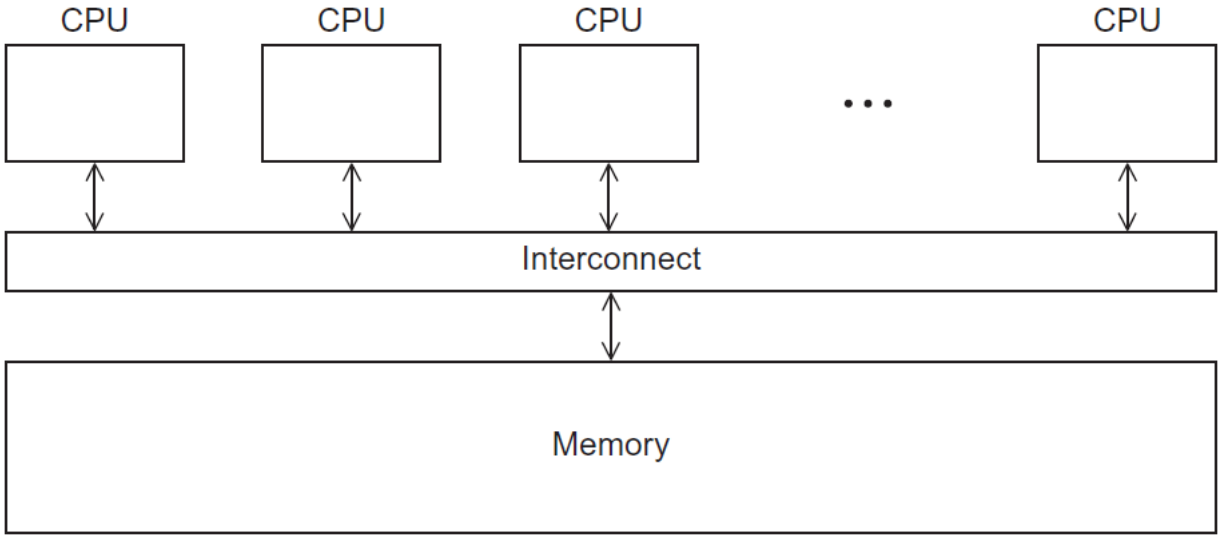
Lecture. 3

OpenMP 소개

Introduction to OpenMP

OpenMP

- **Open M**ulti **P**rocessing
- 공유메모리 환경에서
  다중 스레드 병렬 프로그램 작성을 위한
  프로그래밍 인터페이스(API)



OpenMP의 목표

표준과 이식성

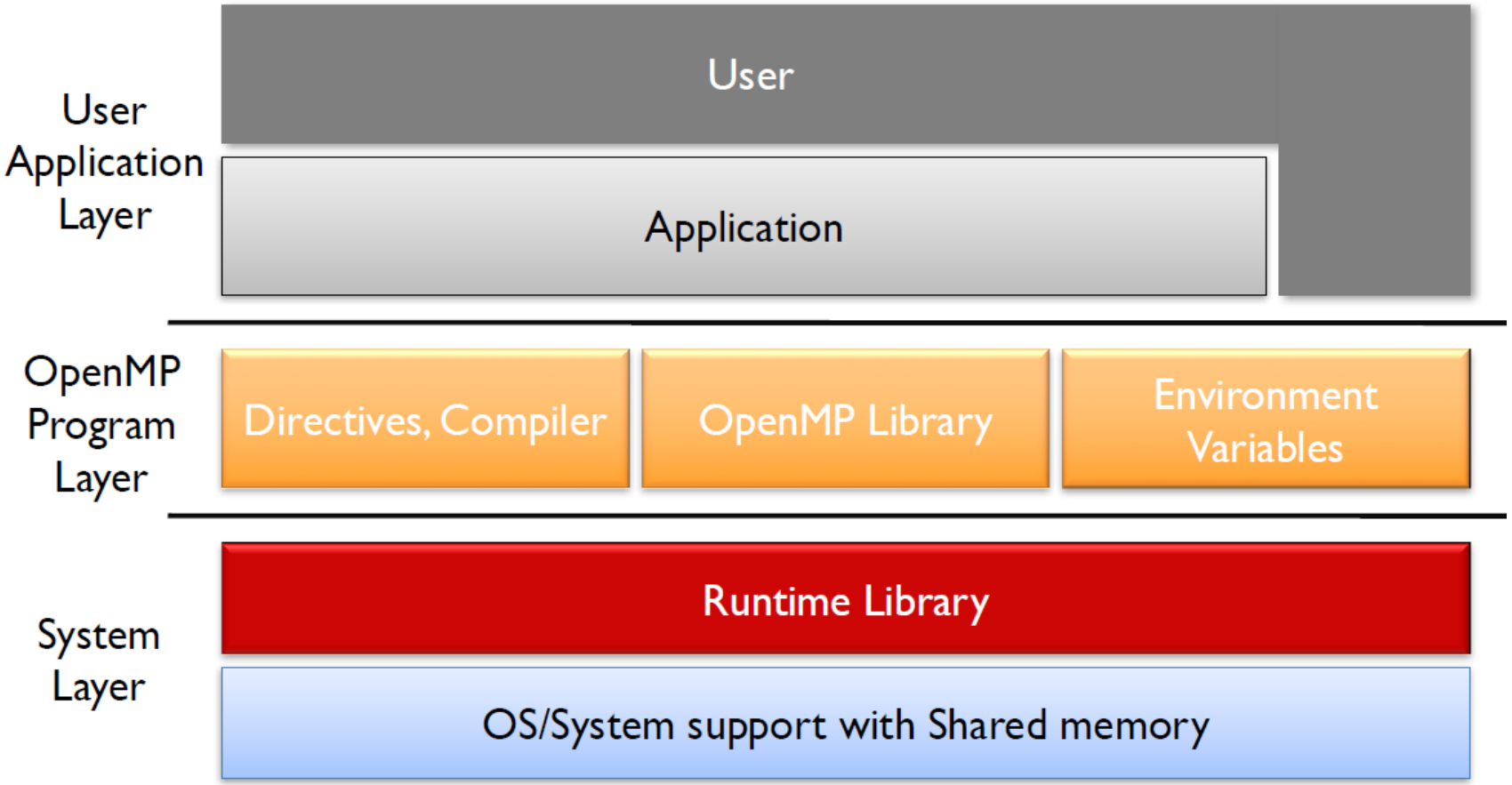
- 간단하고 사용하기 쉬운 **API** 제공
- 공유메모리 병렬 프로그래밍의 사실상 표준
- 대부분의 **OS**에 **OpenMP** 컴파일러가 존재
- **Fortran(77, 90, 95), C/C++** 지원

