

모두의 코드

←

모두의 코드

씹어먹는 C++ - <15 - 1. 동시에 실행을 시킨다고? - C++ 쓰레드(thread)>

- C
 - C++
 - C Refe
 - C++ R
 - 알고리
 - 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램
쓰레드
CPU 의 코어는 한 개가 아니다.
그래서 왜 멀티 쓰레드 이데?

병렬 가능한 (Parallelizable) 작업들

대기시간이 긴 작업들 C++에서 쓰레드 설

쓰레드에 인자 전달 하기

메모리를 같이 접근 하다면?

생각 해보기

□ 제 1

작성일 : 2019-04-01 이 글은 21316 번 읽혔습니다.

1

KG아이티뱅크 프로그래밍 교육

얼리버드등록 추가할인 이벤트

프로그래밍 기초/심화 선택교육, 파이썬/C/C++
자료구조/JAVA/JSP/SPRING 등 기초부터 심
no1-itkorea.com

열기

이번 강좌에서는

- 프로세스와 쓰레드
 - 왜 멀티 쓰레드 프로그래을 만드는가
 - C++ thread 만들기

에 대해 다릅니다.

모두의 코드

▪ C

▪ C++

▪ C Reference

▪ C++ Reference

▪ 알고리즘

▪ 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레
드 인데?

병렬 가능한
(Parallelizable) 작업들

대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

```

    DCHECK_EQ(output_type, TF_FLOAT);
    const OpRegistration* op_registration = TF_OP_REGISTRATION(
        "ShapeInference");
    TF_RETURN_IF_ERROR(op_registration->shape_inference_fn == nullptr) {
        return errors::InvalidArgument(
            "No shape inference function exists for op '", node.op(),
            "', did you forget to define it?'");
    }
    const int next_output_index = 0;
    DataType dtype = enter_output.node()->output_type(0);

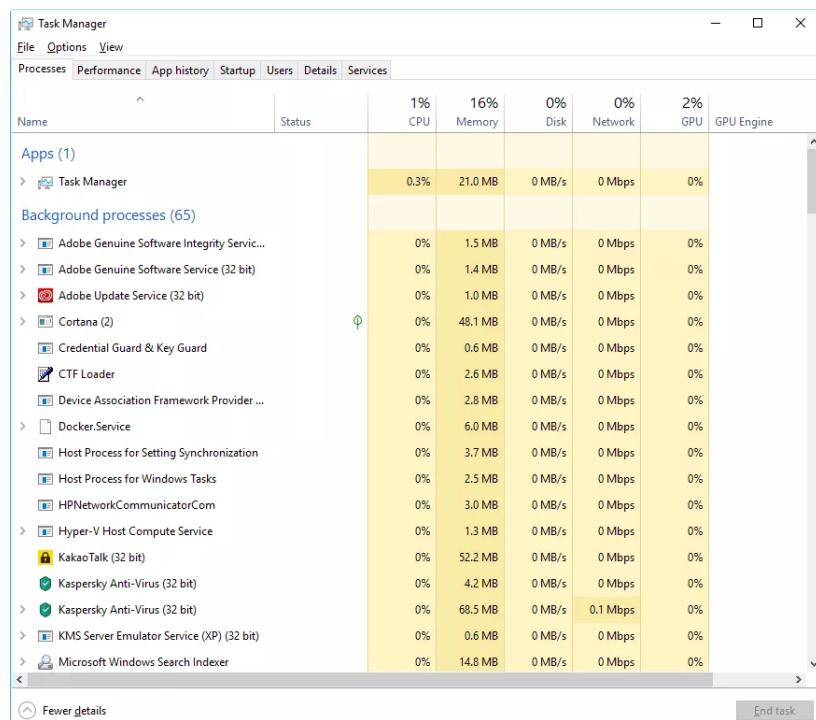
```

씹어먹는 C++

안녕하세요 여러분! 이번 강좌에서는 여태까지 작성하였던 프로그램 차원이 다른 프로그램을 만들어볼 것입니다.

멀티 쓰레드 프로그램

여러분은 작업 관리자 를 실행해 보신적이 있으신가요? 아마 실행했으면, 아래와 같은 화면을 보셨을 것입니다.



윈도우에서 나오는 작업 관리자 모습

| Process Name | % CPU | CPU Time | Threads | Idle Wake Ups | PID | User |
|-------------------------------|-------|------------|---------|---------------|-------|--------|
| | | | | | | |
| Google Chrome | 0.1 | 1:41:26.29 | 38 | 2 | 13626 | jaebum |
| Code | 0.0 | 3:08.05 | 35 | 1 | 13691 | jaebum |
| Code Helper | 0.1 | 1:03.78 | 26 | 7 | 13717 | jaebum |
| Code Helper | 0.5 | 7:23.95 | 25 | 16 | 13693 | jaebum |
| Code Helper | 0.0 | 22.20 | 22 | 1 | 13720 | jaebum |
| Code Helper | 0.0 | 6.90 | 16 | 1 | 13719 | jaebum |
| scep_daemon | 0.3 | 1:47:39.62 | 16 | 0 | 419 | root |
| Code Helper | 0.1 | 1:08.90 | 16 | 4 | 13763 | jaebum |
| Code Helper | 0.0 | 55.21 | 16 | 1 | 13754 | jaebum |
| Google Chrome Helper | 0.0 | 1:50.49 | 15 | 1 | 25171 | jaebum |
| Google Chrome Helper | 0.0 | 2.47 | 15 | 1 | 27086 | jaebum |
| Google Chrome Helper | 0.0 | 12.55 | 14 | 2 | 27164 | jaebum |
| Google Chrome Helper | 0.0 | 2:24.94 | 14 | 0 | 13646 | jaebum |
| Google Chrome Helper | 0.0 | 15.56 | 14 | 1 | 25216 | jaebum |
| Google Chrome Helper | 0.0 | 7.25 | 14 | 1 | 25669 | jaebum |
| Google Chrome Helper | 0.0 | 6.80 | 14 | 1 | 25877 | jaebum |
| Microsoft.VSCode.CPP.Exten... | 0.0 | 1:37.06 | 14 | 8 | 13799 | jaebum |
| Google Chrome Helper | 0.0 | 2.33 | 13 | 0 | 25172 | jaebum |

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

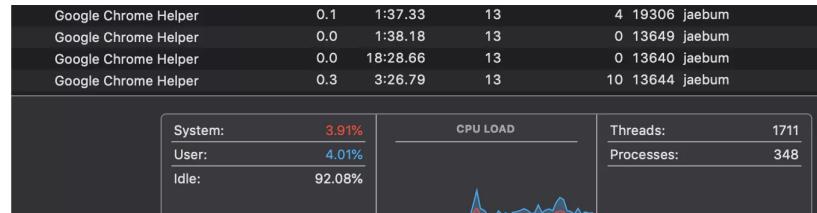
목차

- 멀티 쓰레드 프로그램
- 쓰레드
- CPU 의 코어는 한 개가 아니다.
- 그래서 왜 멀티 쓰레드 인데?

- 병렬 가능한 (Parallelizable) 작업들
- 대기시간이 긴 작업들
- C++ 에서 쓰레드 생성하기

- 쓰레드에 인자 전달하기
- 메모리를 같이 접근한다면?
- 생각 해보기

문제 1



맥에서 나오는 작업 관리자 모습

한 가지 눈여겨 볼 점은, 막대한 개수의 프로세스의 개수입니다. 프로세스란, 운영체제에서 실행되는 프로그램의 최소 단위라고 보시면 됩니다. 즉, 우리가 1 개의 프로그램을 가리킬 때 보통 1 개의 프로세스를 의미하는 경우가 많습니다.

물론 구글 크롬처럼 한 개의 탭이 한 개의 프로세스를 차지해서, 프로그램 자체가 러개의 프로세스로 이루어진 경우도 있습니다.

