컴퓨터 프로그래밍 I
(CSE2003-3)

Mon/Wed 16:30-17:45 pm
Week 1

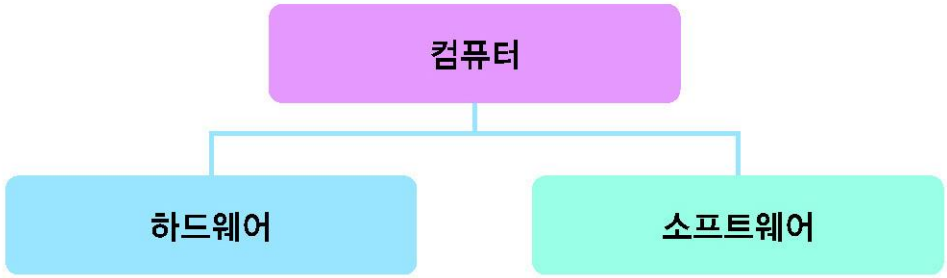
컴퓨터, 데이터, 프로그래밍

컴퓨터와 프로그램

- **컴퓨터**
  - 전자적으로 계산을 수행하는 장치
- **프로그램**
  - 컴퓨터의 행동을 지시하는 명령어

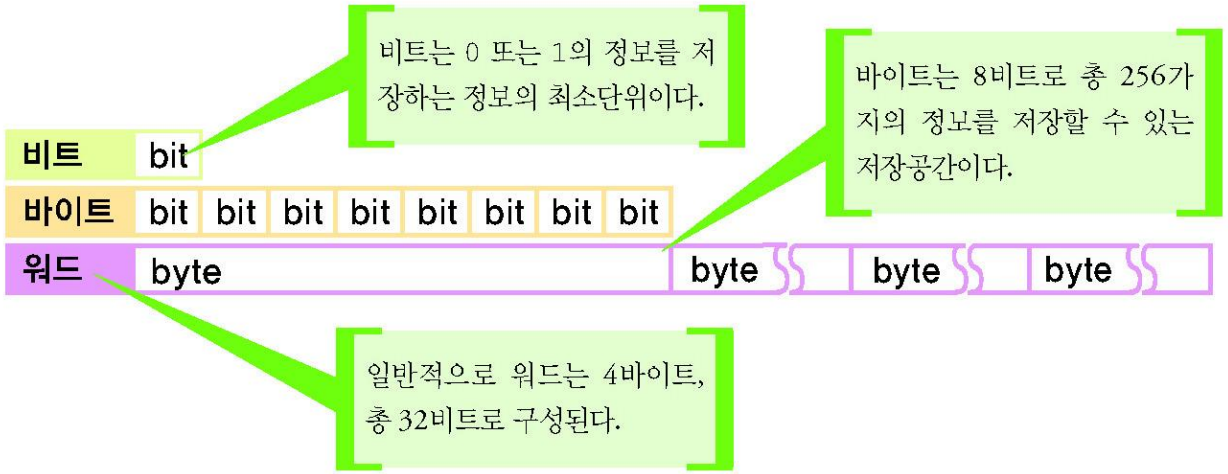
하드웨어와 소프트웨어

- **하드웨어(Hardware)**
  - 컴퓨터를 구성하는 물리적인 장치(device)를 의미
- **소프트웨어(Software)**
  - 컴퓨터가 수행할 작업을 지시하는 명령어들의 집합(프로그램)
  - 소프트웨어는 **응용 소프트웨어**와 **시스템 소프트웨어**로 나누며, 시스템 소프트웨어는 하드웨어를 작동시키는 기본 소프트웨어



컴퓨터의 자료표현

- **자료표현 원리**
  - 2진수 체계를 사용
- **비트와 바이트**
  - Bit
    - Binary Digit, 0 또는 1의 두 개 정보 표현하는 정보의 최소 단위
  - byte
    - 연속된 8개의 비트: 2^8(=256)가지의 서로 다른 정보를 표현할 수 있음
  - 워드
    - 연속된 4 개의 바이트, 총 32비트
    - 실질적으로 컴퓨터 시스템마다 워드의 크기는 다를 수 있음
    - 일반적으로 1워드는 그 시스템에서 표현하는 정수데이터의 기본 크기



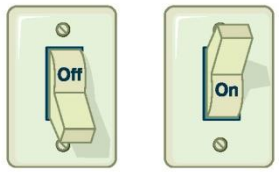


그림 0.4 트랜지스터의 전기적 스위치 0과 1의 2진 표현

비트(Bit), 바이트(Byte), 워드(Word)

