

## 1. 멀티스레드의 사용 목적

- 멀티코어 CPU에서의 프로그램 성능 향상 (O)
- 멀티 CPU 컴퓨터에서의 프로그램 성능 향상 (O)
- 분산 컴퓨터에서의 프로그램 성능 향상 (X)
- 싱글코어(CPU) 컴퓨터 환경에서의 프로그램 성능 향상 (X)
- 프로그램을 모듈화 해서 알아보기 쉽게 하기 위해 (X)

▶ 병렬컴퓨터에서의 프로그램 성능 향상을 위해서만 쓴다.

- 멀티코어 CPU에서의 프로그램 성능 향상 (O)

멀티스레드를 왜 쓰는가? 하나의 프로그램을 더 빠르게 돌리기 위해서 사용한다. 이를 걸리는 것을 하루에 끝내도록. 10 FPS 게임을 20 FPS로 올리고 싶을 때. 동접 수가 3천인데 5천으로 올리고 싶을 때. 처리량을 높이기 위해 멀티스레드 프로그래밍을 하는 것이다.

그러나 멀티코어 CPU가 아니라면 아무 의미가 없다. 멀티스레드를 쓰면 빨라진다. 그럼 싱글코어 컴퓨터에서 속도가 1시간 걸리는 것을 듀얼코어에서는 30분으로 줄일 수 있지 않은가? 절대 아니다. 싱글코어에서 아무리 멀티코어 프로그래밍을 하더라도 절대로 빨라지지 않는다. 멀티코어 프로그래밍은 멀티코어를 활용하는 방법이지 프로그램 성능을 높이는 것이 아니다. 멀티코어의 성능을 높이는 것이다. 싱글인데 그렇게 짜면 무조건 느려진다.

좀 더 자세히 이야기를 하자면, 우리가 만든 싱글 프로그램은 쿼드코어에서 4배 이상 절대로 빨라지지 않고, 듀얼코어에서는 2배 이상 빨라지지 않는다. 듀얼코어인데 6배 빨라졌다? 그건 싱글 스레드 프로그램이 엉터리로 짜인 것이다. 잘못 짠 부분이 수정이 되어 빨라진 것이지 멀티스레드 때문에 빨라진 것이 아니다.

- 멀티 CPU 컴퓨터에서의 프로그램 성능 향상 (O)

멀티CPU는 무엇인가? 컴퓨터 뚜껑을 열고 보면 CPU가 1개가 아니라 2개 4개 꼽힌 것이 있다. 이것을 여러 개 활용해서 성능을 높이고 싶다면 멀티스레드를 쓰면 된다. 멀티코어 CPU랑 멀티 CPU랑 하드웨어 입장에서는 다르지만, 프로그래밍하는 입장에서는 똑같아서 구분이 안 간다. 멀티스레드를 하면 둘 다 커버가 된다. CPU가 여러 개 있고 그 안에 코어가 여러 개. 이런 복잡한 환경에서 어떻게 구현하는가? 신경 안 쓰고 그냥 하던 대로 하면 된다.

- 분산 컴퓨터에서의 프로그램 성능 향상 (X)

분산 컴퓨터에서의 프로그램 성능 향상? 안된다. 왜? 멀티스레드는 스레드를 만드는 것. 그건 같은 프로세스 안에서 돌아가는 것이다. 다른 컴퓨터에서 돌릴 수 없다. 내가 만든 스레드를 옆에 컴퓨터에서 돌리는 것? 있긴한데 연구용이라 우리가 쓰진 않으므로 없다. 윈도우나 리눅스엔 그런 게 없다는 것이다.

- 싱글코어(CPU) 컴퓨터 환경에서의 프로그램 성능 향상 (X)

싱글코어 컴퓨터에서 성능을 높이자. 이건 완전 헛소리이다. 가끔가다 빨라졌다는 것은 싱글 스레드 프로그램이 엉터리였다는 것이다. 제 성능을 이미 내고 있지 않았기 때문에 현재 빨라지는 것처럼 보이는 것이다. 그럼 애초에 그 싱글 프로그램을 다시 짜야한다. 그러면 멀티스레드로 짰던 것보다 빠를 것이다. 왜? 멀티스레드에서만 성능 향상이 있는 게 멀티스레드 프로그래밍이기 때문이다.