그렇다면 이 프로세스들은 어디에서 실행될까요? 바로 컴퓨터의 두노라 하는 CPU 의 코어 (연산하는 부분)에서 실행되고 있습니다. 옛날 (2005년 이전)에는 서버용이 아닌 일반 소비자용 CPU 의 경우 1 개의 코어를 가지는 것이 대부분이었습니다. 대표적으로 펜티엄 4 가 있지요. 이 말은 즉슨, CPU 가 한 번에 한 개의 연산을 수행한다는 것입니다.

근데 CPU 가 한 번에 한 가지 연산 밖에 못한다면, 도대체 그 시절에 인터넷을 하면서 음악을 듣고, 아니면 게임을 하는 등 여러가지 일들이 어떻게 한꺼번에 하였을까요? 분명히 제 기억에는 이러한 일들이 가능했던 것 같기 때문이지요. 그 비밀은 컨텍스트 스위칭(Context switching)이라는 기술에 숨어 있습니다.

컴퓨터에서 프로그램이 실행될 때 겉으로 보기에는 프로그램이 연속으로 쭈르륵 작동하는 것 처럼 보이지만 실제로는 그렇지 않습니다. 그래서 그림을 보면 CPU 코어 하나에서 프로그램들이 어떻게 실행되는지 알 수 있습니다.



모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

- 멀티 쓰레드 프로그램
- 쓰레드
- CPU 의 코어는 한 개가 아니다.
- 그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들
C++ 에서 쓰레드 생
성하기
쓰레드에 인자 전달
하기
메모리를 같이 접근
한다면?

생각 해보기

문제 1

코어 하나에서 프로그램들의 실행 모습

보시다시피, 프로그램 하나가 끊임없이 작동하는 것이 아니라, 프로그램 하나가 잠시 실행되었다가, 다른 프로그램으로 스위칭 되는 것을 볼 수 있습니다. 즉, CPU 는 한 프로그램을 통째로 쭉 실행시키는 것이 아니라, 이 프로그램 조금, 저 프로그램 조금씩 골라서 차례를 돌며 실행시킨다는 것을 알 수 있습니다.

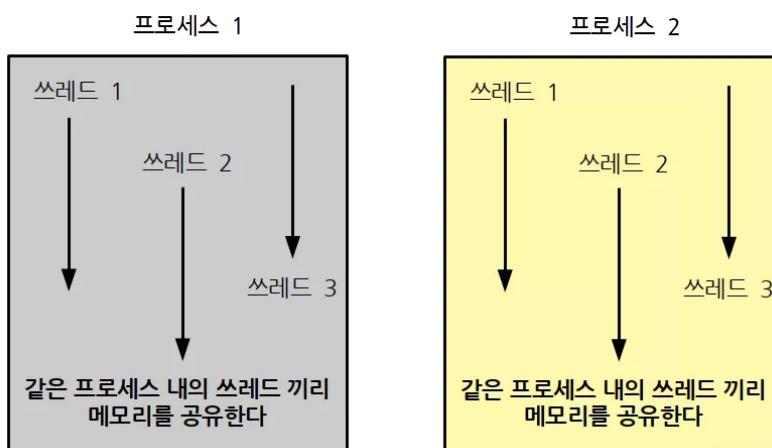
정확히 말하자면, CPU 는 그냥 운영체제가 처리하라고 시키는 명령들을 실행할 뿐, 어떤 프로그램을 실행시키고, 얼마 동안 실행시키고 또 다음에 무슨 프로그램으로 스위치 할지는 운영체제의 스케줄러 (scheduler) 알아서 결정하게 됩니다.

쓰레드

한 가지 중요한 점은, 이 CPU 코어에서 돌아가는 프로그램 단위를 쓰레드 (thread) 라고 부릅니다. 즉, CPU 의 코어 하나에서는 한 번에 개의 쓰레드의 명령을 실행시키게 됩니다.

한 개의 프로세스는 최소 한 개 쓰레드로 이루어져 있으며, 여러 개의 쓰레드로 구성될 수 있게 됩니다. 이렇게 여러개의 쓰레드로 구성된 프로그램을 멀티 쓰레드 (multithread) 프로그램 이라 합니다.

쓰레드와 프로세스의 가장 큰 차이점은 프로세스들은 서로 메모리를 공유하지 않습니다. 다시 말해, 프로세스 1 과 프로세스 2 가 있을 때 프로세스 1 은 프로세스 2 의 메모리에 접근할 수 없고, 마찬가지로 프로세스 2 도 프로세스 1 의 메모리에 접근할 수 없습니다.



프로세스는 서로의 메모리를 접근할 수 없지만, 같은 프로세스 내에 쓰레드끼리는 메모리를 공유한다

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램
쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

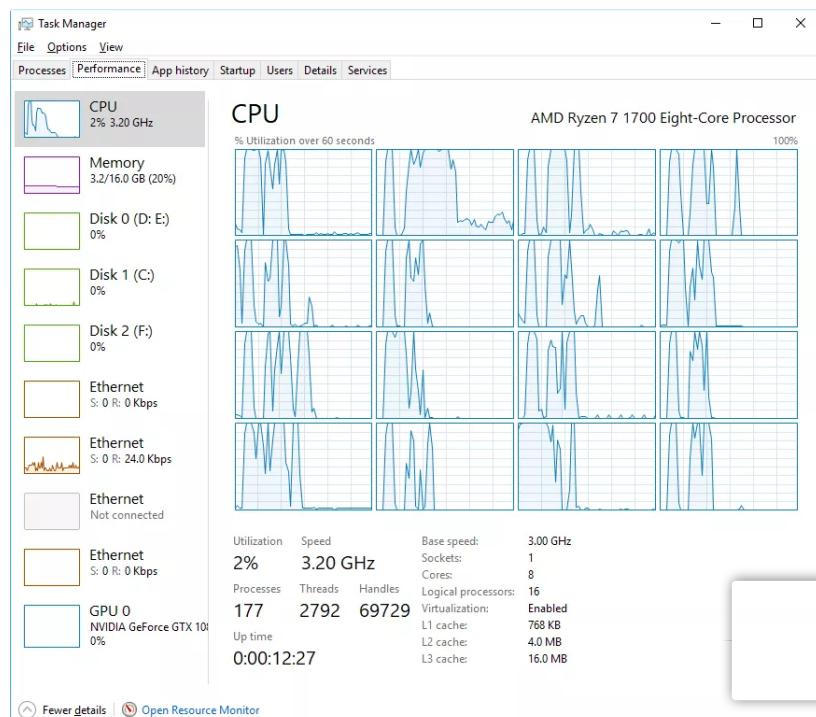
하지만 쓰레드의 경우는 다릅니다. 만일 한 프로세스 안에 쓰레드 1과 쓰레드 2가 있다면, 서로 같은 메모리를 공유하게 됩니다. 예컨대, 쓰레드 1과 쓰레드 2가 같은 변수에 값에 접근할 수 있습니다.

여태까지 여러분이 작성 하였던 프로그램들은 모두 한 개의 쓰레드로 구성된 싱글 쓰레드 프로그램입니다. 하지만, 많은 프로그램들이 멀티 쓰레드 프로그램으로 구성되어 있는데, 맥에서 나오는 작업관리자 사진의 우측 하단에, 현재 시스템의 쓰레드 개수와 프로세스에서도 알 수 있듯이, 프로세스 개수는 348 개인데, 총 쓰레드 수는 1711 개로 써습니다. 대량 프로세스 하나당 5 개의 쓰레드 들로 구성되어 있다고 생각하면 되겠네요.

CPU 의 코어는 한 개가 아니다.

요 근래 들어서는 CPU 의 발전 방향이 코어 하나의 동작 속도를 높이 보다는, CPU 에 장착된 코어 개수를 늘려가는 식으로 발전해왔습니다.

예를 들어서 인텔의 i5 모델의 경우 4 개의 코어가 장착되어 있습니다. 제가 사용하는 AMD 의 라이젠 모델의 경우 아래 그림과 같이 8 개의 코어를 가지고 있습니다. 참고로 SMT 라는 기술을 통해서 마치 16 개의 코어인 것 처럼 보이지만 하단에 Cores 를 보면 실제로는 8 개의 코어만 있다는 점을 확인할 수 있습니다.