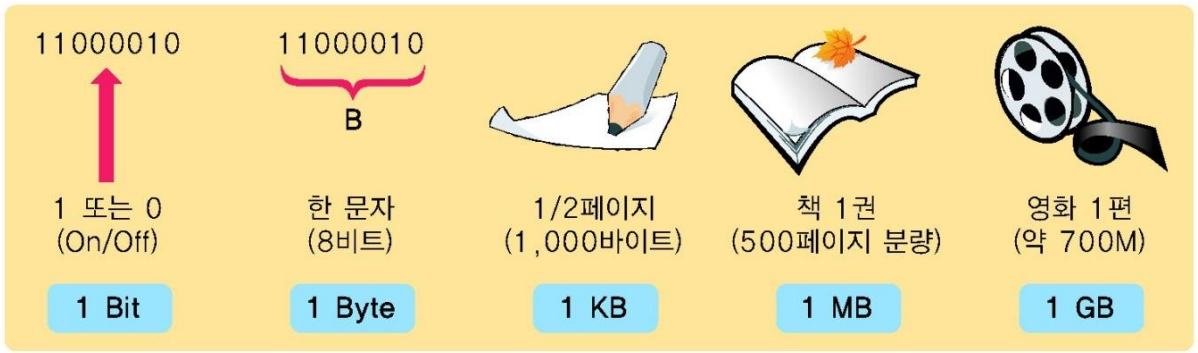
저장단위의 크기

◆ 단위

표기	단위	계산	바이트 수	계량 단위
B	Byte	2^0	1	일
KB	Kilo Byte	2^{10}	1,024	천
MB	Mega Byte	2^{20}	1,048,576	백만
GB	Giga Byte	2^{30}	1,073,741,824	십억
TB	Tera Byte	2^{40}	1,099,511,627,776	조
PB	Peta Byte	2^{50}	1,125,899,906,842,624	천조

정보 저장 용량 단위

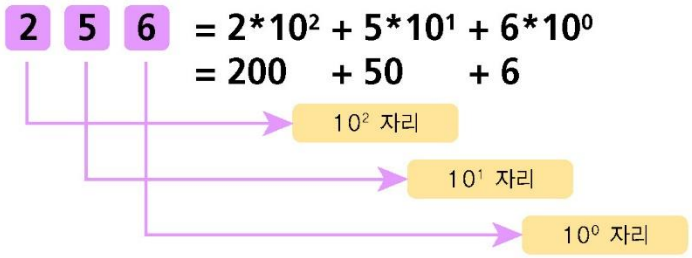
◆ 정보 용량의 비교



정보 저장 용량의 비교

진법과 수의 구성

- 10진법
  - 0에서 9까지의 수를 사용하며, 이 10 가지의 수를 한 자리의 기본 단위로 하는 진법: 0,1,2,3,4,5,6,7,8,9,10,11,....