- 프로그램을 모듈화해서 알아보기 쉽게 하기 위해 (X)

메인 루프에서 하나만 호출해서 돌리는 것이 지저분하다고 느껴질 수 있다. 물리엔진 스레드, 사운드 스레드, 렌더링 스레드, 나누어서 돌리는 게 훨씬 깔끔하고 독립적이어서 디버깅도 쉽고 좀 더 고차원적인 개념 아닌가? 절대 아니다. 이런 짓 하지 마라. 이런 것을 보고 겉멋이라고 부른다. 절대로 그러지 마라. 멀티코어 프로그래밍은 성능 향상을 위해 쓰는 것이지 이게 아니다. 모듈화 되고 독립적이고 알기 쉽게 되는 것 맞다. 그러나, 멀티스레드 프로그램을 해서 페널티 위험성을 고려했을 때 얻는 것보다 잃는 것이 훨씬 더 많다. 그러니 **절대로 성능 이외의 다른 이유로 사용하지 마라**. 작은 프로그램은 한 눈에 볼 수 있어서 관촬을지라도, 커뮤니케이션이 많아지면 성능이 확 떨어지거나 확 죽어버리거나 하게 된다.

**결론적으로, 병렬컴퓨터에서의 프로그램 성능 향상**을 위해서만 쓴다. 멀티코어가 되었던 멀티 cpu가 되었던 거기서 성능 향상을 하기 위해 쓰는 게 멀티스레드이다. 최근 쓰는 것은 다 병렬 컴퓨터니까 프로그램 성능 향상의 이유만 생각하면 된다.

## 2. 멀티코어 CPU

- 한 개 이상의 코어로 구성된 CPU를 뜻한다.
  - 우리가 쓰고 있는 컴퓨터 i3, i5, i7
  - 갤럭시 S8, iPhone X, PlayStation 4
- 현재 멀티코어가 아닌 CPU가 존재하는가?

제주도에 있는 넥슨 컴퓨터 박물관에 가보면 싱글코어 CPU로 돌아가는 컴퓨터를 구경할 수 있지만 돈 주고 살 수는 없다. 그리고 느리다. 윈도우 10도 안 돌아간다.

그리고 게임기 플스, Xbox, 닌텐도 ds, 스위치 다 멀티코어이다. 다 멀티코어. 아닌 컴퓨터가 없다. 작정을 하고 찾아보면 있긴 하지만 어떤 게 있나? 키보드가 있다. 키보드 안에 cpu가 들어있지만 성능이 필요가 없고 그걸 프로그래밍하지 않으므로 신경 쓰지 않는다.

---

## 3. Intel과 ARM에서 멀티코어 프로세서를 만드는 이유

- CPU의 성능을 올려야 한다.
  - 안 그러면 아무도 안 산다
- 클럭 속도를 높일 수 없다
  - 발열 문제 (물리법칙)
- 클럭 속도 말고 CPU의 속도를 올리는 것은?
  - 한계에 부딪혔다. (아키텍처 개선 : 캐시, 파이프라인, 예측 분기, 동적 수행 등)
- 남은 방법
  - 멀티코어

- CPU의 성능을 올려야 한다.

왜 굳이 싱글코어 CPU를 팔지 않고 멀티코어를 파는지 궁금할 수 있다.

왜 싱글코어를 팔지 않는가? 반값으로 팔면 그래도 살 사람이 있지 않을까? 인텔도 그렇고 최하 듀얼코어 이런 것도 보기 힘든데 왜 쿼드코어만 주구장창 파는가? 왜? CPU의 성능을 올려야 하기 때문이다.

Intel과 ARM은 CPU를 팔아서 먹고사는 회사이다. 망하는 방법은 사람들이 CPU를 안 사면 망한다. 사람들이 컴퓨터가 필요해서 사는 시대는 이미 지났다. 요즘엔 컴퓨터가 없는 집이 없다. 요즘엔 느려서, 고장 나서 그래서 CPU를 사는 것이다. 둘 중에 느려서 사는 경우가 더 많다. 그런데 사려고 봤는데 새로 나온 CPU가 지금 쓰는 CPU보다 느리면 사람들은 업그레이드를 하지 않는다. 그래픽카드를 바꾸지 CPU를 바꾸지 않는다. 왜? CPU를 바꾸려면 본체를 통째로 바꾸어야 하니까. 그래서 CPU의 성능을 계속 올리지 않으면 회사가 망한다. 그래서 멀티코어 CPU를 만드는 것이다.

