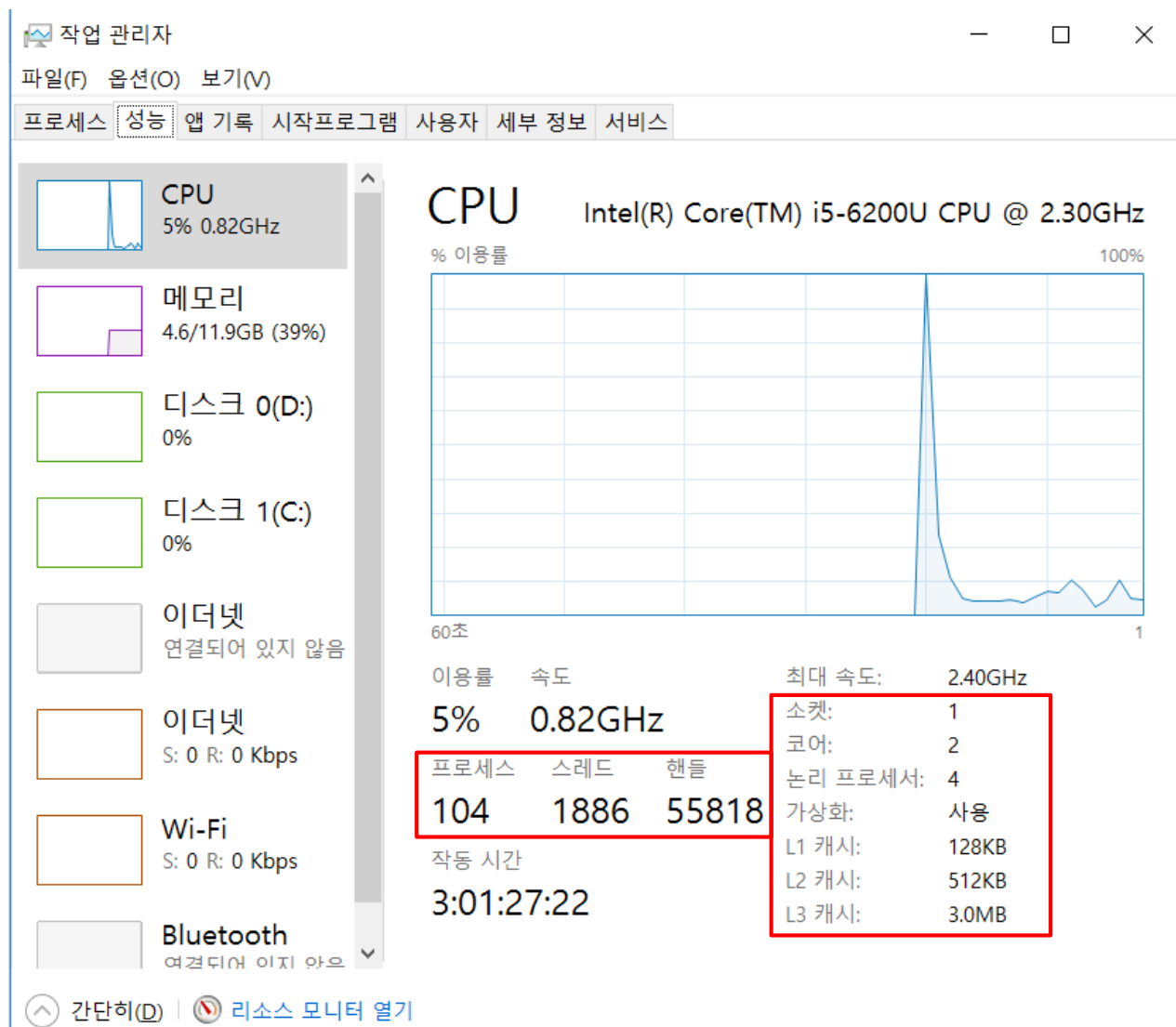


CS SCHOOL

CPU



CPU

Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz

- 클럭(clock)이란?

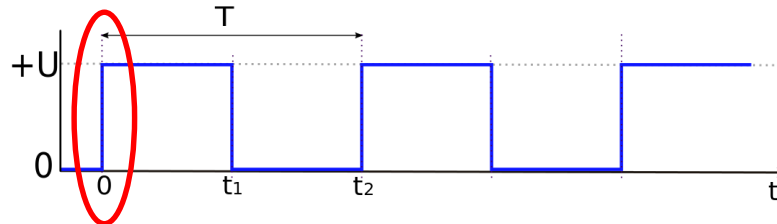
$$\tau = \frac{1}{f}$$

1초에 2,300,000,000번!

