream>  
  
#define MESSAGE "Hello World using Macro"  
  
int main(int argc, char *argv[])  
{  
    std::cout << MESSAGE << std::endl;  
  
    return 0;  
}
```

- hello.cpp -



# Preprocessing (7) – Macros

- C++ 전처리 과정 (3)

- hello.cpp

```
#include <iostream>

#define MESSAGE "Hello World using Macro"

int main(int argc, char *argv[])
{
    #ifdef __MSG__
        std::cout << MESSAGE << std::endl;
    #else
        std::cout << "Hello World" << std::endl;
    #endif

    return 0;
}
```

- 빌드시 옵션: -D (define)

```
kmucs@localhost:~/cpp$ g++ -v --save-temps -o hello hello.cpp -D__MSG__
```

- 실행

```
kmucs@localhost:~/cpp$ ./hello
Hello World using Macro
```

# Preprocessing (8) – Macros

- C++ 전처리 과정 (4)

- hello.ii

```
...  
# 2 "hello.cpp" 2  
  
# 5 "hello.cpp"  
int main(int argc, char *argv[])  
{  
  
    std::cout << "Hello World using Macro" << std::endl;  
  
    return 0;  
}
```

- #ifdef ~ #else 부분은 빌드 시 옵션에 의해 확장되고, “hello.ii” 내에는 존재하지 않음

# Preprocessing (9) – Macros

- C++ 전처리 과정 (4)

- hello.cpp
- 빌드시 옵션: -U (undefine)

```
kmucs@localhost:~/cpp$ g++ -v --save-temps -o hello hello.cpp -U__MSG__
```

또는

```
kmucs@localhost:~/cpp$ g++ -v --save-temps -o hello hello.cpp
```

- 실행

```
kmucs@localhost:~/cpp$ ./hello  
Hello World
```

- hello.ii

```
...  
# 2 "hello.cpp" 2  
...  
# 5 "hello.cpp"  
int main(int argc, char *argv[])  
{  
    std::cout << "Hello World" << std::endl;  
    return 0;  
}
```