싱글코어는 아무도 안 산다. 지금 CPU보다 느린데 아무리 싸도 더 느린 것으로 바꿀 사람은 없다. 새로 살 때나 고려하겠지만, 그러나 그런 경우에도 싸게 팔게 되면 마진이 안 남는다. 일부러 안 만드는 것도 있다.

- 클럭 속도를 높일 수 없다

근데 싱글코어를 그냥 더 빠른 싱글코어로 만들면 되지 않냐, 왜 멀티코어만 성능을 올리냐 궁금할 수 있다. 싱글코어는 가장 빠른 게 클럭 속도를 높이는 것이다. 2 GHz CPU보다 4 GHz CPU가 더 빠르다. 근데 그게 안된다. 클럭 속도를 더 이상 높일 수 없다. 교수님은 2002년도에 NC에 취직해서 첫 월급을 타고 최신 사양 컴퓨터를 맞추었다고 한다. 스타크래프트를 하려고. 그때 최신 사양 컴퓨터 Pentium 4는 클럭 속도가 2.4GHz 였다고 한다. 3 GHz까지 맞출 수 있었지만 돈이 많이 들어 2.4로 하셨다고. 지금 우리 컴퓨터가 3 GHz정도 된다. 20년이 흘렀지만 그때 왜 지금은 거의 클럭 속도가 비슷하다. 높일 수가 없다. 4 GHz로 높이면 컴퓨터가 타버린다. 이것은 물리법칙이다. 애당초 우리 우주가 그렇게 설계되어있다.

클럭 속도를 높이려면 열을 잡아야 하고 그럼 액체질소를 실시간으로 들이부어야 한다. 그래야지 5GHz를 넘을 수 있고 4 GHz정도 한다 하더라도 배보다 배꼽이 더 큰 냉각장치를 써야 한다.



- 클럭 속도 말고 CPU의 속도를 올리는 것은?

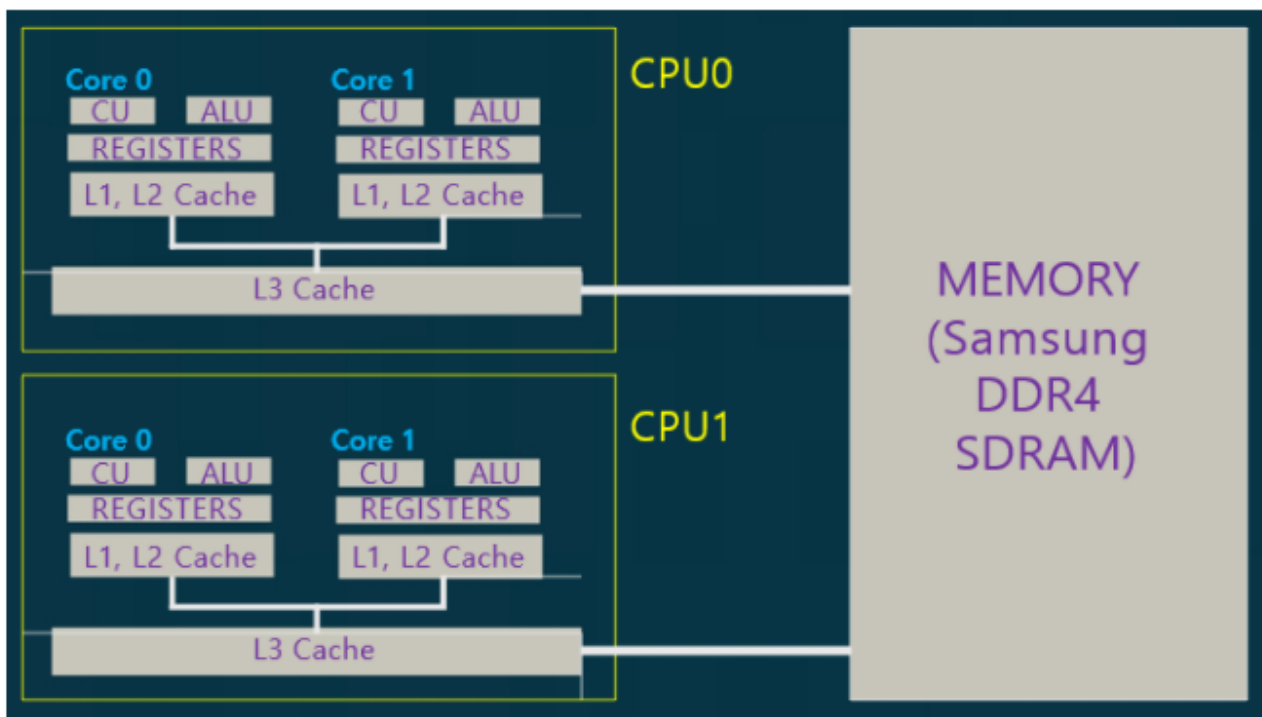
클럭속도를 못 높인다면 다른 방법은? 파이프라인을 새로 만들고 동적 수행을 하고... 등등 싹 다 쓰면 빨라지지 않느냐. 빨라진다. 그런데 그것도 이미 한계에 부딪혔다. 경제 시간에 한계효용의 법칙을 배웠을 것이다. 어떡 나 물건의 가치가 있을 때 투자를 하면 할수록 가치가 올라가지만, 어떤 수준 이상으로는 오르지 않는다. 1천 원과 2천 원은 차이가 안 나는데 1천 원과 1만 원은 차이가 난다. 비쌀수록 처음에는 점점 맛있다가 나중에는 별 차이가 없어진다. 10만 원과 20만 원은 2배 차이가 나지만 맛 차이가 2배가 나는 것이 아니다. CPU도 마찬가지이다. Intel과 AMD는 이미 온갖 것 다 집어넣고 대학생 논문이 새로 나오면 달려들어 구현하고 하면서 갈 때까지 갔다. 지금 CPU에서 캐시를 높인다고 하더라도 2% 정도 빨라진다. 파이프라인을 쪼개면? 이미 쪼갤 만큼 쪼갰고 그 이상 쪼개면 오히려 느려진다. 예측 분기 이미 하고 있고 동적 수행 이미 하고 있다.