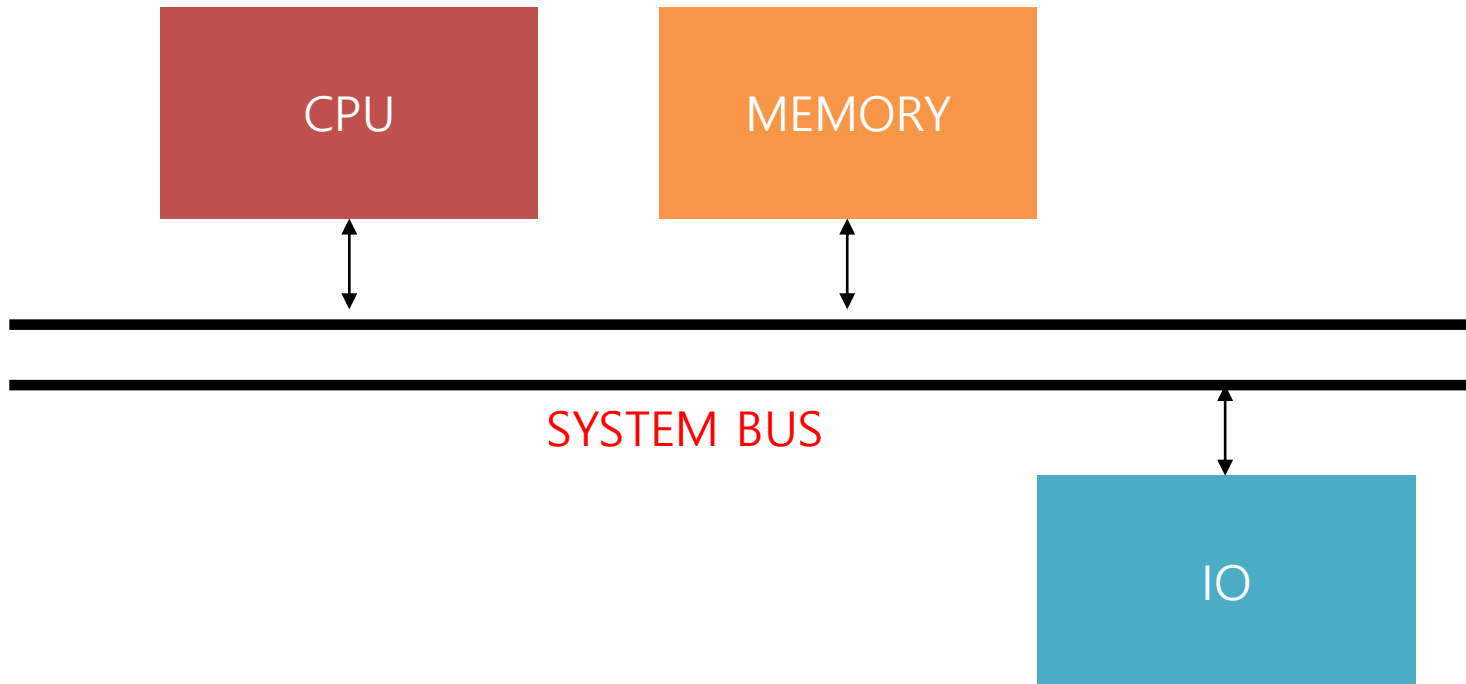


0.43 ns

상승할 때 연산 처리



$$\text{CPU clock} = \text{base clock} \times \text{multiplier}$$

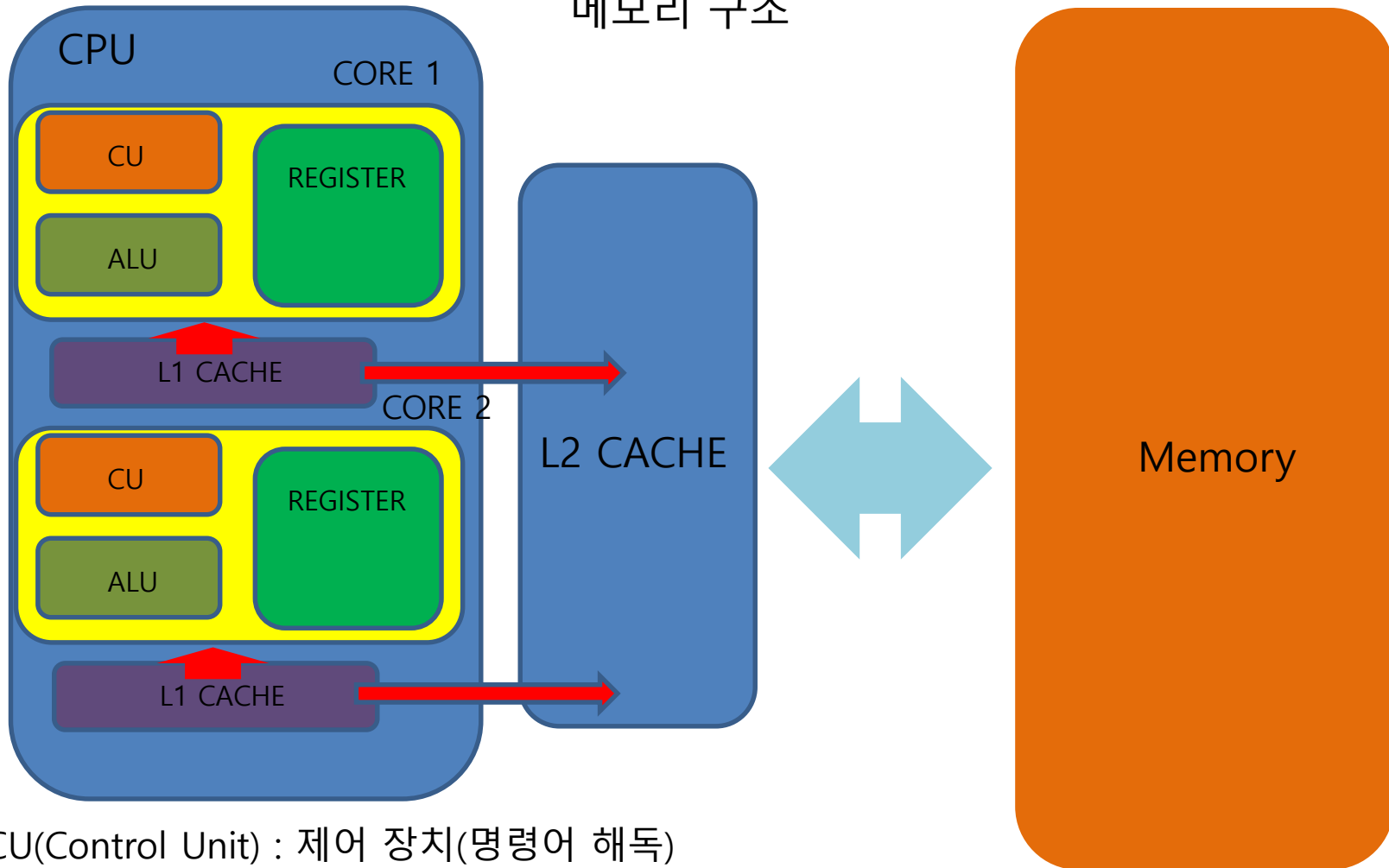


$\text{CPU speed} = \text{system bus speed} \times \text{multiplier}$

CACHE란?

이용률	속도		최대 속도:	2.40GHz
5%	0.82GHz		소켓:	1
			코어:	2
프로세스	스레드	핸들	논리 프로세서:	4
104	1886	55818	가상화:	사용
작동 시간			L1 캐시:	128KB
3:01:27:22			L2 캐시:	512KB
			L3 캐시:	3.0MB

메모리 구조



CU(Control Unit) : 제어 장치(명령어 해독)

ALU(Arithmetic Logic Unit) : 연산 장치(산술 연산, 논리 연산)

Register : 작지만 가장 빠른 메모리

L1 CACHE(각 코어마다 독립적), L2 CACHE(코어가 공유하는 공유 메모리)

