



02

강

프로그램의 번역과 프로그래밍 작성 절차

학습목표

▶ 프로그램의 번역 과정과 프로그래밍의 작성 절차를 이해할 수 있다.

1 프로그램의 번역

(1) 언어 번역 프로그램

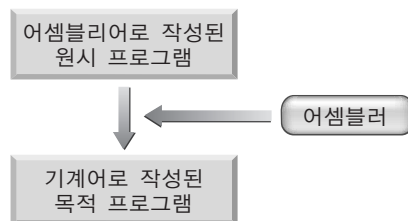
- ① 프로그램은 사람이 작성하지만, 컴퓨터가 실행하므로 컴퓨터가 이해하고 실행할 수 있는 언어로 작성되어야 한다. 따라서 기계어가 아닌 다른 언어로 작성한 프로그램은 컴퓨터가 처리할 수 있는 기계어로 번역해야 실행할 수 있다. 이처럼 기계어로 번역하는 프로그램을 언어 번역 프로그램이라고 한다.
- ② 언어 번역 프로그램은 언어의 종류와 번역 방식에 따라 어셈블러, 컴파일러, 인터프리터 등이 있다.
- ③ 기계어는 컴퓨터가 바로 이해하고 실행할 수 있는 프로그래밍 언어이므로 언어 번역 프로그램이 필요하지 않다.

(2) 번역과 프로그램

- ① 프로그래머가 작성한 프로그램을 원시 프로그램(source program, source code)이라 하며 컴퓨터로 실행하기 위해서는 기계어로 번역할 필요가 있다.
- ② 원시 프로그램을 기계어로 번역한 프로그램을 목적 프로그램(object program)이라 하고, 이것을 직접 실행 가능한 형태의 실행 프로그램으로 만든 후 (링크) 컴퓨터에서 실행한다.

2 어셈블러(Assembler)

- ① 어셈블리 언어로 작성된 원시 프로그램을 기계어로 번역하는 프로그램이다.
- ② 어셈블러는 기계어와 표현 형식만 다르다고 할 수 있는 어셈블리 언어를 기계어로 바꾸어서 표현하는 형식으로 번역하므로 번역 절차가 비교적 간단하다.
- ③ 어셈블러는 기계어와 1:1로 대응되는 표현 형식만 다른 기계어 코드를 만든다.



▲ 어셈블러를 이용한 번역 과정

3 컴파일러(Compiler)

(1) 컴파일이란

- ① 고급 언어로 작성된 프로그램(=원시 프로그램, source program)을 처리하여 컴퓨터가 사용할 수 있는 기계어 코드로 작성된 프로그램(=목적 프로그램, object program)으로 변경시켜 주는 과정을 컴파일(compile)이라 한다. 이때 사용하는 언어 번역 프로그램을 컴파일러(compiler)라고 한다. 어셈블러와는 다르게 기계



EBS tip

● 원시 프로그램

프로그래머가 프로그래밍 언어를 사용하여 작성한 프로그램

● 목적 프로그램

컴퓨터가 이해할 수 있는 기계어로 번역된 프로그램

● 어셈블러(Assembler)

- 어셈블리어로 작성된 프로그램은 어셈블러에 의하여 목적 프로그램으로 만들어지며, 로더에 의하여 실행 가능한 형태로 주 기억 장치에 적재되어 실행한다.
- 어셈블러는 기계어와 1 : 1로 대응되는 표현 형식만 다른 기계어 코드를 만든다.

● 컴파일러(Compiler)

기계어로 컴파일을 해 주는 번역 프로그램이다.

어와 1대 1로 대응하는 명령 코드를 생성하는 것이 아니고 동일한 작업을 수행할 수 있도록 기계어로 작성된 명령어 코드들을 만들어낸다.

- ② 컴파일러는 실행 시에 모든 문장을 먼저 구문적으로 하나씩 분해하고, (다른 문장을 참조하는 경우) 문장이 정확하게 참조될 수 있도록 여러 번의 연속적인 상태에서 결과 코드를 만든다. 이 결과 코드가 바로 목적 프로그램이고, 이 목적 프로그램은 CPU가 한 번에 한 명령씩 처리하거나 실행할 수 있는 기계어 프로그램이다.
- ③ 각 고급 언어마다 고유의 컴파일러가 있다. (포트란 컴파일러, 코볼 컴파일러, 파스칼 컴파일러, C 컴파일러, C++ 컴파일러, 비주얼베이직 컴파일러 등)
- ④ 한 번 실행 파일이 만들어지면 실행하기 위하여 다시 번역할 필요가 없다.
- ⑤ 한 문장이라도 수정하였을 경우에는 프로그램 전체를 다시 컴파일해야 한다.
- ⑥ 컴파일러 방식의 고급 언어에는 C, 코볼, 포트란, 파스칼, C++, 자바 등이 있다.