- 남은 방법

그래서 남은 것은 멀티코어밖에 없다. 그래서 AMD에서 16 코어를 내보내고 그러는 것이다. 그래서 Intel도 12 코어 10 코어 만들어서 팔고 있다. 왜 이렇게 하는가? 싱글로는 더 이상 안되니까 코어 개수로 싸우는 것이다. 코어의 개수가 늘어나지 않는 경우 성능의 개선이 별로 없다. 그러니까 작년에 Intel에서 팔던 cpu랑 올해 파는 cpu, 같은 코어 개수를 사면 성능 변화가 거의 없다. Intel에서는 10% 20% 개선된다고 홍보하지만, 그건 그렇게 벤치 마크된 프로그램의 이야기다. 우리가 쓰는 건 그렇지 않다.

#### 4. 멀티코어 프로세서

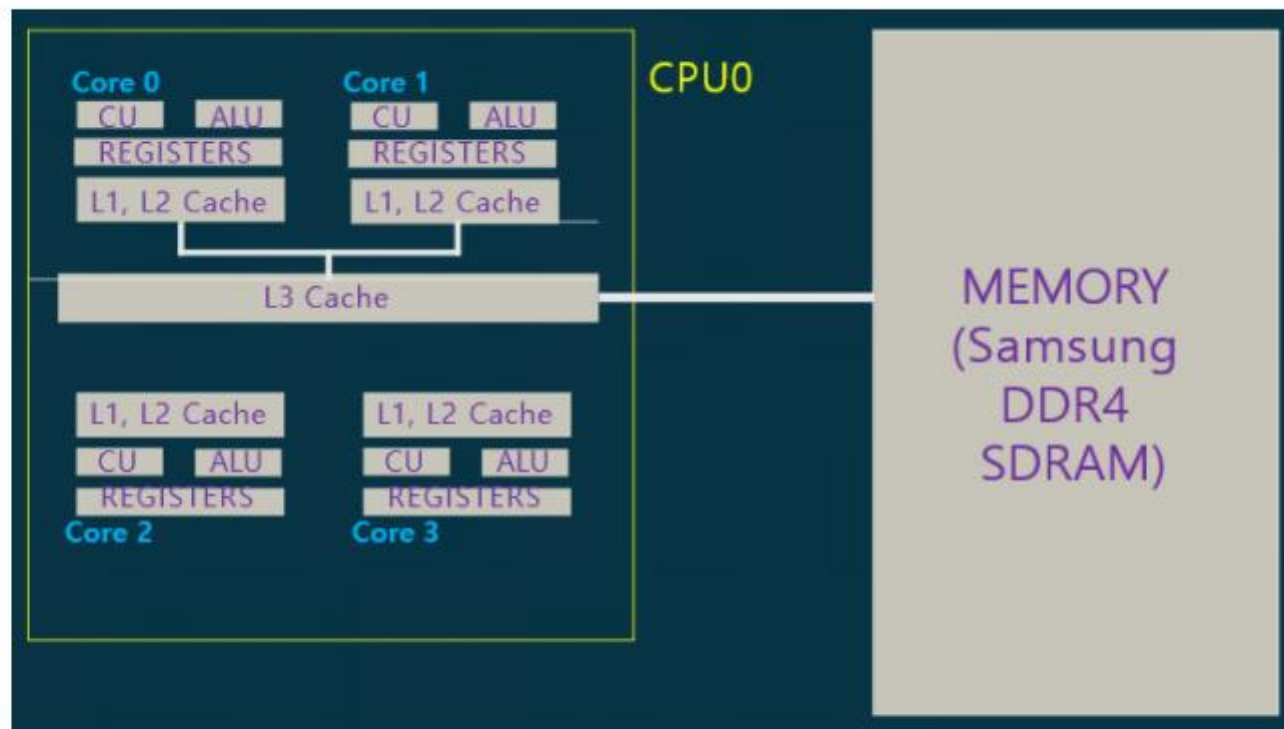
- Dual Core - 2 CPU



어쩔 수 없이 멀티코어 코어 개수를 높여 성능을 올려야 한다. 멀티코어 CPU는 어떻게 생겼나? 이렇게 생겼다. CPU0은 듀얼코어 CPU이다. 코어는 ALU, REGISTER, CU(제어 유닛) 그리고 캐시로 이루어져 있다. 어? 이거 다 CPU이다. CPU를 만들려면 ALU, CU, REGISTER 이 3개가 있어야 한다. 인텔에서 CPU 하나만 들어서 복사 붙여 넣기 하면 듀얼코어가 되고 그걸 또 복붙 하면 쿼드코어, 옥타코어가 되는 것이다. 진짜로 이렇게 한다. 하나하나 반도체 회로 설계하고 절대 그렇지 않다. 진짜 컨트롤 C, 컨트롤 V 한다. 개발비가 공짜다. 그냥 찍어내기만 하면 된다. AMD도 마찬가지이다.

CPU0, CPU1은 멀티 CPU이다. 똑같은 CPU를 꽃아서 사용하니 ID만 0, 1 붙는 것이다. 멀티코어 CPU를 두 개 꽃은 것이다. 이게 Quad Core - 1 CPU 밑에 있는 그림과 무엇이 다를까?