Cache hit, Cache miss

1. CACHE HIT

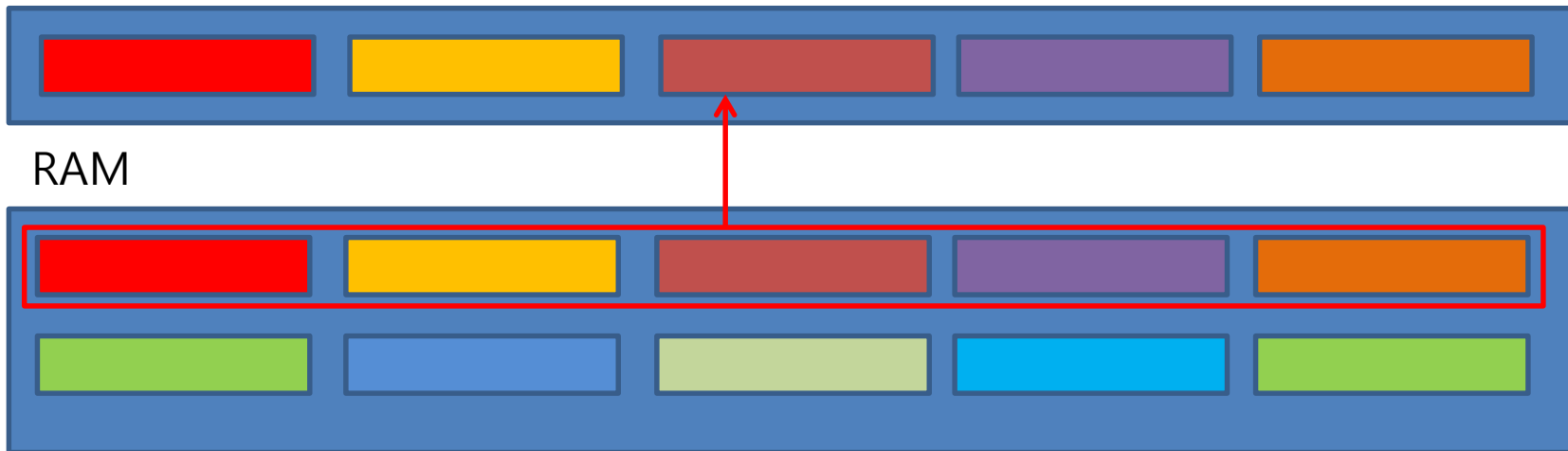
- CPU가 메모리를 요청하면 CACHE가 메모리를 라인으로 읽는데 CPU가 다음에 요청한 메모리가 이미 캐시 안에 있을 경우

1. CACHE MISS

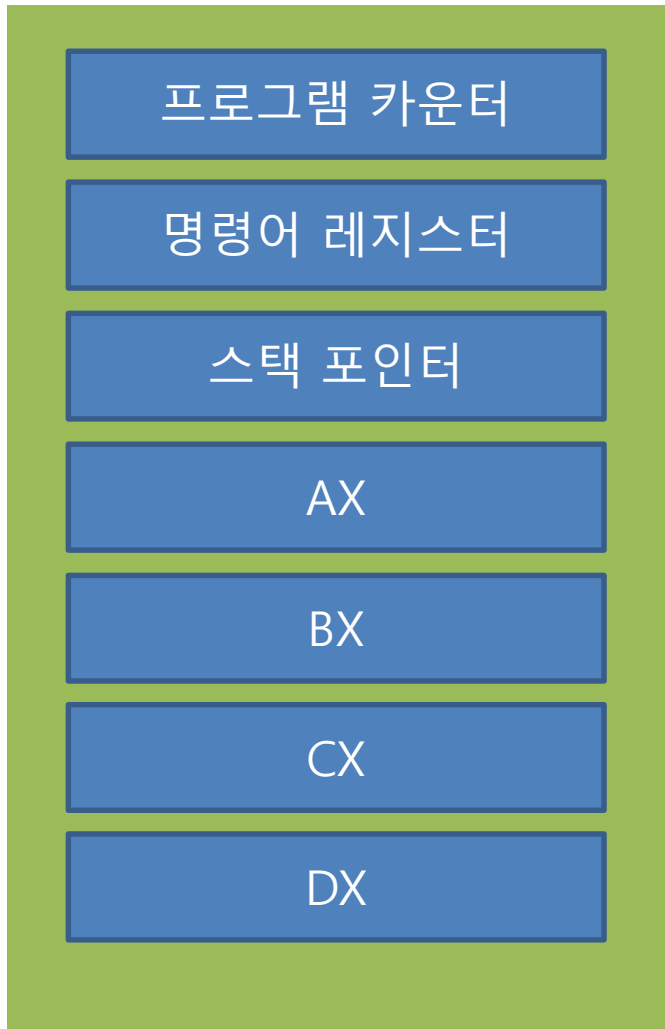
- CPU가 메모리를 요청하면 CACHE가 메모리를 라인으로 읽는데 CPU가 다음에 요청한 메모리가 캐시 안에 없어서 다시 RAM에서 읽어와야 할 경우

64 byte ~ 128 byte

CACHE : 캐시 메모리는 메모리를 읽어올 때 일정 라인 단위로 읽어옵니다.