모두의 코드



실제론 8 코어 CPU 이지만, SMT를 통해서 16개인 것 처럼 보입니다.

그렇다면 실제 이 8 개의 코어에서 프로그램이 실행되는 모습은 아래와 같을 것입니다.

▪ C

▪ C++

▪ C Reference

▪ C++ Reference

▪ 알고리즘

▪ 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

Core 1

| | | |
|-----------------|-----------------------|-----------------|
| 구글 크롬 쓰레드 #1 | 윈도우 업데이트 쓰레드 #2 | 음악 재생 쓰레드 #1 |
|-----------------|-----------------------|-----------------|

Core 2

| | | |
|-------------------------|-----------------|-----------------------|
| Visual Studio 쓰레드 #3 | 구글 크롬 쓰레드 #2 | 윈도우 업데이트 쓰레드 #1 |
|-------------------------|-----------------|-----------------------|

Core 3

| | | |
|-----------------|-------------------------|-----------------|
| 음악 재생 쓰레드 #1 | Visual Studio 쓰레드 #2 | 구글 크롬 쓰레드 #1 |
|-----------------|-------------------------|-----------------|

여러 코어들에서 쓰레드들이 실행되는 모습

따라서 이전에 싱글 코어 CPU 에서 아무리 멀티 쓰레드 프로그램이든 하더라도 결국에는 한 번에 한 쓰레드만 실행할 수 있었겠지만, 멀티 코어 CPU 에서는 여러개의 코어에 각기 다른 쓰레드들이 들어가 동시에 여러개의 쓰레드들을 효율적으로 실행할 수 있습니다.

그래서 왜 멀티 쓰레드 인데?

앞서 현대의 CPU 가 여러개의 코어를 지원함으로써 여러개의 쓰레드를 동시에 실행시킬 수 있다고 하였습니다. 그렇다면 어떨 때 프로그램을 멀티 쓰레드로 만드는 것이 유리할까요? 이에 대해 크게 두 가지 이유를 생각할 수 있습니다.

병렬 가능한 (Parallelizable) 작업들

예를 들어서 1 부터 10000 까지 더하는 작업을 생각해봅시다. 만약에 단일 쓰레드 프로그램으로 짠다면 단순히 `for` 문으로 1 부터 10000 까지 더하는 코드를 쓰면 됩니다.

반면에 이를 쓰레드 10 개로 만든다면 어떨까요. 예를 들어서 쓰레드

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

에서 1 부터 1000 까지 더하고, 쓰레드 2에서 1001 부터, 2000 까지 더하고, ... 쓰레도 10에서 9001 부터 10000 까지 더하게 한다면 어까요? 모든 쓰레드의 작업이 완료된 후에, 각각의 결과를 합치는 식으로 말이지요.

```
for (i=1; i <= 10000; i++) {
    sum += i;
}

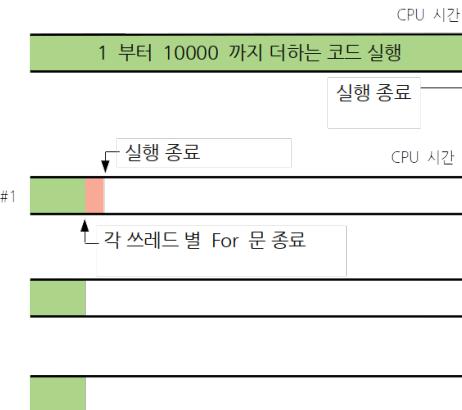
// Thread #1
for (i=1; i <= 1000; i++) {
    sum[1] += i;
}

// 모든 쓰레드 실행 끝난 후
for (i=1; i <= 10; i++) {
    sum += sum[i];
}

// Thread #2
for (i=1001; i <= 2000; i++) {
    sum[2] += i;
}

// Thread #3
for (i=2001; i <= 3000; i++) {
    sum[3] += i;
}
...

```



쓰레드 1 개를 사용할 때와 10 개를 사용할 때

CPU 코어에서 덧셈 한 번에 1 초가 걸린다고 가정해봅시다. 그렇다면 단일 쓰레드의 경우 10000 초가 걸리게 됩니다.

하지만, 멀티 쓰레드를 사용하였을 경우 CPU에 코어가 10 개가 있어서 각 쓰레드들이 동시에 실행될 수 만 있다면, 각 쓰레드에서 덧셈은 1000 초가 걸리고, 마지막으로 다 합칠 때 10 초가 걸려서 총 1010초가 걸리게 됩니다.

싱글 쓰레드의 경우보다 속도가 무려 10 배가 향상된 수치입니다!

이렇게, 어떠한 작업을 여러개의 다른 쓰레드를 이용해서 좀 더 빠르게 수행하는 것을 병렬화(parallelize)라고 합니다. 하지만 모든 작업들이 이렇게 병렬화가 가능한 것이 아닙니다. 예를 들어서 피보나치 수를 계산하는 프로그램을 생각해봅시다. 아마 아래와 같이 작성할 것입니다.

C/C++ 확대 축소

```
int main() {
    int bef = 1, cur = 1;

    // 물론 100 번째 피보나치 항을 구한다면, int 오버플로우가 나겠지만
    // 점은 여기서 무시하도록 합시다.
```

모두의 코드



• C

• C++

• C Reference

• C++ Reference

• 알고리즘

• 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그

램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레
드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

```
for (int i = 0; i < 98; i++) {
    int temp = cur;
    cur = cur + bef;
    bef = temp;
}
std::cout << "F100 : " << cur << std::endl;
}
```

위와 같은 프로그램을 여러 쓰레드를 사용하는 방식으로 실행 속도를 높일 수 있을까요?

피보나치의 n 번째 항인 F_n 을 계산하기 위해서는 F_{n-1} 과 F_{n-2} 를 알아야 합니다. 다시 말해 F_3 을 구하기 위해서는 F_1 과 F_2 를 알아야 하고, F_4 를 구하기 위해서는 F_3 과 F_2 를 알아야 합니다.

예를 들어서 F_3 을 쓰레드 1, F_4 를 쓰레드 2 에서 계산한다고 생각해봅시다. 쓰레드 2 가 값을 계산하기 위해서는 F_3 의 값이 필요합니다. 그런데, F_3 은 쓰레드 1 에서 계산되고 있으므로, 쓰레드 1 의 연산이 끝날 때 까지 쓰레드 2 가 기다려야 합니다. 따라서 최종 실행 속도는 그냥 쓰레드 1 에서 F_3 과 F_4 모두를 계산하는 것과 차이가 없게 됩니다.

결과적으로 이와 같은 방법으로 피보나치 수열을 계산하는 프로그램 병렬화 하는 것이 매우 까다롭습니다. 이러한 문제가 발생하는 근본적인 이유는 어떠한 연산 (연산 A) 을 수행하기 위해 다른 연산 (연산 B) 의 결과가 필요하기 때문이라 볼 수 있습니다. 이와 같은 상황을 A 가 B 에 의존(dependent) 한다 라고 합니다.

프로그램 논리 구조 상에서 연산들 간의 의존 관계가 많을 수록 병렬화가 어려워지고, 반대로, 다른 연산의 결과와 관계없이 독립적으로 수행할 수 있는 구조가 많을 수록 병렬화가 매우 쉬워집니다.

대기시간이 긴 작업들

인터넷에서 웹사이트들을 긁어 모으는 프로그램을 생각해봅시다. 아래와 같이 구성할 수 있을 것입니다.

C/C++

확대

축소

```
int main() {
```

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한 개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한 (Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생성하기

쓰레드에 인자 전달하기

메모리를 같이 접근한다면?

생각 해보기

문제 1

```
// 다운 받으려는 웹사이트와 내용을 저장하는 맵
map<string, string> url_and_content;
for (auto itr = url_and_content.begin(); itr != url_and_
    ++itr) {
    const string& url = itr->first;

    // download 함수는 인자로 전달받은 url 에 있는 사이트를 다운받아
    itr->second = download(url);
}
}
```

이 임의로 만든 `download` 함수는 인자로 전달한 `url` 에 위치한 웹사이트를 다운 받아서 리턴합니다.