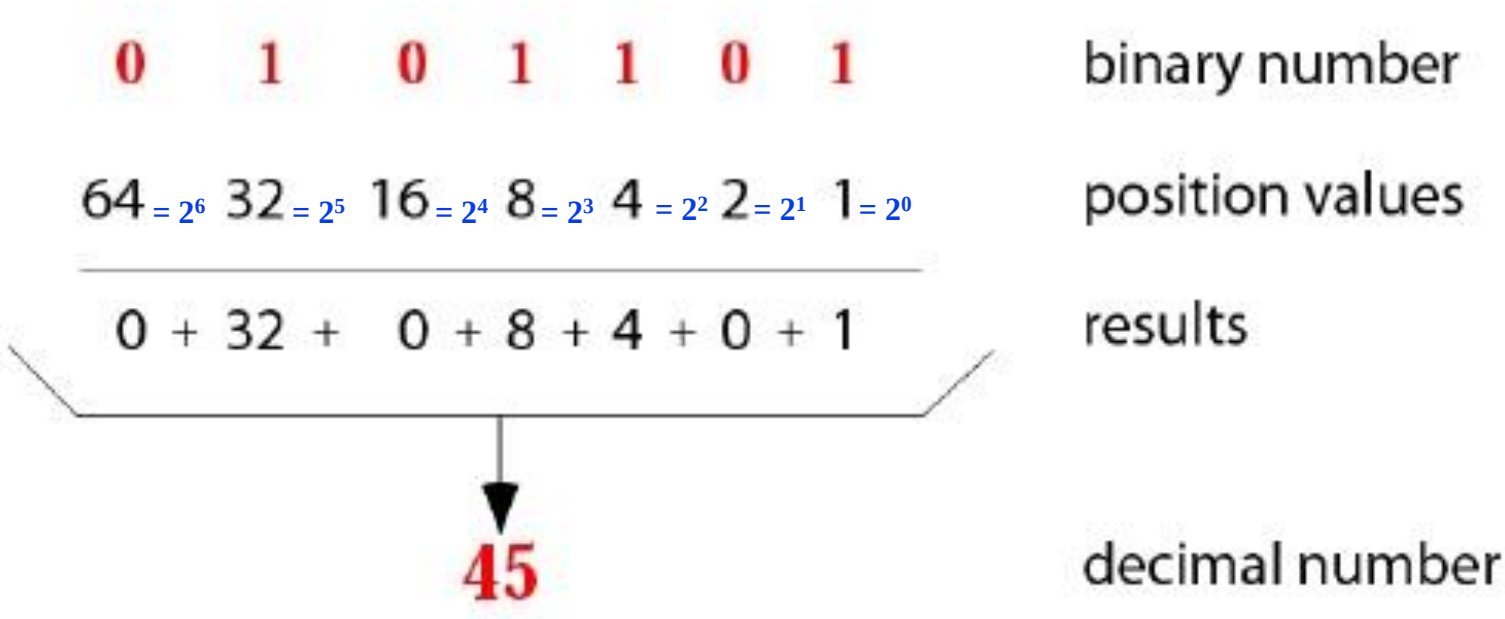
10진수에서 각 자릿수의 의미

- 2진법
  - 0과 1의 조합으로 숫자를 표시하는 방법, 0,1,10,11,100,101,110,111,...
  - 컴퓨터의 자료 표현은 한 비트가 두 가지 표현이 가능
- 16진법
  - 0에서 9, A에서 F까지: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10,11,...

진법과 변환(cont'd)

◆ 2진수 10진 변환

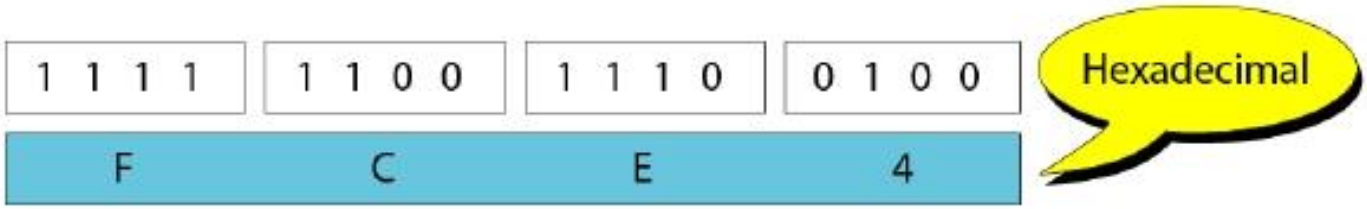
. 2진수에서 각 자리의 숫자에 그 자리의 가중치를 곱한다.



2, 8, 16진수간 상호관계

◆ 2진수와 16진수 사이의 관계

- 4자리 2진수는 0, 1, 2,…,15까지 표현할 수 있다. 따라서 2진수 4자리씩을 16진수로 바꾸면 자연스럽게 16진수로 변환된다.



◆ 16진수 표기법

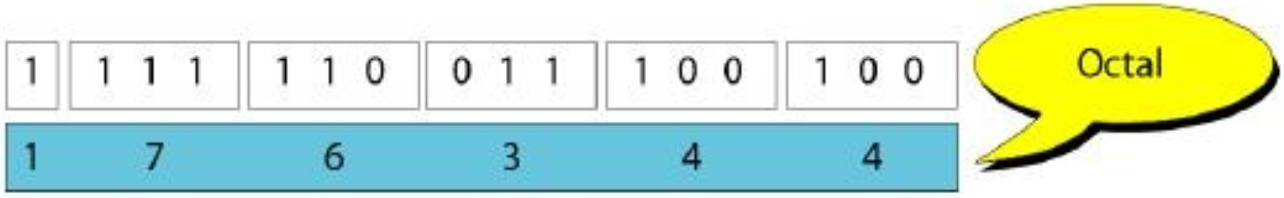
1. 숫자 앞에 소문자(또는 대문자) x를 붙여 준다 **(예: xFCE4)**
2. 숫자(16)를 16진수 뒤에 표기한다.