Register 구조



다음 실행할 명령어의 주소

지금 실행할 명령어

스택을 가리키는 포인터

범용 레지스터
:메모리에서 데이터를 읽어와 저장

MOV AX [0x1234] : [0x1234]에서 AX로 이동

ADD AX BX : AX와 BX를 더해 AX에 저장

AND AX BX : AX와 BX를 AND 연산하여 AX에 저장

메모리 영역

코드 영역

프로세스 실행 후
실행코드 로드

(프로세스 실행 시)

데이터 영역

전역변수
static 지역변수

프로세스 실행 시
로드 되어
프로그램 종료 시
해제

스택 영역

지역변수
매개변수

스코프
{...}

내에서만 생성 후

스코프를 벗어나면
해제

힙 영역

프로그래머의 영역

프로그래머가
할당하면

프로그래머가
해제하기 전까지
사라지지 않는다.

프로그램, 프로세스, 스레드

이용률 속도
5% 0.82GHz

프로세스 스레드
104 1886

핸들
55818

작동 시간
3:01:27:22

최대 속도: 2.40GHz
소켓: 1
코어: 2
논리 프로세서: 4
가상화: 사용
L1 캐시: 128KB
L2 캐시: 512KB
L3 캐시: 3.0MB

프로그램과 프로세스

1. 프로그램(Program)

: 실행시키기 전 하드디스크에 저장되어 있는 실행파일

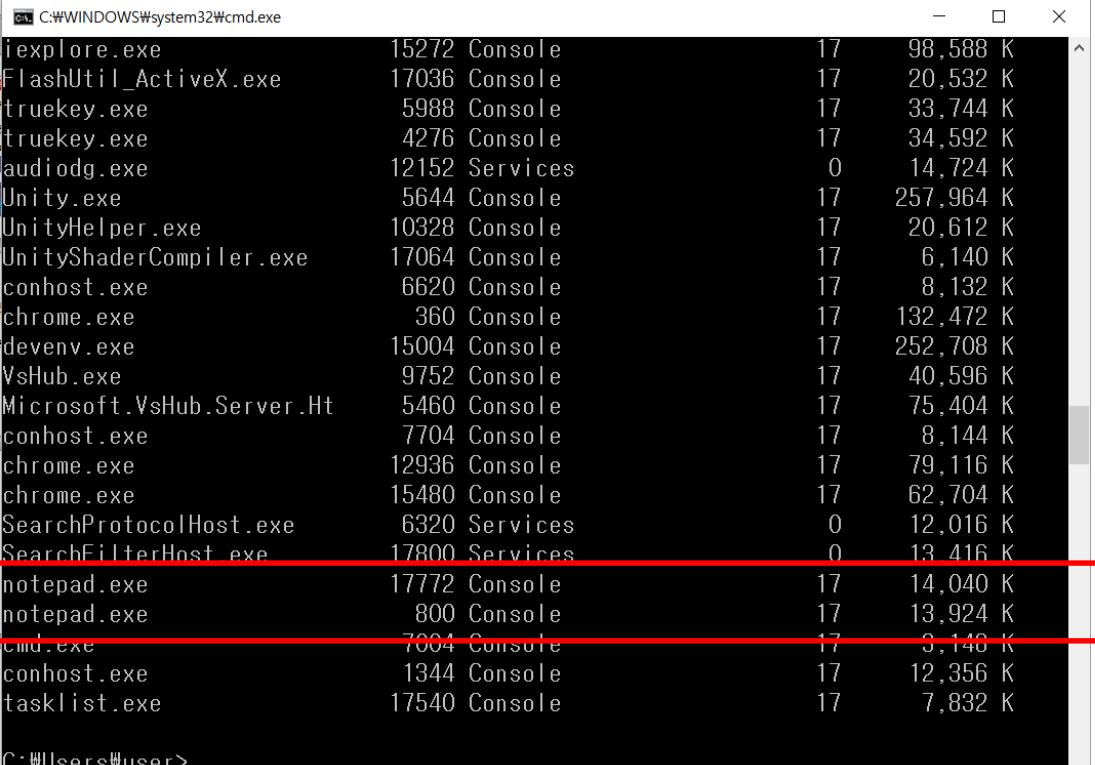
-> 프로그램은 하드디스크에 하나!