- Quad Core - 1CPU



쿼드코어 CPU를 꼽은 것이다. 듀얼코어 CPU 2개, 쿼드코어 CPU 1개, 차이는 없다. 걱정하고 달려들면 차이는 알 수 있다. 메인보드에서 CPU가 몇 개 꽂혀있는지 확인하는 것. 근데 그거 말고 프로그램으로는 구분이 안된다. 억지로 한다 하면 L3 캐시를 공유하니 느려지고... 하는 등의 것을 속도를 측정하면 짐작할 수 있겠지만 그것 말고 프로그래밍엔 아무 의미 없다. 4배까지 성능을 올릴 수 있다는 것은 같다.



- 왜 이렇게 늦게 나왔는가?
  - 프로그래머가 싫어하기 때문
    - 프로그램을 다시 작성하지 않으면 성능 항상 "0"
    - 전혀 다른 알고리즘을 사용해야 한다.
    - 디버깅이 어렵다
  - 옛날부터 있었다.
    - 과학기술 계산용 : 기상예측, QCD(원자핵 사이의 강력 계산), 유체역학(비행기, 잠수함 계산), N-body problem(은하계끼리 충돌했을 때 계산)
      - 수출 금지 품목
      - 무지 비쌌다
    - CRAY X-MP(1982), Cmpaq Proliant 1500(1995)


멀티코어는 2005년 정도 나왔다. 2000년대 초반 정도. 근데 인텔은 cpu를 1930년대부터 만들었으면서 왜 늦게 나왔냐? 프로그래머가 정말 싫어하기 때문이다. 프로그램을 다시 짜는 게 매우 어렵고, 디버깅도 어렵고, 배운 알고리즘도 못쓴다. 그래서 인텔이 70년대에도 만들 수 있었지만 안 팔려서 안 만들었다. 근데 지금은? 어쩔 수 없어서. 벼랑 끝에 몰려서. 더 빠른 cpu를 팔지 않으면 망하고, 클럭 속도는 못 올리고. AMD는 Intel에 지지 않기 위해 8 코어 16 코어를 내놓고.