(예: $FCE4_{16}$ = 1111110011100100_2 = 64740_{10})

2, 8, 16진수간 상호관계

◆ 2진수와 8진수 사이의 관계

- 3자리 2진수는 0, 1, 2,..., 7까지 표현할 수 있다. 따라서 2진수 3자리씩을 8진수로 바꾸면 자연스럽게 8진수로 변환된다.



◆ 8진수 표기법

1. 숫자 앞에 O(영문 Oh)을 붙여 준다 (**예 : O634**)
2. 숫자(8)를 8진수 뒤에 표기한다 (**예: 634_8**)

음수 표현 방법 (2's Complement)

- ✓ **2's complement (2의 보수)**
  - 이진수로 음수를 표현하는 방식 중 가장 많이 사용되는 방식이 2's complement
- ✓ **어떤 숫자의 2's complement를 구하는 방법**
  - 예를 들어 -4를 4비트 이진수로 표현하자면, 먼저 $4_{10} = 0100_2$의 1's complement(1의보수)를 구하고 '1011_2', 거기에 1을 더해준다 '1100_2'. 즉 -4를 '1100_2'로 표현한다.

단계	방법	결과
1단계 (양수를 2진수로)	양수 2를 4비트의 2진수로	0100
2단계 (1의 보수 구하기)	4비트에서 비트를 각각 0은 1로, 1은 0으로	1011
3단계 (1 더하기)	4비트에서 1 더하기	1100

4비트 이진수 -4의 2's complement를 구하는 과정

숫자	1의 보수	2의 보수
+7	0111	0111
+6	0110	0110
+5	0101	0101
+4	0100	0100
+3	0011	0011
+2	0010	0010
+1	0001	0001
0 -0	0000 1111	0000
-1	1110	1111
-2	1101	1110
-3	1100	1101
-4	1011	1100
-5	1010	1011
-6	1001	1010
-7	1000	1001
-8	표기 못함	1000

4비트 2진수의 1's complement와 2's complement

문자데이터의 표현방법

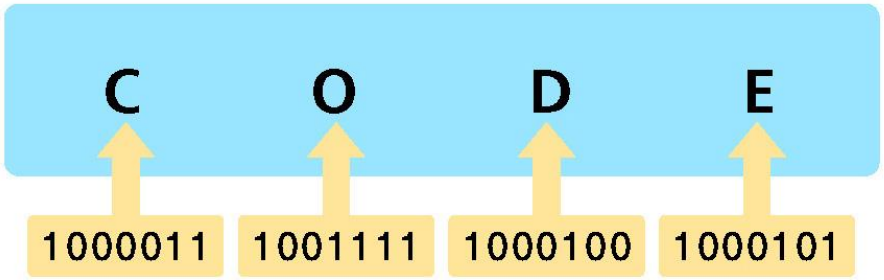
◆ 문자 코드

- N비트의 조합에 일정한 문자를 할당하여 지정한 것을 문자 코드
- 국제 표준인 문자 코드는 ASCII코드, EBCDIC코드(IBM코드), Unicode(유니코드)

◆ 아스키(ASCII) 코드: 1963년 미국표준협회 지정 문자 코드

- ASCII(American Standard Code for Information Interchange)
- 국제적인 표준으로 사용하는 문자 코드 체계로서 7비트를 사용하여 128(2^7)개의 문자, 숫자, 특수문자 코드를 규정
- 대문자 A의 코드는 1000001 B는 1000010 C는 1000011이며, 소문자 a의 코드는 1100001 b는 1100010 c는 1100011과 같이 순서대로 지정했다

문자의 비트 표현은 **7-bit** 이진수로 볼 수도 있다. 즉 같은 값의 데이터라도 문자로 읽는 것과 숫자로 읽는 것에 따라 데이터의 형태가 달라진다



문자의 비트 표현

아스키코드표

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
010	Space	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
110	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

1 2

ASCII 코드표 (행으로 먼저 읽고 열의 값을 붙인다):

(예) 1000001 = A = 65_{10} , 1000010 = B = 66_{10} , … , 1011010 = Z = 90_{10} 즉, A부터 Z까지 1씩 증가.
1100001 = a = 97_{10} , 1100010 = b = 98_{10} , … , 1111010 = z = 122_{10} 즉, a부터 z까지도 1씩 증가.
'a' - 'A' = 32, 'b' - 'B' = 32, … , 'z' - 'Z' = 32
즉, 소문자 = 대문자 + 32

문자의 표현방법: 한글, 한자, 일본어...