2. 프로세스(Process)

: 실행시킨 후 메모리에 올라와서 실행 중인 코드

-> RAM 상에 여러 개 존재 가능!

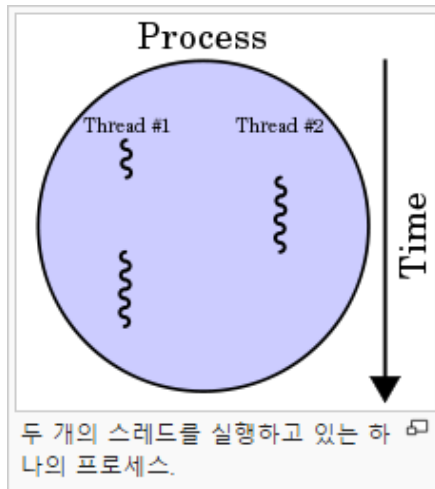
메모장 프로그램은 하나
실행 중인 메모장 프로세스는 두 개



Process Name	PID	Session ID	Architecture	Working Set (K)
explorer.exe	15272	Console	17	98,588 K
FlashUtil_ActiveX.exe	17036	Console	17	20,532 K
truekey.exe	5988	Console	17	33,744 K
truekey.exe	4276	Console	17	34,592 K
audiodg.exe	12152	Services	0	14,724 K
Unity.exe	5644	Console	17	257,964 K
UnityHelper.exe	10328	Console	17	20,612 K
UnityShaderCompiler.exe	17064	Console	17	6,140 K
conhost.exe	6620	Console	17	8,132 K
chrome.exe	360	Console	17	132,472 K
devenv.exe	15004	Console	17	252,708 K
VsHub.exe	9752	Console	17	40,596 K
Microsoft.VsHub.Server.Ht	5460	Console	17	75,404 K
conhost.exe	7704	Console	17	8,144 K
chrome.exe	12936	Console	17	79,116 K
chrome.exe	15480	Console	17	62,704 K
SearchProtocolHost.exe	6320	Services	0	12,016 K
SearchFilterHost.exe	17800	Services	0	13,416 K
notepad.exe	17772	Console	17	14,040 K
notepad.exe	800	Console	17	13,924 K
cmd.exe	7004	Console	17	9,148 K
conhost.exe	1344	Console	17	12,356 K
tasklist.exe	17540	Console	17	7,832 K

C:\Users\User>

프로세스와 스레드(thread)



출처 : 위키백과

프로세스 : 메모리 상에서 실행되는 프로그램

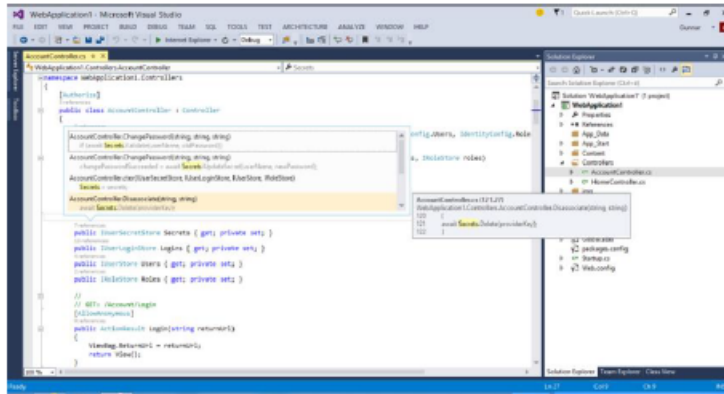
-> 완전히 독립적인 메모리를 가진다.