그래서 이렇게 늦게 나왔다. 스타 1은 싱글 스레드 프로그램이다. 듀얼코어에서 돌린다고 2배 이상 빨라지지 않는다. 유닛을 많이 만들면 그래서 버벅거린다. 싱글 스레드 프로그램이기 때문에 듀얼코어에서도 똑같이 버벅거린다. 코어를 하나밖에 못쓰기 때문이다.

지금에서야 프로그래머들이 어쩔 수 없이 쓰지만, 멀티스레드 또는 멀티코어 컴퓨터는 옛날부터 있었다. 일반인들은 안 쓰는 특수한 용도에. 기상예측 슈퍼컴퓨터, QCD 원자핵 사이의 강력 계산, 유체역학 비행기와 잠수함 계산, N-Body problem 은하계끼리의 충돌 계산 이런 거. 그런데 이걸 특수 컴퓨터, 슈퍼컴퓨터로 불리는 것들이라 무척 비쌌다. 몇 억, 10억 넘어간다. 우리나라에 80년대엔 1대뿐이었다.

우리나라는 CRAY X-MP는 돈 없어서 못 사고 87년도에 그다음 버전을 샀었다. Cmpaq Proliant 1500 이걸 리니지 1이 처음 서비스될 때 그땐 싼 컴퓨터에서 멀티 cpu로 서버 만들자 해서 이걸 사서 했었다. Intel cpu가 펜티엄 2개인가 3을 4개 꽂은 컴퓨터였다. 코어가 4개. 20년 전 리니지의 클라이언트 소스코드는 싱글코어였다. 멀티코어는 손톱만큼도 없었다고. (교수님께서 직접 보셨다고 하셨다) 그런데 서버는 처음부터 멀티코어로 돌아갔다고.

- 현재
  - 4 Core (Quad Core)가 대세
    - 2017년부터 코어 개수 급속 증가
  - 인텔 제온 : 22 Core Processor



**■ 인텔 제온 스케일러블 골드 6152 (스카이레이크)**

인텔(소켓3647) / 14nm / 22코어 / 캐시 44개 / 2.10GHz / 30.25MB / 140W / 하이퍼스레딩

등록일 2017.09 | 상품유형 45건 | 관심상품

정품 **4,806,450원** 40원

올해(2019) 초에 56 코어짜리 제온 cpu가 나왔다. 가격은 cpu 1개에 5천만 원 정도. 30 코어는 천만 원 정도 한다. 22 코어는 5백만 원 정도.



이런 흉악한 물건도 있었다. 세계에서 제일 빨랐던 중국 슈퍼컴퓨터에서도 사용되었으나 미국이 수출 금지를 시켰었다. 최하 3천만 원이었고, 멀티스레드 프로 그래밍을 해야 하는데 다른 방식으로 따로 했어야 했다. 윈도에서 지원 안 함. 언리얼에서 지원 안 함. 우리가 쓸 일은 없고 어디에 쓰나? 슈퍼컴퓨터 만들 때 쓴다. 작년까지만 해도 지구 상에서 제일 빠른 컴퓨터가 중국에 있던 텐허 2에서 이것을 썼었다. 그런데 미국의 트럼프가 수출 금지를 시켰었다. 그러자 중국에서 자체 개발한 cpu를 사용해서 컴퓨터를 만들었고, 절대 속도는 빠르는데 그 컴퓨터에서만 빠르게 돌아가도록 특수 제작된 프로그램만 빠르지 제대로 된 슈퍼컴퓨터 칩을 사용하지 않았으니 사람들은 진짜 빠를 거라 믿지 않는다. AMD는 32까지, Intel은 72까지 코어를 올렸다.