문제는 우리의 CPU 의 처리 속도에 의해 인터넷은 매우 느리다는 점입니다.

우리가 흔히 `ping` 이라고 부르는 것은, 내가 보낸 요청이 상대 서버에 도착해서 다시 나에게 돌아오는데 걸리는 시간을 의미 합니다. 보통 우리나라 안에서 웹사이트에 요청을 보낼 시에 `ping` 이 30 밀리초 정도 나오고, 해외의 경우 (예컨대 미국), 150 밀리초에서 멀면 300 밀리초 까지 걸리게 됩니다.

이 시간은 웹사이트 전체를 다운 받는데 걸리는 시간을 말하는 것이 아닙니다. 내가 다운로드 요청을 보내서, 첫 번째 응답이 돌아올 때 까지 걸리는 시간을 말합니다.

150 밀리초라 한다면 사람 기준에서 얼마 안되는 시간처럼 보입니다. 0.15 초 이기 때문이지요. 하지만, 실제로 컴퓨터는 0.15 초 동안 정말 많은 일들을 할 수 있습니다. 보통의 CPU 는 1 초에 10^9 번 연산을 할 수 있기 때문에 0.15 초 동안 응답을 단순히 기다리기만 한다면, 1.5×10^8 번 연산을 수행할 수 있는 시간을 버리게 되는 것입니다. 즉 CPU 코어를 비효율적으로 사용하게 되는 셈이지요. 한창 일해야될 CPU 를 놀게 놔둔다니요!

| | | | | | | |
|-----------------|---------|----------------|-----------------|---------|----------------|-----------------|
| 웹사이트 1 요청 보냄 | 기다림 ... | 웹사이트 1 다운로드 | 웹사이트 2 요청 보냄 | 기다림 ... | 웹사이트 2 다운로드 | 웹사이트 3 요청 보냄 |
|-----------------|---------|----------------|-----------------|---------|----------------|-----------------|

쓰레드 1 개 만을 사용할 때

하지만 만일 `download` 함수를 호출하는 것을 여러 쓰레드에서 부르면 어떨까요?

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레
드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1



쓰레드 여러개 만을 사용할 때

위 그림은 같은 코어 안에서 쓰레드들이 컨텍스트 스위칭을 통해 기다리는 시간 없이 CPU 를 최대한으로 사용하는 것을 볼 수 있습니다. 초록색 쓰레드에서 웹사이트 1 에 요청을 보낸 후, 이전에는 웹사이트 1에서 데이터를 다운로드를 시작하기 까지 기다려야 했지만, 이 경우 초홍색 쓰레드로 컨텍스트 스위칭 되어서, 기다리는 시간을 낭비하지 않고 바로 웹사이트 2 에 요청을 보내는 것을 볼 수 있습니다.

위와 같이 처리하게 된다면 CPU 시간을 낭비하지 않고 효율적으로 작업을 처리할 수 있게 됩니다.

C++에서 쓰레드 생성하기

이전에는 C++ 표준에 쓰레드가 없어서, 각 플랫폼마다 다른 구현을 사용해야만 했습니다. (예를 들어서 윈도우즈에서는

`CreateThread` 로 쓰레드를 만들지만 리눅스에서는
`pthread_create` 로 만듭니다)

하지만 C++ 11 에서부터 표준에 쓰레드가 추가되면서, 쓰레드 사용이 매우 편리해졌습니다.

이제 첫 번째 멀티 쓰레드 프로그램을 만들어보겠습니다.

↑ 코드 크기 줄이기 C/C++ 확대 축소

```
// 내 생애 첫 쓰레드
#include <iostream>
#include <thread>
using std::thread;

void func1() {
    for (int i = 0; i < 10; i++) {
        std::cout << "쓰레드 1 작동중! \n";
    }
}
```

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드
CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생성하기

쓰레드에 인자 전달하기

메모리를 같이 접근 한다면?

생각 해보기

문제 1

```

}

void func2() {
    for (int i = 0; i < 10; i++) {
        std::cout << "쓰레드 2 작동중! \n";
    }
}

void func3() {
    for (int i = 0; i < 10; i++) {
        std::cout << "쓰레드 3 작동중! \n";
    }
}

int main() {
    thread t1(func1);
    thread t2(func2);
    thread t3(func3);

    t1.join();
    t2.join();
    t3.join();
}

```

입
력

프로그램 입력값을 여기에 입력하세요.

코드 수
정

C
실
행

성공적으로 컴파일 하였다면 (참고로 리눅스에서 컴파일 하는 분은 터미널
파일 옵션에 `-pthread` 를 추가로 넣어야 합니다.)

실행 결과

```

쓰레드 1 작동중!
쓰레드 1 작동중!
쓰레드 1 작동중!
쓰레드 1 작동중!
쓰레드 3 작동중!

```

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

쓰레드 3 작동중!
쓰레드 1 작동중!
쓰레드 1 작동중!
쓰레드 1 작동중!
쓰레드 1 작동중!
쓰레드 2 작동중!
쓰레드 2 작동중!
쓰레드 1 작동중!
쓰레드 3 작동중!
쓰레드 2 작동중!

와 같이 나옵니다.

C++ 11 에서 쓰레드를 생성하는 방법은 매우 간단합니다.

C/C++ 확대 축소

```
#include <thread>
```

일단 위 처럼 `thread` 헤더파일을 추가하고,

C/C++ 확대 축소

```
thread t1(func1);
```

`thread` 객체를 생성하는 순간 끝입니다. 이렇게 생성된 `t1` 은
자로 전달받은 함수 `func1` 을 새로운 쓰레드에서 실행하게 됩니다

즉

C/C++ 확대 축소

```
thread t1(func1);
```

모두의 코드



```
thread t2(func2);
thread t3(func3);
```

를 실행하게 되면, `func1`, `func2`, `func3` 가 각기 다른 쓰레드 상에서 실행되게 됩니다.

▪ C

▪ C++

▪ C Reference

▪ C++ Reference

▪ 알고리즘

▪ 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들

대기시간이 긴 작업들

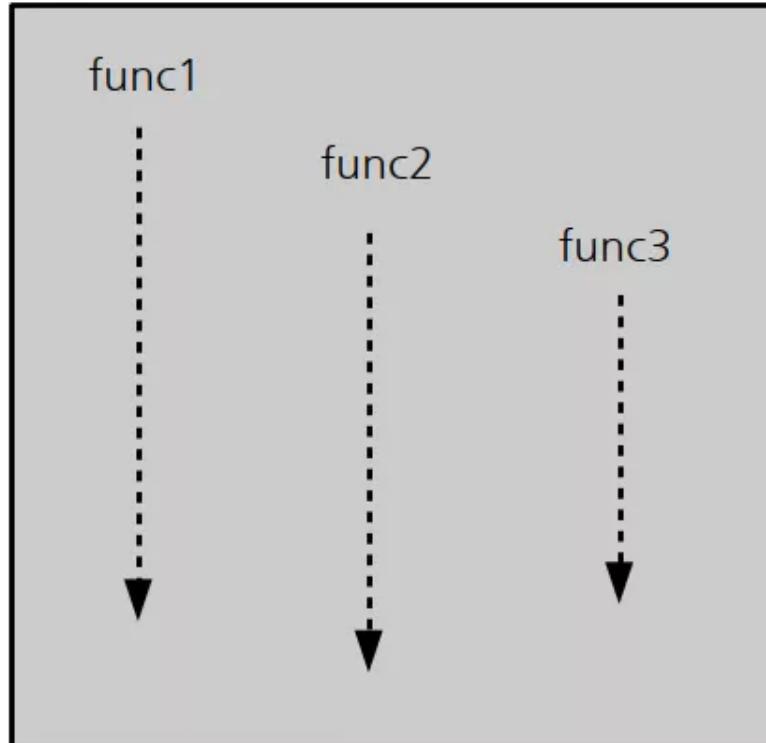
C++ 에서 쓰레드 생성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1



`func1`, `func2`, `func3` 가 각기 다른 쓰레드에서 실행된다.