스레드 : 프로세스 내 실행 흐름의 단위

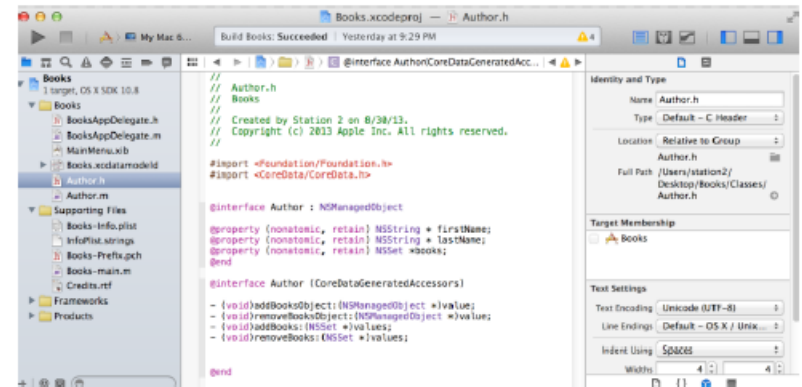
-> 데이터, 힙, 코드 영역을 공유

공유자원에 의해 Critical section이 생기고
이로 인해 경합조건(Race Condition)이 발생

IDE(Integrated Development Environment : 통합 개발 환경)



Visual Studio



X-Code

1. Text Editor
2. Debug Mode
3. 컴파일러 내장
4. 프로젝트 관리
5. 오류 내용 확인

1. 줄 번호 지정

Tools -> options -> Text Editor->C/C++ -> General -> Line numbers

2. 언어 설정

Tools -> options -> Environment -> International Settings

3. 콘솔 프로젝트 만들기

- 1) 콘솔 프로젝트로 시작(Create directory for solution 체크 해제)
- 2) Console application
- 3) Empty project

4. 컴파일러 선택

New Item 생성시 확장자가 .c : C compiler
 확장자가 .cpp : C++ compiler

프로그래밍 언어의 종류

컴파일러의 유무에 따라

결국 마지막에 컴퓨터에 명령을 내리는 것은 **기계어**

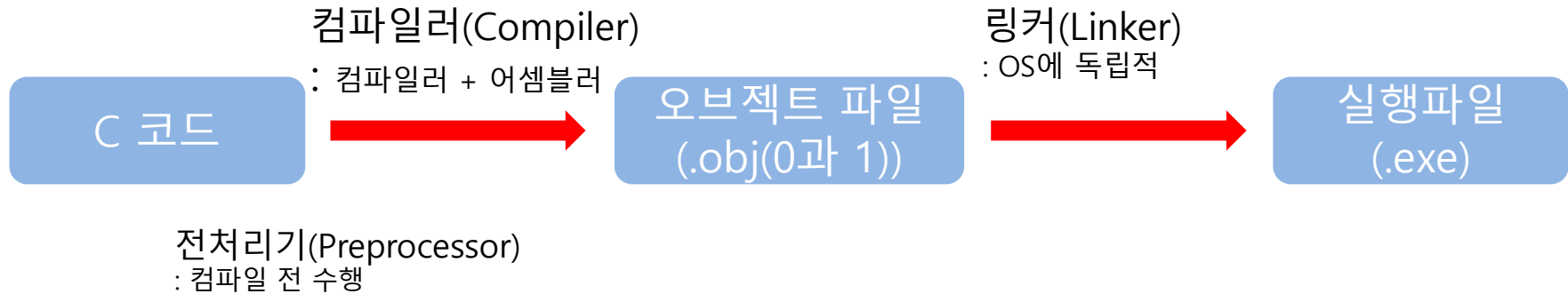
1. 컴파일러 언어

- 기계어로 미리 만들어 두었다가(Compile time)
→ 프로세스 시작하면 만들어 둔 기계어를 돌린다(Run time)

2. 인터프리터 언어

- 프로세스 시작하면 코드 한 줄을 만날 때 마다
→ 그에 대응하는 기계어를 찾아내 실행

프로그래밍 과정



2진수와 16진수

1. 10진수
: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
2. 2진수
: 0, 1
3. 16진수
: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f

- 진수 변환

1. 10진수 -> 2진수

예) $25 \rightarrow 16 + 8 + 1 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0$
 $= 11001$

2. 2진수 -> 16진수 (2진수 4자리가 16진수 1자리 수)

예) $1001 \rightarrow 8 + 1 = 9$

$1010 \rightarrow 8 + 2 = 10 = a$

$1111 \rightarrow 8 + 4 + 2 + 1 = 15 = f$

변수(Variable)

```
int num = 15;
```

(이 한 문장 안에 엄청난 의미가 담겨있다)