From Jeong's slides @ KISTI

공유메모리 병렬 프로그래밍 API

- **Pthreads (POSIX threads)**
  - 함수 라이브러리
  - **Low-level API**
    - 사용자가 제어
      - 스레드 생성, 분배 등
  - **세밀한 제어 가능 (flexible)**
  - 구현이 복잡함
    - 처음부터 병렬 알고리즘 작성 필요
- **OpenMP**
  - **지시어(directive)기반**
    - 컴파일러가 전처리 및 병렬 코드 생성
  - **High-level API**
    - 컴파일러 및 런타임의 제어
  - **구현이 간편함**
    - 지시어만 추가 하여 serial 코드를 병렬화 가능
  - **제한적 제어 기능**
    - But enough!

OpenMP Solution Stack



Images from Prof. Jeong @ SKKU

구현의 예: Hello World!

- **Pthreads (POSIX threads)**

```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

/* Global variable:  accessible to all threads */
int thread_count;

void *Hello(void* rank);  /* Thread function */

int main(int argc, char* argv[]) {
   long       thread;  /* Use long in case of a 64-bit system */
   pthread_t* thread_handles;

   /* Get number of threads from command line */
   thread_count = strtol(argv[1], NULL, 10);

   thread_handles = malloc (thread_count*sizeof(pthread_t));

   for (thread = 0; thread < thread_count; thread++)
      pthread_create(&thread_handles[thread], NULL,
          Hello, (void*) thread);

   printf("Hello from the main thread\n");

   for (thread = 0; thread < thread_count; thread++)
      pthread_join(thread_handles[thread], NULL);

   free(thread_handles);
   return 0;
}  /* main */

void *Hello(void* rank) {
   long my_rank = (long) rank;  /* Use long in case of 64-bit system */

   printf("Hello from thread %ld of %d\n", my_rank, thread_count);

   return NULL;
}  /* Hello */
```

- **OpenMP**

```c
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>

void Hello(void);  /* Thread function */

int main(int argc, char* argv[]) {
   /* Get number of threads from command line */
   int thread_count = strtol(argv[1], NULL, 10);

#  pragma omp parallel num_threads(thread_count)
   Hello();

   return 0;
}  /* main */

void Hello(void) {
   int my_rank = omp_get_thread_num();
   int thread_count = omp_get_num_threads();

   printf("Hello from thread %d of %d\n", my_rank, thread_count);

}  /* Hello */
```

구현의 예: Data Parallelism

- **Pthreads (POSIX**

```
node  A[N], B[N];

main()  {
    for (i=0;  i<nproc;  i++)
       thread_create(par_distance);
     for (i=0;  i<nproc;  i++)
       thread_join();
}
void par_distance() {
   tid = thread_id();     n = ceiling(N/nproc);
   s = tid * n;           e = MIN((tid+1)*n, N);
   for (i=s;  i<e;  i++)
      for (j=0;  j<N;  j++)
          C[i][j] = distance(A[i], B[j]);
}
```