한 가지 중요한 사실은 이 쓰레드들이 CPU 코어에 어떻게 할당되고 또 언제 컨텍스트 스위치를 할 지는 전적으로 운영체제의 마음에 달려 있다는 점입니다.

쓰레드 3 개를 만들었다고 해서 반드시 3 개의 각기 다른 코어에 할당되는 것이 아니다. 운이 좋으면 그렇게 되겠지만, 그냥 한 코어에 쓰레드 3 개가 컨텍스트 스위치를 하면서 돌아갈 수도 있습니다.

예를 들어서 우리의 실행 결과를 살펴봅시다.

처음에 쓰레드 1 작동중! 이 조금 나오다가, 쓰레드 3 작동중! 이 나옵니다. 그 다음에 쓰레드 2 작동중! 또 나오다가, 뒤죽박죽 순서가 바뀌어서 나오는 것을 볼 수 있습니다. 한 가지 더 재미있는 점은, 프로그램을 실행 할 때마다 그 결과가 달라진다는 점입니다. 운영체제가 쓰레

모두의 코드

▪ C

▪ C++

▪ C Reference

▪ C++ Reference

▪ 알고리즘

▪ 잡담

Copyright © 2018.

Jaebum Lee. All rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한 개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한 (Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생성하기

쓰레드에 인자 전달하기

메모리를 같이 접근한다면?

생각 해보기

문제 1

드들을 어떤 코어에 할당하고, 또 어떤 순서로 스케줄 할지는 그 때 그 때마다 상황에 맞게 바뀌기 때문에 그 결과를 정확히 예측할 수 없습니다.

아무튼 쓰레드 작동중! 메세지를 통해 그때 그때 어떠한 쓰레드의 코드가 출력되는지 짐작할 수 있습니다.

C/C++ 확대 축소

```
t1.join();
t2.join();
t3.join();
```

마지막으로 `join` 은, 해당하는 쓰레드들이 실행을 종료하면 리턴하는 함수입니다. 따라서 `t1.join()` 의 경우 `t1` 이 종료하기 전까지 리턴하지 않습니다.

그렇다면 만약에 `t2` 가 `t1` 보다 먼저 종료된다면 어떨까요? 상관 없습니다. `t1.join()` 이 끝나고 `t2.join()` 을 하였을 때 쓰레드 `t2` 가 이미 종료된 상태라면 바로 함수가 리턴하게 됩니다.

그렇다면 만약에 `join` 을 하지 않는다면 어떻게 될까요?

코드 크기 줄이기 C/C++ 확대 축소

```
#include <iostream>
#include <thread>
using std::thread;

void func1() {
    for (int i = 0; i < 10; i++) {
        std::cout << "쓰레드 1 작동중! \n";
    }
}

void func2() {
    for (int i = 0; i < 10; i++) {
        std::cout << "쓰레드 2 작동중! \n";
    }
}
```

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

```
void func3() {
    for (int i = 0; i < 10; i++) {
        std::cout << "쓰레드 3 작동중! \n";
    }
}

int main() {
    thread t1(func1);
    thread t2(func2);
    thread t3(func3);
}
```

입
력

프로그램 입력값을 여기에 입력하세요.

코드 수
정

C 실
행

성공적으로 컴파일 하였다면

실행 결과

terminate called without an active exception
쓰레드 2 작동중!

[1] 1871 abort (core dumped) ./test

와 같이 나옵니다. 일단, 보시다시피 쓰레드들의 내용이 채 실행되기 전에 `main` 함수가 종료되어서 쓰레드 객체들 (`t1`, `t2`, `t3`)의 소멸자가 호출되었음을 알 수 있습니다.

C++ 표준에 따르면, `join` 되거나 `detach` 되지 않는 쓰레드들
소멸자가 호출된다면 예외를 발생시키도록 명시되어 있습니다. 따라서,
우리의 쓰레드 객체들이 `join` 이나 `detach` 모두 되지 않았으므로 위와 같은 문제가 발생하게 됩니다.

아, 그렇다면 `detach` 가 무엇일까요? `detach` 는 말 그대로, 해당 쓰레드를 실행 시킨 후, 잊어버리는 것이라 생각하시면 됩니다. 딴 쓰레드는 알아서 백그라운드에서 돌아가게 됩니다. 아래 예제를 통해 살펴보겠습니다.

코드 크기 줄이기

C/C++

확대

축소

```
#include <iostream>
#include <thread>
```

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

```
using std::thread;

void func1() {
    for (int i = 0; i < 10; i++) {
        std::cout << "쓰레드 1 작동중! \n";
    }
}

void func2() {
    for (int i = 0; i < 10; i++) {
        std::cout << "쓰레드 2 작동중! \n";
    }
}

void func3() {
    for (int i = 0; i < 10; i++) {
        std::cout << "쓰레드 3 작동중! \n";
    }
}

int main() {
    thread t1(func1);
    thread t2(func2);
    thread t3(func3);

    t1.detach();
    t2.detach();
    t3.detach();

    std::cout << "메인 함수 종료 \n";
}
```

입
력

프로그램 입력값을 여기에 입력하세요.

코드 수
정

실
행

성공적으로 컴파일 하였다면

실행 결과

메인 함수 종료

혹은

실행 결과

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드
CPU 의 코어는 한
개가 아니다.
그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들
C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기
메모리를 같이 접근
한다면?
생각 해보기

문제 1

```
쓰레드 1 작동중!
쓰레드 1 작동중!
쓰레드 1 작동중!
쓰레드 2 작동중!
메인 함수 종료
쓰레드 3 작동중!
```

등등 여러가지 결과가 나옵니다.

기본적으로 프로세스가 종료될 때, 해당 프로세스 안에 있는 모든 쓰레드들은 종료 여부와 상관없이 자동으로 종료됩니다. 즉 `main` 함수에서 `메인 함수 종료!`를 출력하고, 프로세스가 종료하게 되면, `func1`, `func2`, `func3` 모두 더 이상 `쓰레드 작동중!`을 출력할 수 없게 됩니다.

먼저 첫번째 출력 결과가 왜 저런 식으로 나왔는지 생각해봅시다. 쓰레드를 `detach` 하게 된다면 `main` 함수에서는 더이상 쓰레드들이 종료될 때 까지 기다리지 않습니다.

따라서

C/C++ 확대 축소

```
t1.detach();
t2.detach();
```

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

- 멀티 쓰레드 프로그램
- 쓰레드
- CPU 의 코어는 한 개가 아니다.
- 그래서 왜 멀티 쓰레드 인데?
- 병렬 가능한 (Parallelizable) 작업들
- 대기시간이 긴 작업들
- C++ 에서 쓰레드 생성하기
- 쓰레드에 인자 전달하기
- 메모리를 같이 접근한다면?
- 생각 해보기

문제 1

```
t3.detach();

std::cout << "메인 함수 종료 \n";
```

위 부분이 그냥 푸르륵 실행되어서 쓰레드들이 채 문자열을 표시하기 전에 프로세스가 종료된 것이지요.

반면에 후자의 경우에는 프로세스가 종료되기 전에 운이 좋게도 생성된 쓰레드들에서 적당히 메세지를 출력하고 프로세스가 종료되었습니다. 그래도 쓰레드 1 의 경우 메세지를 3 개 밖에 작성하지 못하고 종된 것을 볼 수 있습니다.

쓰레드에 인자 전달하기

이번 예제에서는 이전에 이야기한 1 부터 10000 까지의 합을 여러 스레드들을 소환해서 빠르게 계산하는 방법을 살펴보도록 하겠습니다.