- **아스키(ASCII) 코드는 1 Byte (= 8 bits)로 표현**
  - 7bits만 사용하고 첫번째 bit는 항상 0 값을 갖는다
- **한글코드는?**
  - 1987년 국가 표준으로 KSC5601이라는 **2 Bytes** 완성형코드가 제정
    - 한글 2,350자, 한자 4,888자
  - 1989년 한글 1930자, 고어 1673자 추가
  - 1992년 2 Bytes 조합형 제정 (완성형과 공동 표준으로 사용)
- **Unicode는 전세계의 모든 언어들을 하나의 표준으로 표기할 수 있도록 시도**
  - 1992년 ISO10646 코드 제정
  - 아직도 각 나라마다 자기들의 코드체계를 사용하고 있는 경우가 많은데, 최근 들어서 점차 확산되고 있는 추세.
  - 기본형 2Bytes(16bits), 확장형 4Bytes(32bits)

Programming Language

- **프로그래밍 언어의 필요성**
  - 사람과 컴퓨터가 서로 의사교환을 하기 위해 컴퓨터가 이해할 수 있는 언어가 필요.
    - 인간의 언어는 자연어(Natural Language)
    - 프로그래밍 언어는 형식언어(Formal Language)
  - 사람이 컴퓨터에게 지시할 명령어를 프로그래밍 언어로 기술

기계어와 어셈블리어

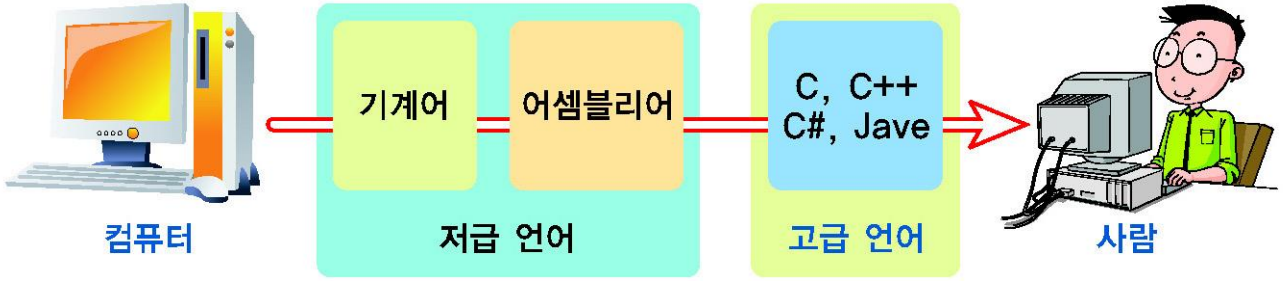
- **기계어(Machine language)**
  - 0과 1로 표현되는 프로그래밍 언어로서 컴퓨터가 직접 이해할 수 있는 유일한 언어
  - CPU 명령 회로에 입력으로 사용되는 것으로 이해하면 됨
- **어셈블리어(Assembly language)**
  - 기계어의 명령코드와 피연산자(operand)를 프로그래머가 좀 더 이해하기 쉬운 기호(symbol) 형태로 일대일 대응시킨 기계어 수준의 프로그래밍

순서	기계어	어셈블리어	의미
명령어1	0101 000000000100	LDA A	메모리 A의 내용을 누산 레지스터(AC)에 저장
명령어2	0111 000000000110	ADD B	메모리 B의 내용과 누산 레지스터(AC)의 값을 더하여 누산 레지스터(AC)에 다시 저장
명령어3	0100 000000000111	STA C	누산 레지스터(AC)의 값을 메모리 C에 저장
명령어4	0011 000000000000	HLT	프로그램 종료

두 정수의 합을 위한 기계어 프로그램의 예: C = A + B

High level languages vs. Low level languages

- **High level language**
  - 인간의 언어를 닮은 고급언어: C, JAVA, Python, FORTRAN 등
- **Low level language**
  - 컴퓨터가 이해할 수 있는 machine language(기계어)와 그것을 문자화한 assembly language (어셈블리 언어)



컴퓨터에 가까운 기계어, 어셈블리어와 인간의 언어에 가까운 고급언어

고급 언어의 종류

- **FORTRAN**
  - 포트란(FORTRAN)은 FORmula TRANslating system(수식 번역 시스템)의 약자
  - 과학과 공학 및 수학적 문제들을 해결하기 위해 고안된 프로그래밍 언어
- **COBOL**
  - 코볼(COmmon Business Oriented Language)은 기업의 사무처리에 적합한 프로그래밍 언어
- **Basic**
  - 베이직(BASIC)은 Beginner's All-purpose Symbolic Instruction Code의 약어
  - 초보자도 쉽게 배울 수 있도록 만들어진 대화형 프로그래밍 언어
- **PASCAL**
  - PASCAL은 프랑스의 수학자인 파스칼(Pascal)의 이름에서 따온 언어
  - 프로그램의 작성 즉, 구조적 프로그래밍 및 알고리즘 교육에 적합하도록 개발된 프로그래밍 언어