- **OpenMP**

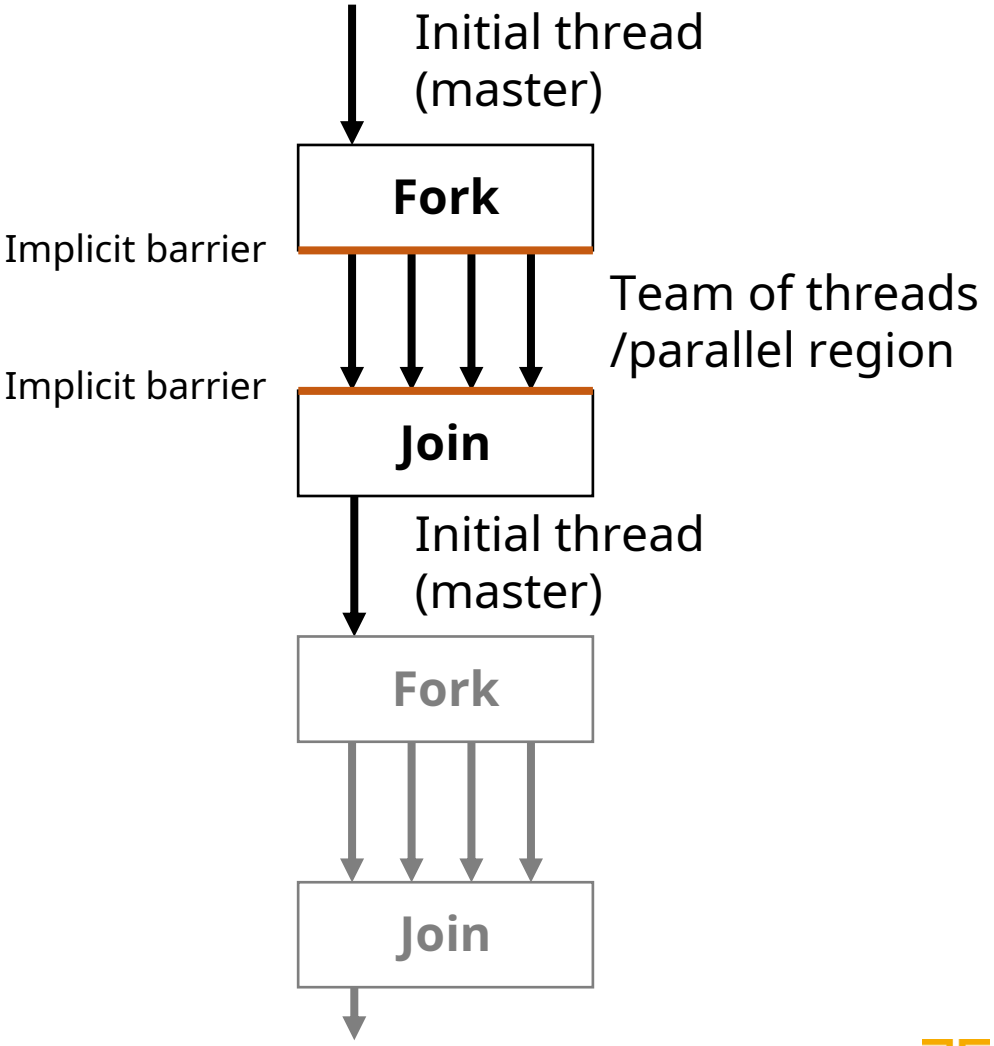
```
node  A[N], B[N];

#pragma omp parallel for
for (i=0;  i<N;  i++)
    for (j=0;  j<N;  j++)
        C[i][j] = distance(A[i], B[j]);
```

Images from Prof. Jeong @ SKKU

OpenMP Programming Model

- 스레드(Thread)기반
- Fork-join 모델
  - Implicit barrier



OpenMP Programming Model

- **Parallel construction**
  - 병렬화 영역 지정
- **Sharing work among threads**
  - 일 분배 방식 지정
- **Declare scope of variables**
  - 데이터(변수) 공유 영역 설정
- **Synchronization**
  - 스레드 간 동기화 설정