(2) 컴파일러의 분류

목적 코드가 실행되는 방법에 따라 네이티브 컴파일러, 크로스 컴파일러, 바이트 코드 컴파일러로 분류된다

- ① 네이티브 컴파일러(native compiler) : 컴파일러가 실행되는 컴퓨터나 운영 체제가 컴파일러의 목적 코드가 실행될 컴퓨터나 운영 체제일 경우에 네이티브 컴파일러라 한다.
- ② 크로스 컴파일러 : 다른 컴퓨터나 운영 체제에서 실행되도록 제작되는 경우를 크로스 컴파일러(cross compiler)라 한다. 이는 임베디드 시스템 등 소프트웨어 개발에 충분한 환경을 갖추지 못한 환경에서 동작할 프로그램을 만들기 위해 사용된다. 예를 들어, 휴대 전화에서 실행되는 프로그램을 개발하여 컴파일할 때는 PC에서 컴파일하여 휴대 전화에 이식한다. 이때 PC에서 사용하는 컴파일러가 크로스 컴파일러이다.
- ③ 바이트 코드 컴파일러 : 어떤 컴퓨터나 운영 체제에서 직접 실행 가능한 프로그램의 형태로 컴파일하는 것이 아니라 가상 머신(virtual machine)에서 동작하는 프로그램으로 컴파일하는 컴파일러이다. 컴파일 후 출력물이 가상 머신을 위해 제작된 바이트코드 형태의 기계어로 된다. 이들 가상 머신에는 저스트 인 타임 컴파일러가 탑재되어 실행 직전에 현재 플랫폼의 기계어로 다시 한 번 컴파일(저스트 인 타임 컴파일)되어 실행된다. 여기에는 스몰토크, 자바, 마이크로소프트 공통 중간 언어(CIL) 등이 있다.

4 링커(Linker)

전통적으로 보통의 운영 체제에서는 컴파일 이후에 또 하나의 추가 과정이 요구되는 데, 하나 이상의 목적 모듈이 동시에 실행되고 그것들이 서로의 명령어나 데이터를 상호 참조하게 될 때, 그 명령어나 데이터 간의 상대적 위치를 정하는 과정을 말한다. 이 과정을 링크 편집(linkage editing)이라 하며, 그 결과물은 로드 모듈(load module)이라 한다.

5 로더(Loader)

로더는 목적 프로그램을 읽어들여 주기억 장치에 적재시킨 후에 실행하는 프로그램으로 목적 프로그램의 재배치, 주기억 장치 내에 프로그램을 적재시킬 공간의 확보, 여러 개의 목적 프로그램이나 라이브러리 루틴과의 링크 작업, 또는 실제 프로그램을 보조 기억 장치에서 읽어 들이는 기능을 한다.



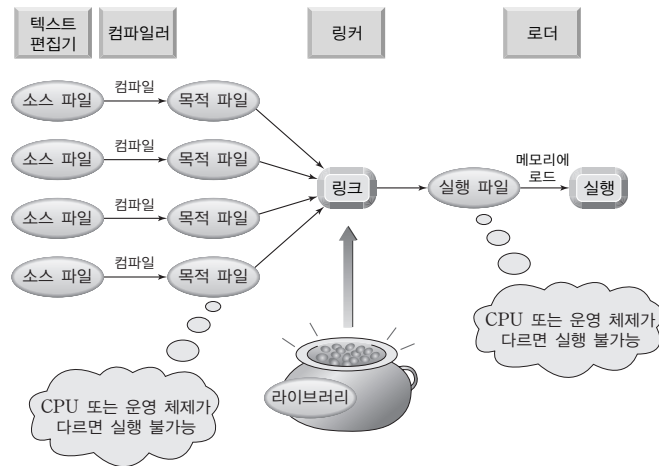
● 프리프로세서

- 전처리기와 함.
- 프로그램을 컴파일하기 전에 타당성 검사, 기호 변환, 매크로 확장, 주석 제거 등의 작업을 수행하는 프로그램

● 링커

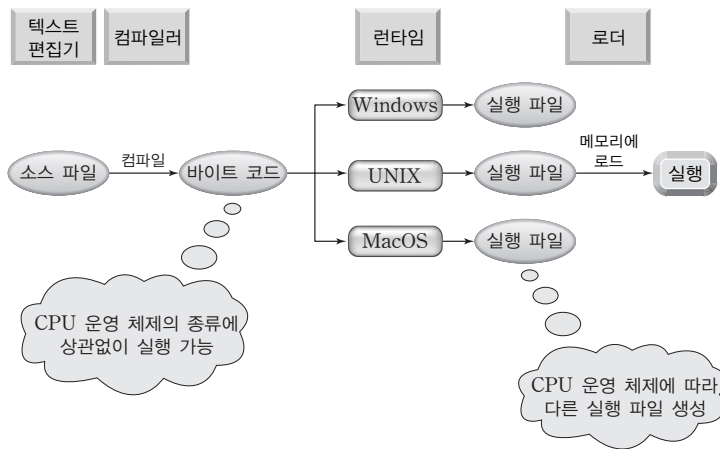
실행 가능한 프로그램을 만들기 위해 컴파일된 목적 파일을 서로 연결시키는 기능을 하는 프로그램이다.