- 게임 콘솔

Microsoft Xbox One vs. Sony PlayStation 4 Spec comparison				
	Xbox 360	Xbox One	PlayStation 4	Mine
CPU Cores/Threads	3/6	8/8	8/8	4/8
CPU Frequency	3.2GHz	1.6GHz (est)	1.6GHz (est)	3.8GHz
CPU $\mu$ Arch	IBM PowerPC	AMD Jaguar	AMD Jaguar	Intel i7
Shared L2 Cache	1MB	2 x 2MB	2 x 2MB	3 x 8MB
GPU Cores		768	1152	576 x 2
Peak Shader Throughput	0.24 TFLOPS	1.23 TFLOPS	1.84 TFLOPS	1.26 TFLOPS x 2
Embedded Memory	10MB eDRAM	32MB eSRAM	-	1GB x 2
Embedded Memory Bandwidth	32GB/s	102GB/s	-	340GB/s
System Memory	512MB 1400MHz GDDR3	8GB 2133MHz DDR3	8GB 5500MHz GDDR5	24GB 2133MHz DDR3
System Memory Bus	128-bits	256-bits	256-bits	256-bits
System Memory Bandwidth	22.4 GB/s	68.3 GB/s	176.0 GB/s	176.0 GB/s
Manufacturing Process		28nm	28nm	45nm

[https://www.reddit.com/r/gaming/duplicates/1g7gze/fixed\\_microsoft\\_xbox\\_one\\_vs\\_sony\\_playstation\\_4\\_vs/](https://www.reddit.com/r/gaming/duplicates/1g7gze/fixed_microsoft_xbox_one_vs_sony_playstation_4_vs/)

pc 말고 게임기는 어떻게 생겼을까? 게임기도 멀티코어다. 오히려 데스크톱보다 코어가 많았다.

최신이라고 해봤자 컴퓨터는 4개인데 게임기는 왜 8 코어씩 쓰나? 이유는 클럭 속도가 컴퓨터의 반토막이다. 데스크톱 pc보다 속도가 반밖에 안된다. 그러니 성능이 pc보다 반밖에 안된다. 성능 격차 메꾸는 방법은 코어를 2배 올리는 것. 그러니까 쿼드코어랑 성능이 많이 비슷해졌다. 그래도 데스크톱보다 빠르진 않다. 근데 왜 클럭 속도가 이 모양이냐? 이유는 열 때문에. 3기가 클럭 넣어도 잘 돌아간다. 근데 그러면 게임기에 fan을 달아야 한다. 어쩔 수 없이 클럭 속도를 낮췄고 성능이 안 나와서 코어를 높인 것이다. 콘솔 회사에서 일하려면 멀티코어를 열심히 공부해야 한다. 코어를 다 활용하지 않으면 성능이 안 나오기 때문이다.

수업을 듣다 궁금했던 점이 생겼었다. 메인 스레드 이외엔 스레드를 Create 해야만 늘어난다고 배웠는데, 만들 때 듀얼코어를 대상으로 만들면 쿼드코어에서는 2배 이상의 성능이 안나는 건가? 근데 그러면 또 코어 개수에 맞춰서 다시 프로그래밍해야 하는 건가? 22 코어 56 코어 나올 때마다? 하는 질문이었다

교수님께 질문했더니 간단하게 그래서 loop를 돌려서 Create 한다고 대답해 주셨다 ^^..

또 궁금했던 것은 프로세스 밑에 스레드가 있고 그 스레드가 여러 개 있을 수 있다면,, 그럼 프로세스당 최소 하나의 스레드가 있는 건데 쿼드코어여도 몇십 개의 프로세스가 돌아갈 수 있는 게 빠르게 번갈아가면서 실행이 되어서 가능하다 하셨는데.. 그럼 스레드는 중간에 종료될 수 없으니까 한 스레드가 끝나야 다른 프로세스의 스레드가 돌아가게 순서가 되는 건지...??

이것도 교수님께 질문했더니 아니다. 스레드는 중간에 종료될 수 있고, 강제로 종료되는 것이 안 되는 것. 그리고 프로세스가 시분할로 돌아가면서 실행되듯 스레드도 시분할로 운영되어서 중간에 다른 스레드가 실행될 수 있는 것. 그래서 코드 상 선언 한 스레드 개수와 코어의 개수는 일치하지 않아도 상관없이 되는 것이다.라고 대답해 주셨다