^ 코드 크기 줄이기
C/C++
⊕ 확대
⊖ 축소

```
#include <cstdio>
#include <iostream>
#include <thread>
#include <vector>
using std::thread;
using std::vector;

void worker(vector<int>::iterator start, vector<int>::iterator end, int* result) {
    int sum = 0;
    for (auto itr = start; itr < end; ++itr) {
        sum += *itr;
    }
    *result = sum;
}

// 쓰레드의 id 를 구한다.
thread::id this_id = std::this_thread::get_id();
printf("쓰레드 %x에서 %d 부터 %d 까지 계산한 결과 : %d \n", this_id, *(start - 1), *(end - 1), sum);
}

int main() {
    vector<int> data(10000);
```

모두의 코드



- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램
쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

```

for (int i = 0; i < 10000; i++) {
    data[i] = i;
}

// 각 쓰레드에서 계산된 부분 합들을 저장하는 벡터
vector<int> partial_sums(4);

vector<thread> workers;
for (int i = 0; i < 4; i++) {
    workers.push_back(thread(worker, data.begin() + i * 2500,
                           data.begin() + (i + 1) * 2500));
}

for (int i = 0; i < 4; i++) {
    workers[i].join();
}

int total = 0;
for (int i = 0; i < 4; i++) {
    total += partial_sums[i];
}
std::cout << "전체 합 : " << total << std::endl;
}

```

입
력

프로그램 입력값을 여기에 입력하세요.

코드수정

실행

성공적으로 컴파일 하였다면

실행 결과

```

쓰레드 a754700 에서 0 부터 2499 까지 계산한 결과 : 312375
0
쓰레드 9752700 에서 5000 부터 7499 까지 계산한 결과 : 156
23750
쓰레드 9f53700 에서 2500 부터 4999 까지 계산한 결과 : 937
3750
쓰레드 8f51700 에서 7500 부터 9999 까지 계산한 결과 : 218
73750
전체 합 : 49995000

```

와 같이 나옵니다.

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들

대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

C/C++ 확대 축소

```
void worker(vector<int>::iterator start, vector<int>::iterator
            int* result);
```

먼저 `worker` 함수는 덧셈을 수행할 데이터의 시작점과 끝점을 받
서 해당 범위 내의 원소들을 모두 더한 후, 그 결과를 `result` 에 저장하게 됩니다.

참고로 쓰레드는 리턴값이란것이 없기 때문에 만일 어떠한 결과를 보
환하고 싶다면 포인터의 형태로 전달하면 됩니다.

C/C++ 확대 축소

```
vector<thread> workers;
for (int i = 0; i < 4; i++) {
    workers.push_back(thread(worker, data.begin() + i * 2500
                             data.begin() + (i + 1) * 2500, i))
```

다음에 `main` 함수 안에서 각 쓰레드에게 임무를 할당하고 있는 모
습입니다. 보시다시피, 각 `worker` 들이 덧셈을 수행해야 할 범위는
`data.begin() + i * 2500, data.begin() + (i + 1) * 2500` 임을 알 수 있습니다. 즉, 첫 번째 쓰레드는 0 부터 2499 까지
두 번째 쓰레드는 2500 부터 4999 까지 쭈르륵 할당하게 됩니다.

쓰레드를 생성할 때 함수에 인자들을 전달하는 방법은 매우 간단합니
다. 우리가 이전에 `std::bind` 를 사용했던 방법을 떠올리면 됩니다.

C/C++ 확대 축소

```
thread(worker, data.begin() + i * 2500, data.begin() + (i + 1) * 2500, &partial_sums[i])
```

`thread` 생성자의 첫번째 인자로 함수 (정확히는 `Callable` 은
다 됩니다) 를 전달하고, 이어서 해당 함수에 전달할 인자들을 쭈르륵
써주면 됩니다.

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한 개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한 (Parallelizable) 작업들

대기시간이 긴 작업들

C++ 에서 쓰레드 생성하기

쓰레드에 인자 전달하기

메모리를 같이 접근한다면?

생각 해보기

문제 1

자 이제 그렇다면;

C/C++ 확대 축소

```
int sum = 0;
for (auto itr = start; itr < end; ++itr) {
    sum += *itr;
}
*result = sum;
```

실제로 `worker` 함수의 내부를 보면 정확히 해당 범위의 원소들의 덧셈을 수행하고 있음을 알 수 있습니다.

C/C++ 확대 축소

```
thread::id this_id = std::this_thread::get_id();
```

각 쓰레드에는 고유 아이디 번호가 할당 됩니다. 만약에 우리가 지금 어떤 쓰레드에서 작업중인지 보고싶다면 `this_thread::get_id()` 함수를 통해서 현재 내가 돌아가고 있는 쓰레드의 아이디를 알 수 있습니다.

C/C++ 확대 축소

```
printf("쓰레드 %x에서 %d 부터 %d 까지 계산한 결과 : %d \n", this
      *(end - 1), sum);
```

그리고 마지막으로 `printf` 함수를 통해 부분합 결과를 출력해주고 있습니다.

여기서 한 가지 궁금한 점이 있습니다. 왜 난데없이 `printf` 함수를 사용하였을까요?

한 번 위 출력 부분을 그대로 `std::cout` 으로 바꿔서 실행해보도록 하겠습니다.

C/C++ 확대 축소

```
std::cout << "쓰레드 " << hex << this_id << "에서 " << dec <
```

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

```
<< *(end - 1) << " 까지 계산한 결과 : " << sum << std::endl
```

로 치환해서 실행한다면 아래와 같이 나옵니다.

실행 결과

```
쓰레드 쓰레드 쓰레드 쓰레드 7f2d6ea5c700 에서 7f2d6f25d700  
0 에서 7f2d6fa5e70075005000 에서 부터 부터 2500 부터  
4999 까지 계산한 결과 : 93737509999 까지 계산한 결과 : 21  
8737507499 까지 계산한 결과 : 156237507f2d7025f700 에  
서  
0 부터 2499 까지 계산한 결과 : 3123750  
  
전체 합 : 49995000
```

왜 이런일이 발생하였을까요? 한 번 여러분이 컴퓨터라고 생각하고 `std::cout` 명령을 실행한다고 생각해보세요. 만약에 `std::cout << "쓰레드 "` 까지 딱 실행했는데 운영체제가 갑자기 다른 쓰레드를 실행시키면 어떨까요? 그렇다면 화면에는 쓰레드 딱 나오고 그 뒤로 다른 쓰레드의 메세지가 표시될 것입니다.

따라서 위와 같이 `std::cout` 의 `<<` 를 실행하는 과정 중간 중간에 계속 실행되는 쓰레드들이 바뀌면서 결과적으로 메세지가 뒤섞여 나타나게 됩니다.

`std::cout` 의 경우 `std::cout << A;` 를 하게 된다면 A 의 내용이 출력되는 동안 중간에 다른 쓰레드가 내용을 출력할 수 없게! 장을 해줍니다 (그 사이에 컨텍스트 스위치가 되더라도 말이지요). 하지만 `std::cout << A << B;` 를 하게 되면 A 를 출력한 이후에 B 를 출력하기 전에 다른 쓰레드가 내용을 출력할 수 있습니다.

반면에 `printf` 는 조금 다릅니다. `printf` 는 "... " 안에 있는 문자열을 출력할 때, 컨텍스트 스위치가 되더라도 다른 쓰레드들이 그 사이에 메세지를 집어넣지 못하게 막습니다. (자세한 내용은 여기 [참고](#))

따라서, 방해받지 않고 전체 메세지를 제대로 출력할 수 있게 해줍니다.

모두의 코드



C/C++ 확대 축소

```
for (int i = 0; i < 4; i++) {
    workers[i].join();
}
```

```
int total = 0;
for (int i = 0; i < 4; i++) {
    total += partial_sums[i];
}
```

■ C

■ C++

■ C Reference

■ C++ Reference

■ 알고리즘

■ 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그
램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레
드 인데?

병렬 가능한
(Parallelizable) 작업들

대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

마지막으로 `main` 함수에서 위와 같이 모든 쓰레드들이 종료될 때까지 기다립니다. 각 쓰레드에서 계산한 결과는 `partial_sums` 의 각 원소들에 저장되어 있습니다.