상수도 스택 영역에 할당된 후 다음 행에서 소멸
호출할 수단이 없으므로

'같다'가 아닌 대입

1. 변수 선언 위치에 따라 메모리 저장 위치와 생성 소멸 시기가 결정

```
1  #include <stdio.h>
2
3  int num = 5; //전역변수
4
5  void Func(int n) //매개변수
6  {
7      int num = 5; //지역변수
8  }
```

자료형(Data type)

구분	자료형	범위	바이트
정수형	char	-128 ~ 127	1(8)
	unsigned char	0 ~ 255	1(8)
	short	-32768 ~ 32767	2(16)
	int	-2,147,483,648 ~ 2,147,483,647	4(32)
	long	-2,147,483,648 ~ 2,147,483,647	4(32)
	unsigned short	0~65535	2(16)
	unsigned int unsigned long	0~4,294,967,295 0~4,294,967,295	4(32) 4(32)
실수형	float	$8.4 \times 10^{-37} \sim 3.4 \times 10^{38}$	4(32)
	double	$2.2 \times 10^{-308} \sim 1.8 \times 10^{308}$	8(64)
나열형	enum	정수를 대신하여 사용하는 별명, int형의 크기	
무치형	void	실제 자료는 없음을 명시적으로 선언	

2의 보수(two's complement)

: 컴퓨터가 수를 저장하는 방식1
- 정수형

1. 2의 보수란?

$$\begin{array}{r} 101011 \\ + \textcolor{red}{010100} \text{ (101011의 1의 보수)} \\ \hline 111111 \text{ (자리수가 바뀌기 전의 수)} \end{array}$$

$\textcolor{red}{010101}$ (101011의 2의 보수)

2의 보수 만드는 방법
: $\textcolor{red}{1\text{의 보수}} + 1$



2의 보수(two's complement)

: 컴퓨터가 수를 저장하는 방식1
- 정수형

2. 정수를 저장하는 방식

char형의 경우

1) 양의 정수

0 000 1010 10진수 : 10

sign number

그렇다면 음의 정수 -10은

1 000 1010인가??

정답은 NO!

2) 음의 정수

1 111 0110

sign 2의 보수

111 0110

000 1001

000 1010

10진수 : -10

1의 보수

+1

2의 보수



부동소수점(floating point number)

: 컴퓨터가 수를 저장하는 방식2

- 실수형

3. 실수를 저장하는 방식

$$\pm 1.m \times 2^{e-b}$$

m : mantissa(가수)

e : exponent(지수)

b : bias(바이어스수)

$$\text{Bias} = 2^{n-1} - 1$$

float의 경우 127

float형의 경우

0 10000110 01010000000000000000000000 0x43280000(16진수)

sign exponent

mantissa

$$1.0101 \times 2^{e-b}$$

$$e = 2^7 + 2^2 + 2^1 = 134$$

$$b = 127$$

$$1.0101 \times 2^7 = 1 * 2^7 + 1 * 2^{7-2} + 2^{7-4} = 168.0$$

부동소수점(floating point number)

: 컴퓨터가 수를 저장하는 방식2

- 실수형

3. 실수를 저장하는 방식

$$\pm 1.m \times 2^{e-b}$$

m : mantissa(가수)

e : exponent(지수)

b : bias(바이어스수)

float형의 경우

$$\text{Bias} = 2^{n-1} - 1$$

float의 경우 127

17.25를 float형으로 나타내보자.

$$e-b = 4$$

$$e-127 = 4$$

$$e = 131$$

1) 2진수로

$$17.25 = 16 + 1 + 0.25 = 2^4 + 2^0 + 2^{-2} = 10001.01$$

2) 정규화 : 가수의 첫 번째 수를 밑보다 작은 자연수로

$$10001.01 = 1.000101 \times 2^4$$

0 10000011 000101000000000000000000

0x418a0000 (16진수)

부동소수점(floating point number)

: 컴퓨터가 수를 저장하는 방식2

- 실수형

4. 2의 지수를 쉽게 알 수 있는 꿀팁

0.511718750은 2진수로??

$$\log_2 0.511718750 = -0.966576998$$

$-0.966576998 > -1$,
which means 2^{-1} 을 포함

$$0.511718750 = 0.5 + 0.011718750$$

$$\log_2 0.011718750 = -6.415037499$$

$-6.415037499 > -7$,
which means 2^{-7} 을 포함

$$0.511718750 = 0.5 + 0.007812500 + 0.003906250$$

$$\log_2 0.003906250 = -8$$

$-8 = -8$,
which means 2^{-8} 이다

0.10000011