C와 C++

- **C**
  - C는 유닉스(UNIX)의 운영체제 작성을 위한 시스템 프로그래밍 언어로 설계된 언어
  - 컴파일러나 수많은 소프트웨어 도구(Tool)들도 C언어로 개발
  - 다른 고급언어에 비하여 하드웨어에 대한 보다 확실한 통제가 가능
  - 특정 컴퓨터 기종에 의존하지 않으므로 프로그램의 이식성(portability)이 높음
  - 풍부한 연산자와 데이터 형(data type)을 갖고 있기 때문에 범용 프로그래밍 언어로서 널리 보급되었으며, 응용 소프트웨어의 개발에 널리 이용
- **C++**
  - C++은 객체지향 프로그래밍(OOP: Object-Oriented Programming)을 지원하기 위해 C언어가 가지는 장점을 그대로 계승하면서 객체의 상속성(inheritance) 등의 개념을 추가한 효과적인 언어
  - C++는 C언어의 확장이라고 볼 수 있으므로 기존의 C언어로 개발된 모든 프로그램들을 수정 없이 그대로 사용 가능

JAVA

JAVA

- 자바(JAVA)의 시초는 1992년 미국의 SUN 사에서 가전 제품들을 제어하기 위한 언어에서부터 비롯됨
- 운영체제나 CPU와는 독립적으로 실행 가능한 프로그래밍 언어
- 자바는 C++언어의 기초 위에 객체지향 개념을 구현하도록 설계된 언어
- 분산 네트워크상에서의 프로그래밍이 용이
- 자바 프로그램의 실행은 운영체제의 가상 머신(Virtual Machine) 위에서 인터프리터 방식으로

현대 프로그래머에게 중요한 언어 C, C++, Java, Python

- Python은 배우기 매우 쉽고, 특히 최근 Tensorflow나 PyTorch와 같은 인공지능의 Deep Learning 시스템에서 사용되고 있어서 반드시 배워야 할 언어 중 하나로 최근 가장 빠르게 확산되고 있는 프로그래밍 언어이다.

Program vs. Software

- **프로그램**
  - 프로그램은 컴퓨터에게 어떤 일의 수행을 지시하는 명령어(instruction) 집합
  - 소스와 실행파일
    - 특정한 프로그램 언어로 이 명령어의 집합을 모아 놓은 파일을 프로그램 소스(source)
    - 이 소스 파일로부터 만들어진 실행 파일은 컴퓨터가 이해할 수 있는 기계어로 명령어를 모아 놓은 파일
- **소프트웨어**
  - 소프트웨어는 보통 '프로그램'이라고 부르는 것 외에도 데이터와 문서까지를 포함하는 포괄적인 개념