모든 쓰레드에서 연산이 끝난 후에, 최종적으로 `main` 함수에서 부분 합들을 모두 더해서 최종 결과를 얻을 수 있겠네요.

앞에서도 이야기 했지만 쓰레드들은 서로 메모리를 공유한다고 하였습니다. 실제로 각 쓰레드들에서 `data` 와 `partial_sums` 에 (다 부분이긴 했지만) 서로 접근할 수 있었습니다.

그렇다면 여기서 궁금한게 있습니다. 만약에, 서로 다른 쓰레드들이, 같은 메모리에 서로 접근하고 데이터를 쓴다면 어떠한 일이 발생할까요?

메모리를 같이 접근한다면?

아래 예제는 서로 다른 쓰레드들에서 `counter` 라는 변수의 값을 씩 계속 증가시키는 연산을 수행합니다.

코드 크기 줄이기 C/C++ 확대 축소

```
#include <iostream>
#include <thread>
#include <vector>
using std::thread;
using std::vector;

void worker(int& counter) {
```

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

```

for (int i = 0; i < 10000; i++) {
    counter += 1;
}

int main() {
    int counter = 0;

    vector<thread> workers;
    for (int i = 0; i < 4; i++) {
        // 레퍼런스로 전달하려면 ref 함수로 감싸야 한다 (지난 강좌 bind 키워드 참고)
        workers.push_back(thread(worker, std::ref(counter)));
    }

    for (int i = 0; i < 4; i++) {
        workers[i].join();
    }

    std::cout << "Counter 최종 값 : " << counter << std::endl;
}

```

입
력

프로그램 입력값을 여기에 입력하세요.

코드 수
정

실
행

성공적으로 컴파일 하였다면

참고로 컴파일러 최적화를 키면, 위 for 문을 그냥 `counter += 10000;` 으로 대체해리는 경우도 있으니, 정확한 효과를 보기 위해서는 컴파일러 최적화 옵션을 꺼야 합니다.

실행 결과

Counter 최종 값 : 26459

흠 결과가 조금 이상하네요? 분명히 각 쓰레드에서 10000 씩 더했기 때문에 정상적인 상황이였다면 40000 이 출력되어야 했을 것입니다. 그런데, 모든 쓰레드들이 종료되고 최종적으로 Counter 에 써진 것은 10000 이 되었습니다.

C/C++

확대

축소

```
for (int i = 0; i < 10000; i++) {
```

모두의 코드



```
    counter += 1;
}
```

이 부분을 살펴봅시다. 틀림없이 `counter` 에 1 을 10000 번 더하는 코드입니다. 그렇다면 `counter += 1` 이 문제였을까요?

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All rights reserved.

목차

- 멀티 쓰레드 프로그램
- 쓰레드
- CPU 의 코어는 한 개가 아니다.
- 그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들

대기시간이 긴 작업들

C++ 에서 쓰레드 생성하기

쓰레드에 인자 전달하기

메모리를 같이 접근 한다면?

생각 해보기

문제 1

생각 해보기

문제 1

피보나치 수열을 멀티 쓰레딩을 활용해서 빠르게 계산할 수 있는 방법은 없을까요?

강좌를 보다가 조금이라도 궁금한 것이나 이상한 점이 있다면 꼭 댓글을 남겨주시기 바랍니다. 그 외에도 강좌에 관련된 것이라면 어떤 것도 질문해 주셔도 상관 없습니다. 생각해 볼 문제도 정 모르겠으면 댓글을 달아주세요.

현재 여러분이 보신 강좌는 <15 - 1. 동시에 실행을 시킨다고? - C++ 쓰레드(thread)>입니다. 이번 강좌의 모든 예제들의 코드를 보지 않고 짤 수준까지 강좌를 읽어 보시기 전까지 다음 강좌로 넘어가지 말아주세요

다음 강좌 보러가기

< 이전 강좌

씹어먹는 C ++ - <14. 함수를 객체로! (C++ std::function, std::mem_fn, std::bind)>

다음 강좌 >

씹어먹는 C ++ - <15 - 2. C++ 뮤텍스(mutex) 와 조건 변수 (condition variable)>

모두의 코드



궁금한 것은 직접 찾아보세요!



- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.



댓글이 **23** 개 있습니다!

댓글은 글쓴이에게 큰 힘이 됩니다

강좌에 관련 없이 궁금한 내용은 [여기](#)를 사용해주세요

다음으로 로 또는 직접 입력하세요 (댓글 수정시 비밀번호가 필요합니다)



이름

비밀번호

[댓글 달기](#)

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들

대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1

모두의 코드



초보 2020-01-05T05:09:16.468Z

안녕하세요.

본문에 있는 코드를 다시 구현해 봤는데, 에러가 나오네요. 뭐가 잘못된 것인지 도무지 못찾겠어요.

▪ C

에러

▪ C++

terminate called after throwing an instance of 'std::system_error'

▪ C Reference

what(): Invalid argument

▪ C++ Reference

signal: Aborted

▪ 알고리즘

<소스>

▪ 잡담

```
#include <iostream>
```

Copyright © 2018.
Jaebum Lee. All
rights reserved.

```
#include <vector>
```

목차

멀티 쓰레드 프로그램

mutex m;

쓰레드

```
void doing(int start, int end, int& num)
```

CPU 의 코어는 한
개가 아니다.

{

그래서 왜 멀티 쓰레드 인데?

m.lock();

병렬 가능한
(Parallelizable) 작업들

```
for(int i=start; i<end; i++)
```

대기시간이 긴 작업들

num+=i;

C++ 에서 쓰레드 생
성하기

m.unlock();

쓰레드에 인자 전달
하기

}

메모리를 같이 접근
한다면?

int main()

생각 해보기

{

vector<int> count(4,0);

무제 1

int tot=0;

int target=100;

int howmany=1;

vector<thread> th_vec(4);

모두의 코드



- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램
쓰레드
CPU 의 코어는 한
개가 아니다.
그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들
C++ 에서 쓰레드 생
성하기
쓰레드에 인자 전달
하기
메모리를 같이 접근
한다면?

생각 해보기

문제 1



Jaebum Lee 2020-01-06T01:25:17.119Z

`vector<thread> th_vec(4);`

로 하면 vector 에 쓰레드 4 개가 생성이 됩니다.

따라서 뒤에 `push_back` 으로 추가한 쓰레드들은 `h_vec[4], .. th_vec[7]` 가 되죠. 이 때문에 제대로 `join` 이 되지 않고 있습니다.

[추천](#) | [답글 달기](#) | [답글 삭제](#)



open69 2020-01-07T02:37:14.941Z

`vector<thread> th_vec;` 로 수정하니 되는군
요. 정말 감사합니다. ^^

[추천](#) | [답글 달기](#) | [답글 삭제](#)



유라 2019-12-03T08:39:55.997Z

글쓴이님 보시면 답변 부탁드립니다.

현재 해석 소프트웨어를 개발하고 있습니다. mfc로 메인
프로그램을 작성하고 있고, 솔버라고 해서 계산하는 부
은 포트란으로 개발 중이구요.

계산이 5초 10초 내외로 완료된다면 메인 프로그램에서
함수 호출하고 기다리면 되지만, 30분에서 길게는 몇 시
간이나 걸립니다. 따라서 계산을 하고 있는 중에 메인
프로그램을 조작하는게 가능해야하는데 강좌에서 나온 스
레드 개념을 이용해서 구현이 가능할까요?

[추천](#) | [답글 달기](#) | [답글 삭제](#)



Jaebum Lee 2019-12-05T21:56:08.272Z

네. 쓰레드를 사용하기 딱 좋은 상황이네요

[추천](#) | [답글 달기](#) | [답글 삭제](#)



노정호 2019-10-25T10:19:06.993Z

미쳤네요.. 이렇게 좋은 글이 있다니... 모거같아요 쌀두
의코드님

[추천](#) | [답글 달기](#) | [답글 삭제](#)

모두의 코드



- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한 개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들

대기시간이 긴 작업들

C++ 에서 쓰레드 생성하기

쓰레드에 인자 전달하기

메모리를 같이 접근한다면?

