

11주 1강

# 어셈블리 프로그램의 이해

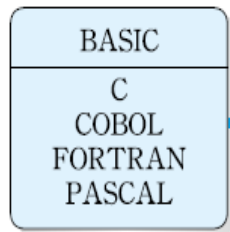


- 컴퓨터 프로그래밍 언어의 표현하는 방식에 따른 분류
  - 고급 프로그래밍 언어, 저급 프로그래밍 언어

## ★ 고급 프로그래밍 언어(high level programming language)

- 인간의 언어 체계와 유사한 문법으로 프로그램 작성이 용이
- 컴파일러(compiler)나 인터프리터(interpreter)에 의해 기계어로 번역되어 실행
- 대표적인 고급언어로는 FORTRAN, PASCAL, COBOL, C 언어 등이 있다.
- 컴파일과 인터프리터의 과정

고급 프로그래밍 언어



인터프리터

컴파일러

기계어 프로그램



00001111 10101010 11111111

# 저급 프로그래밍 언어(low level programming language)



- 컴퓨터에서 바로 처리 가능한 프로그래밍 언어로
- 일반적으로 기계어와 어셈블리어를 일컫는다.

## ★ 기계어(machine language)

- 컴퓨터가 직접 해독할 수 있는 2진 숫자(binary digit)로 표현된 언어.
- 컴퓨터 명령 형식은 기계어로, 이를 해독해서 컴퓨터는 동작을 수행
- 컴퓨터에 따라 고유의 명령 형식이 존재, 기계어 구조도 컴퓨터에 따라 구성된다.
- 기계어의 명령 단위는 명령 코드부와 주소부로 나뉘어진다.
- 기계어는 프로그램 작성이 어렵고, 많은 노력과 시간이 필요하다.

## ★ 변수 para에 데이터 3을 저장하는 고급 언어의 표현 예

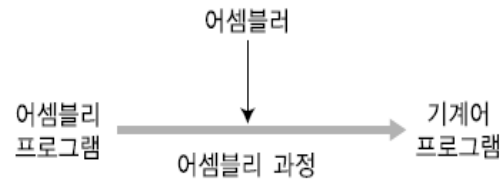
para = 3

- 컴파일러를 통해서 기계어로 변환하면 다음의 비트 형태로 표시될 수 있다.
- 11000111 00000110 00000000 00000000 00000011 00000000

# 어셈블리(Assembly) 언어



- 비효율적인 기계어의 비트 형식을 연상 코드로 나타낸 것
  - 기계어를 사람이 사용하는 언어에 가깝게 문자로 기호화해서 나타낸다.
  - 어셈블리어로 작성된 프로그램들도 중앙처리장치에 따라 다르다.



- 어셈블리어를 번역하여 오브젝트 코드를 생성하는 프로그램을 어셈블러(Assembler)라 한다.

- 기계어를 어셈블리어로 나타내는 예

`MOV para, 3`

- 3이라는 데이터를 para라는 기억 장소로 이동시키라는 의미

- 어셈블리 프로그램이 어셈블러(Assembler)에 의하여 기계어로 변환되는 과정

- 어셈블리 과정은 컴파일 과정보다 빨리 수행, 어셈블리어가 기계어와 유사하기 때문

# 어셈블리 관련 명령어



명령어	동작
ADD	덧셈
SUB	뺄셈
MUL	곱셈
DIV	나눗셈
MOV	데이터 이동
LOAD	기억장치로부터 데이터 적재
STOR	기억장치로 데이터 저장

# 8086 어셈블리 언어의 명령 형식



## ① 레이블(Label)부

- JUMP, LOOP 같은 순환이나 반복명령어에서 해당 레이블로 프로그램 카운터를 이동시킬 때 사용. 일반적인 명령어에서는 생략된다.
- 8문자 이내의 영문자/숫자를 사용하며, 이름 중에 공백이 있으면 안 된다.

DOSTART:	ADD	X	;X와 가산기를 더하고 그 결과를 가산기에 저장
레이블 부	연산 부	오퍼랜드 부	주석문 부

## ② 연산(Operation)부

- 연산부는 명령어나 지시어가 위치하는 곳이다.

## ③ 오퍼랜드(Operand)부 또는 피 연산자부

- 레지스터 이름, 정수, 연산자, 주소 등을 쓰는 곳이다.

## ④ 주석문(Command)부

- 해당 프로그램 줄(line)에 대한 설명을 표기하는 곳이다.
- 세미콜론(;)으로 시작하며, 어셈블러는 주석문부를 모두 무시하기 때문에 아무데나 사용하여도 된다.



## ● 피 연산자는 모두 기억장치의 주소

- 기억장치 250번지에서 데이터를 누산기에 적재
- 기억장치 251번지의 데이터와 덧셈을 수행하고 결과를 다시 누산기에 저장
- 기억장치 251번지에 결과를 저장한다.
- 프로그램의 주소 170번지로 점프한다.

주소	명령어	기계 코드
100	LOAD 250	1250
101	ADD 251	5251
102	STOR 251	2251
103	JUMP 170	8170

## ● 주기억장치, CPU 레지스터, ALU의 동작 과정

- 명령어 LOAD 250은 기억장치 100번지에서 인출되어 명령어 레지스터(IR)에 저장
- 명령어가 해독되어, 기억장치 250 번지의 데이터가 누산기(AC)로 이동
- 프로그램 카운터(PC)가 다음 명령어를 수행하기 위해서 하나 증가

$$PC = PC + 1 = 101$$

다음 시간

## 11주 2강. 명령어 사이클