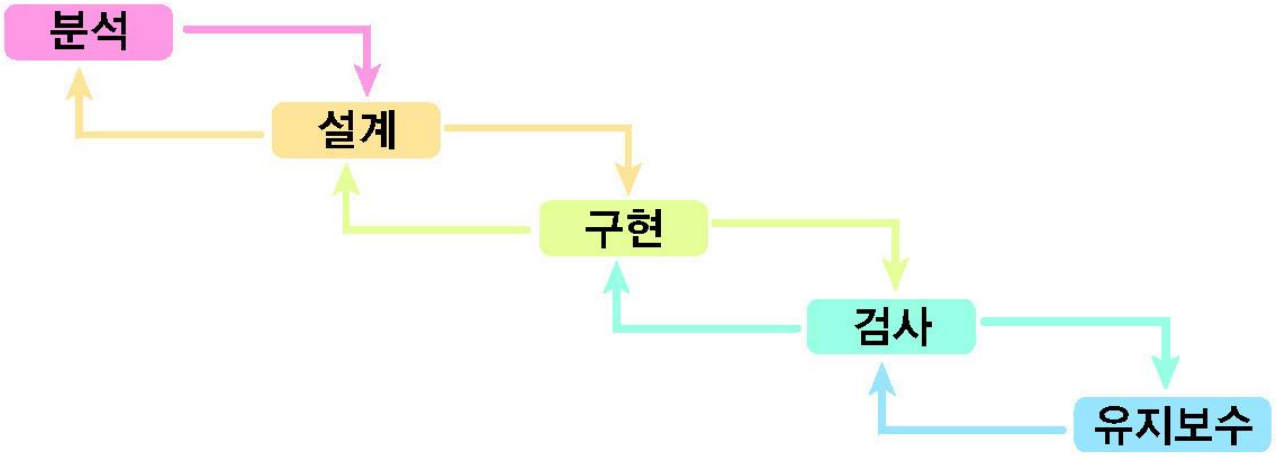
소프트웨어 공학과 개발단계

◆ 소프트웨어 공학

- 신뢰성 있고 실제 기계에서 효과적으로 작동하는 소프트웨어를 경제적으로 얻기 위해서 올바른 공학적 원리들을 체계화시킨 학문

◆ 소프트웨어의 생명주기

- 소프트웨어 개발단계는 분석, 설계, 구현, 검사, 유지보수의 과정을 거치며, 개발과정에서 이 단계를 순환적으로 반복



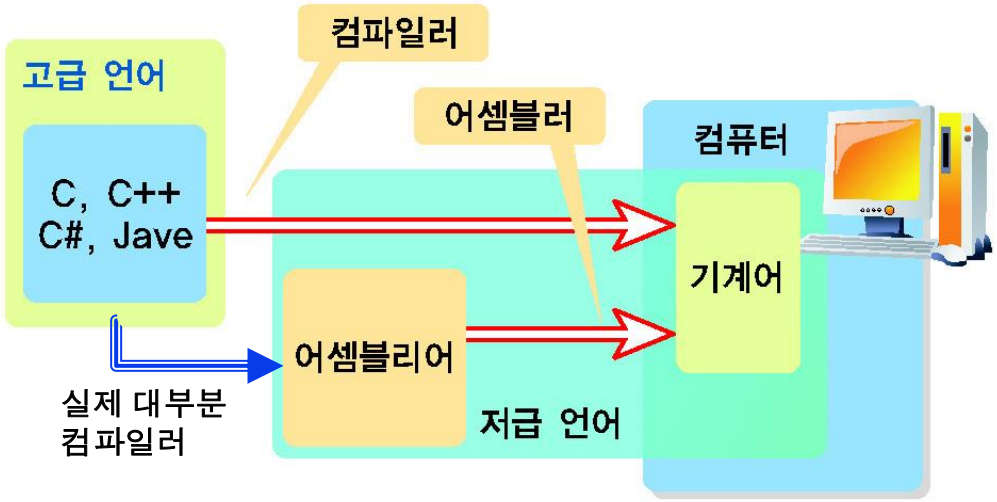
프로그램 개발 환경

통합 개발 환경

- 생산성 향상을 위한 Integrated Development Environments
- 프로그램을 개발하는데 필요한 컴파일러, 디버거, 링커, 에디터 등을 통합적으로 제공하는 개발 환경
- 예:
  - MS Visual Studio (MS Windows 용)
  - 리눅스 환경에서는 여러가지 통합 도구들이 공개되어 있음 vi에 다양한 도구를 확장한 것과 emacs라고 하는 텍스트 에디터를 확장한 통합도구, 그 외에 Eclipse, CodeBlocks 등 다양한 오픈소스 통합 개발 환경 도구들이 널리 쓰이고 있음
  - 컴퓨터공학과 학생들은 양쪽 도구를 다 자유롭게 활용해 보는 것을 권장