생각 해보기

문제 1



Jaebum Lee 2019-12-05T21:55:36.971Z

감사합니다

[추천](#) | [답글 달기](#) | [답글 삭제](#)



스레드 질문 드립니다. 2019-10-23T06:18:45.701Z

스레드 사용 시 성능에 대해 궁금한 것이 있습니다.
멀티 스레드 사용 시, 성능이 오히려 감소하는데요. 이와 같은 경우에는 성능을 개선시킬 방법이 있을지요?

[추천](#) | [답글 달기](#) | [답글 삭제](#)



이선흥 2019-10-24T00:35:38.021Z

스레드에서 어떤 작업을 하시는지는 모르겠지만 만약에 lock을 사용하여 스레드간 리소스를 공유하고 있다면 lock 때문에 성능이 저하될 수 있습니다. 이를 피하기 위한 lock free 기법들도 많이 나와있구요

[추천](#) | [답글 달기](#) | [답글 삭제](#)



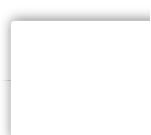
1111 2019-10-24T02:48:48.669Z

답변 감사합니다.

근데 아무것도 하지 않고 루프 100만번 돌리는 것과 메인에서 50만번 돌리고, 스레드 하나를 만들어 50만번 돌리는 것보다 성능이 더 좋습니다.

쓰레드를 만드는 것이 성능 개선에 도움을 주지 않는 것인가요? 오히려 성능이 더 나빠질 것 같아서요.

[추천](#) | [답글 달기](#) | [답글 삭제](#)



모두의 코드



- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

- 멀티 쓰레드 프로그램
- 쓰레드
- CPU 의 코어는 한 개가 아니다.
- 그래서 왜 멀티 쓰레드 인데?
- 병렬 가능한 (Parallelizable) 작업들
- 대기시간이 긴 작업들
- C++ 에서 쓰레드 생성하기
- 쓰레드에 인자 전달하기
- 메모리를 같이 접근한다면?
- 생각 해보기

문제 1



Jaebum Lee 2019-12-05T21:55:27.421Z

쓰레드를 여러개 만들었다고 해서 CPU에서 동시에 돌아간다는 보장이 없습니다. 운영체제가 어떤 식으로 스케줄을 했느냐에 달려 있지요.

특히 아무 것도 안하고 루프 50만번은! 언가 성능을 이야기 하기에 너무 짧은 시간입니다.

[추천](#) | [답글 달기](#) | [답글 삭제](#)



Gatsby 2019-10-19T05:36:41.789Z

Thread에서 실행하는 worker function의 마지막 인자 int *result 를 레퍼런스인 int& result 로 바꾸고, main의 thread 할당하는 부분에 파라미터로 partial_sums[i] 를 넣으면 이부분에서 컴파일 에러가 납니다. 이유가 무엇까요?

[추천](#) | [답글 달기](#) | [답글 삭제](#)



321 2019-10-19T05:49:07.728Z

자문자답이 되어버렸는데,
std::ref(partial_sums[i]) 로 하니 동작하네요.
<https://stackoverflow.com/a/45823556>

[추천](#) | [답글 달기](#) | [답글 삭제](#)



질문 있습니다 2019-10-05T07:02:28.464Z

하나의 스레드에서 여러개의 함수를 돌리고 싶으면 어떻게 해야 하나요?

[추천](#) | [답글 달기](#) | [답글 삭제](#)



Jaebum Lee 2019-10-21T22:16:02.755Z

여러개의 함수를 돌린다는 뜻이 여러개의 함수를 동시에" 돌린다는 의미라면 불가능합니다. 쓰레드 하나에서 한 번에 한 가지 일만 할 수 있습니다.

[추천](#) | [답글 달기](#) | [답글 삭제](#)

모두의 코드

- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드
CPU 의 코어는 한
개가 아니다.
그래서 왜 멀티 쓰레드 인데?
병렬 가능한
(Parallelizable) 작업들
대기시간이 긴 작업들

C++ 에서 쓰레드 생성하기

쓰레드에 인자 전달하기

메모리를 같이 접근한다면?

생각 해보기

문제 1



궁금합니다 2019-08-30T04:12:23.811Z

예제처럼 쓰레드간 자료의 의존성이 없는 1에서 1억까
1씩 더하는 루프문을 돌렸을때 4개의 쓰레드로 나눠서
연산하는것이랑 단순 루프를 돌렸을때 오히려 1개의
인스레드로만 돌리는게 더 빠르던데 이건 왜그런것인가
요? 대략 제 4코어 노트북기준 1에서 1억까지 더하는작
업을 4스레드로 나누었을때 2.2초 정도나오고 싱글스레
드로 돌렸을때는 0.576 초 정도 나옵니다.

[추천](#) | [답글 달기](#) | [답글 삭제](#)



Jaebum Lee 2019-09-11T23:55:30.332Z

일단 멀티 쓰레드를 사용한다고 해서 CPU 여러 층
에서 돌아간다는 보장이 없습니다. 쓰레드들을
떤 코어에 할당할지는 운영체제가 정하게 됩니다.

위 경우는 모든 코어가 사용된거 같지는 않네요.

[추천](#) | [답글 달기](#) | [답글 삭제](#)



질문있습니다! 2019-08-18T08:10:41.056Z

detach된 무한loop 스레드를 프로세스 종료없이 접근하
서 종료시키는 방법이 있나요?

[추천](#) | [답글 달기](#) | [답글 삭제](#)



Jaebum Lee 2019-08-24T22:02:14.243Z

안타깝게도 C++ 쓰레드 라이브러리에는 쓰레드를
강제로 종료 시키는 함수가 없습니다. 다만 이를 해결하는
방법으로는 detach 된 쓰레드에서 어떤 flag의 값을 계속 확인해서 flag 가 false 이면 종료하고
되도록 하면 됩니다. 그리고 그 flag 는 다른 쓰레드에서 설정해주면 되고요.

참고로 리눅스의 경우 thread 라이브러리 대신에 그냥 pthread 라이브러리를 직접 사용하면 쓰레드를 종료시키는 함수를 제공해줍니다.

[추천](#) | [답글 달기](#) | [답글 삭제](#)

모두의 코드



- C
- C++
- C Reference
- C++ Reference
- 알고리즘
- 잡담

Copyright © 2018.
Jaebum Lee. All
rights reserved.

목차

멀티 쓰레드 프로그램

쓰레드

CPU 의 코어는 한
개가 아니다.

그래서 왜 멀티 쓰레드 인데?

병렬 가능한
(Parallelizable) 작업들

대기시간이 긴 작업들

C++ 에서 쓰레드 생
성하기

쓰레드에 인자 전달
하기

메모리를 같이 접근
한다면?

생각 해보기

문제 1



감사합니다 2019-08-14T12:17:15.137Z

좋은 자료로 좋은 교육을 실천해주셔서 감사합니다.
지식 나눔을 통해 더불어 성장할 수 있는 사회를 만들고
주신 것도 감사드립니다.

[추천](#) | [답글 달기](#) | [답글 삭제](#)



Jaebum Lee 2019-08-24T22:00:31.715Z

감사합니다 :)

[추천](#) | [답글 달기](#) | [답글 삭제](#)



모두의 코드 만세 2019-07-03T14:38:15.235Z

잘 보고 갑니다 감사합니다

thread::id this_id = this_thread::get_id(); -> 이 부분
컴파일 에러가 나네요 비주얼스튜디오 2019 버전에서
요

thread::id this_id = std::this_thread::get_id(); -> 요
게 바꾸면 비주얼 스튜디오 2019 버전에서 잘 돌아가도

행복하세요 복 받으실거에요

[추천](#) | [답글 달기](#) | [답글 삭제](#)



Jaebum Lee 2019-07-04T07:45:55.079Z

감사합니다! 수정하였습니다.

[추천](#) | [답글 달기](#) | [답글 삭제](#)