Hello OpenMP!

```
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>

void main (int argc, char*argv[]){
        int thread_count = strtol(argv[1], NULL, 10);

        #pragma omp parallel num_threads(thread_count)
        {
            printf("[Thread %d] Hello OpenMP!\n", omp_get_thread_num());
        }
        return 0;
}
```

```
> hello.exe 4
[Thread 0] Hello OpenMP!
[Thread 1] Hello OpenMP!
[Thread 2] Hello OpenMP!
[Thread 2] Hello OpenMP!
```

```
> hello.exe 4
[Thread 4] Hello OpenMP!
[Thread 1] Hello OpenMP!
[Thread 2] Hello OpenMP!
[Thread 0] Hello OpenMP!
```

#Pragma

- 특별한 컴파일러 지시어
  - 기본 C/C++ 에 확장 기능을 제공
  - 컴파일러가 지원하지 않는 Pragma는 무시 됨
- 컴파일러가 지시어를 참고하여 다중 스레드 코드 생성
- **OpenMP**를 지원하는 컴파일러 필요
  - Visual studio, gcc 등
  - 컴파일러 별로 지원하는 버전이 다를 수 있음

OpenMP의 문법

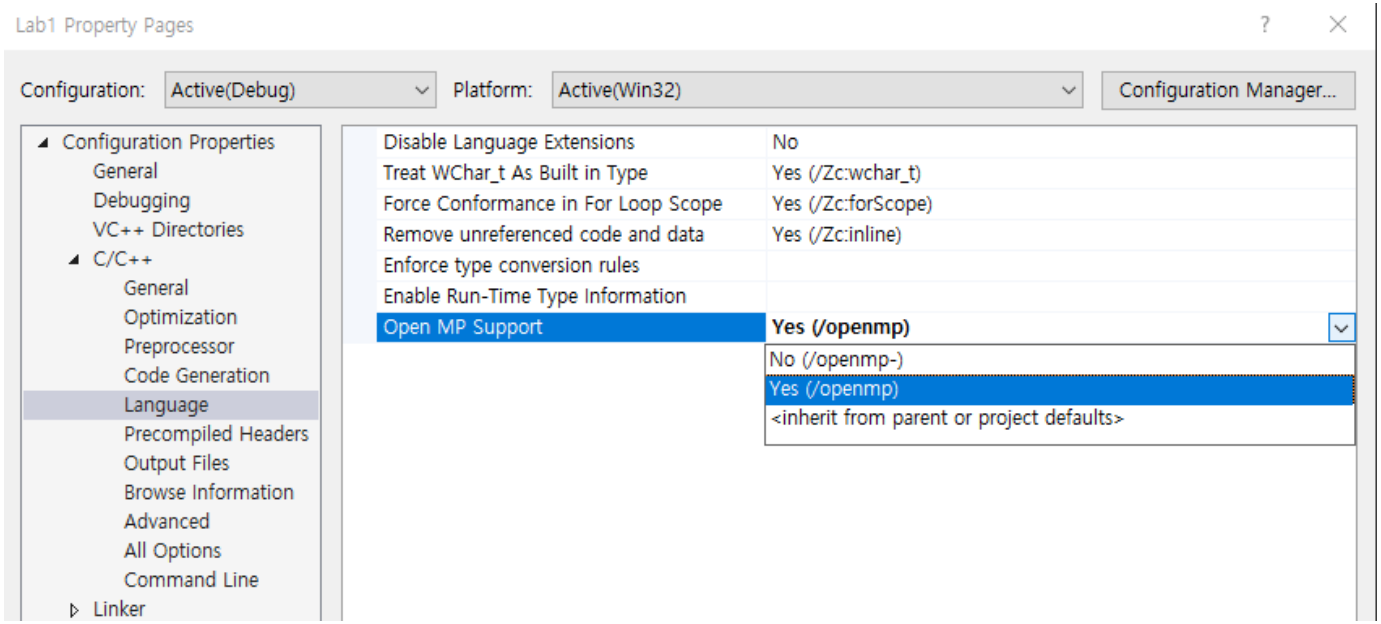
#pragma omp directive [clause list]
/* structured block */

- **Directive**
  - OpenMP의 동작 제어
    - 병렬화 영역 지정, 작업 분할 방법 등
- **Clause**
  - 동작 세부 조율
    - 스레드 수 지정, 변수 공유 설정 등

컴파일 환경 설정

- **gcc (Linux/unix)**
  - gcc –g –Wall **–fopenmp** –o omp_hello omp_hello.c

- **Visual studio 20XX**



Run Your First OpenMP Program!

- Hello OpenMP!

main.cpp

Lab1 | (Global Scope) | main(int argc, ch

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   #include <omp.h>
4
5   int main(int argc, char*argv[]) {
6       int thread_count = strtol(argv[1], NULL, 10);
7
8       #pragma omp parallel num_threads(thread_count)
9       {
10          printf("[Thread %d] Hello OpenMP!\n", omp_get_thread_num());
11      }
12
13      return 0;
14  }
15
```