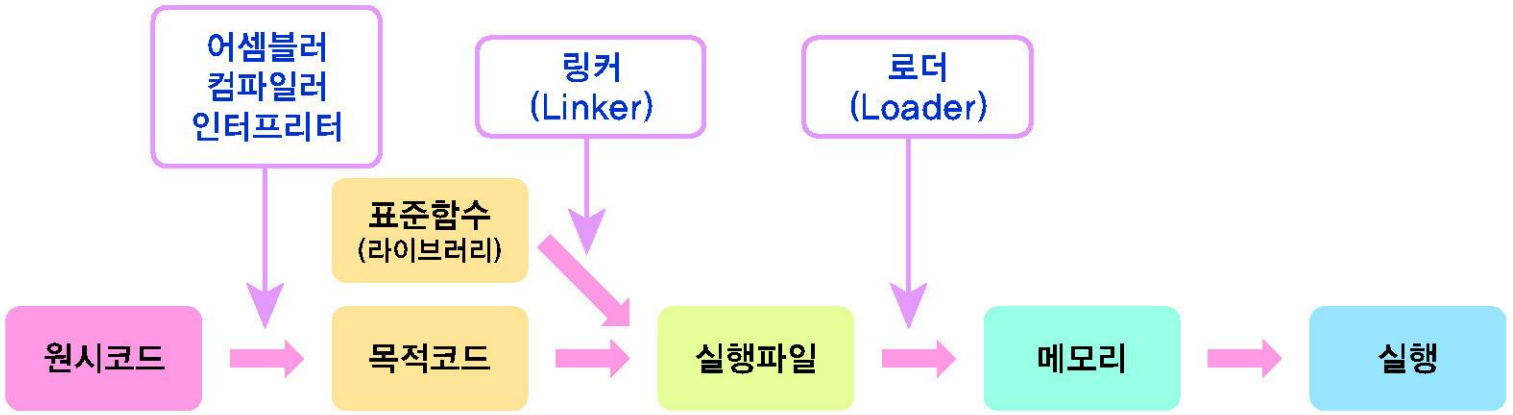
컴파일러와 어셈블러

- **컴파일러(compiler)**
  - 고급언어로 작성된 프로그램을 기계어로 바꾸어주는 프로그램
- **어셈블러(assembler)**
  - 어셈블리 언어로 작성된 프로그램을 기계어로 바꾸어주는 프로그램



프로그램 개발 과정

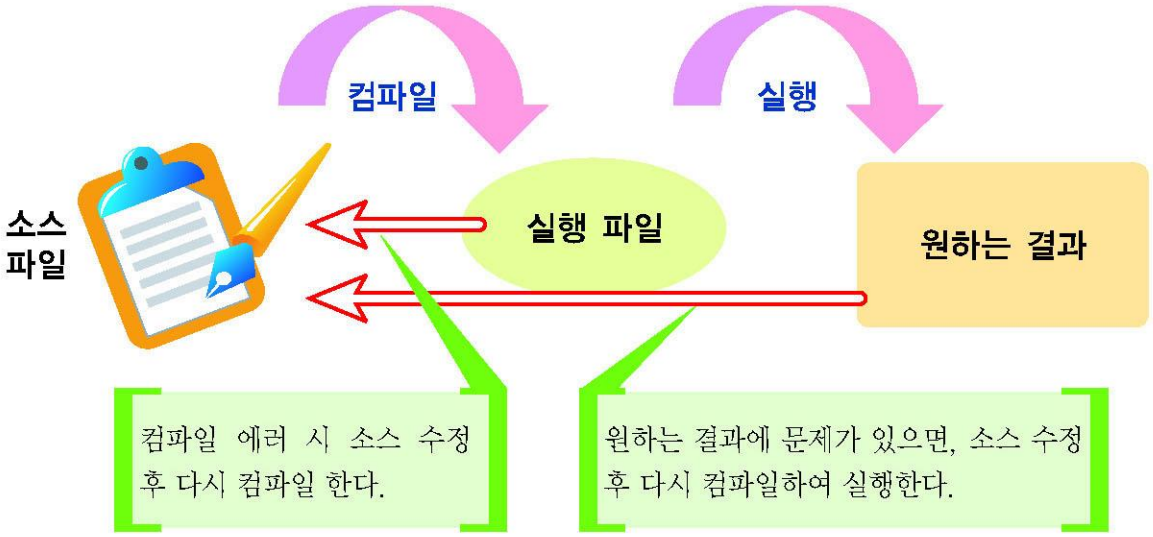
- **링커(linker)**
  - 이러한 여러 개의 목적 파일들을 라이브러리 함수와 연결해서 하나의 파일인 실행파일을 생성하는 작업을 수행
- **로더(loader)**
  - 작성된 프로그램을 컴퓨터의 주기억장치에 로드(load)함으로써 프로그램을 실행 가능하게 하는 역할을 수행



원시코드(source program)을 실행파일로 바꾸어서 실제 실행되는 과정

오류 수정

- **디버깅(debugging)**
  - 프로그램 개발과정에서 컴파일 에러나 실행 에러를 수정하는 과정



정리

- **Bit, byte**
- **2진수, 8진수, 16진수**
- **음수 표현방법**
  - 2’s complement
- **문자 표현 방법**
- **컴퓨터와 프로그래밍 언어: 어셈블러, 컴파일러…**
- **프로그램 vs. 소프트웨어**
- **소프트웨어 공학**