① 프로그램 개발과 실행 과정(OS, CPU에 종속)



▲ C 언어를 사용한 프로그램의 개발 과정과 실행

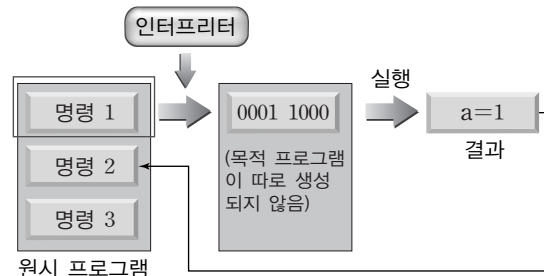
② 프로그램 개발과 실행 과정(OS, CPU에 독립)



▲ 자바를 사용한 프로그램의 개발 과정과 실행

6 인터프리터(Interpreter)

① 고급 언어로 작성된 원시 프로그램을 명령문 단위로 하나씩 번역한 후, 바로 실행하는 언어 번역 프로그램이다.



▲ 인터프리터를 이용한 번역 과정

■ 인터프리터 방식의 고급 언어에는 베이직, 리스프(LISP), 프롤로그, 웹 스크립트 언어(PHP, ASP) 등이 있다.

- ② 실행할 때마다 원시 프로그램을 번역하여 처리하므로 많은 시간이 걸린다.
 ③ 프로그램의 작성 과정에서 오류 검색, 추가, 삭제가 쉽다.



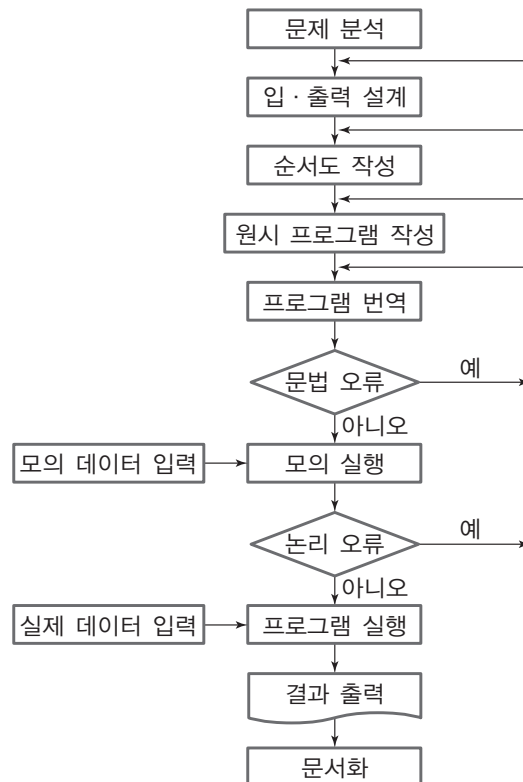
7 컴파일러와 인터프리터 방식의 비교

특징 \ 방식	컴파일러 방식	인터프리터 방식
번역 방법	프로그램 전체 번역함.	실행되는 행 단위로 번역함.
장점	한번 번역하여 생성된 실행 파일은 다시 번역 과정 없이 실행할 수 있어 실행 속도가 빠름.	오류 수정이 쉽고, 번역 과정이 비교적 간단함.
단점	프로그램의 일부를 수정하는 경우에도 전체를 컴파일해야 함.	실행할 때마다 번역해야 함.
결과물	목적 프로그램과 실행 프로그램 생성	목적 프로그램과 실행 프로그램 없이 곧 바로 실행하여 결과물을 산출함.
사용 언어	포트란, 코볼, 파스칼, C, C++, Java	베이직, 리스프, 프롤로그, ASP, PHP

● 종합 개발 툴(IDE)

특정 언어를 사용하여 프로그램을 개발하기 위한 툴로, 소스 코드를 기술하기 위한 툴이나 컴파일러, 디버거 등의 툴이 하나의 패키지에 모두 포함되어 있다.

8 프로그래밍 작성 절차



▲ 프로그래밍 절차



① 문제 분석 단계

- 프로그램에서 해결해야 할 업무나 문제가 무엇인지 규명하는 단계이다.
- 문제를 해결하기 위한 여러 가지 방법을 비교·분석하여 보다 나은 방법을 결정한다.
- 입력 데이터와 출력 정보 및 프로그래밍에 소모되는 비용, 기간 등을 조사, 분석하여 타당성을 검토한다.

② 입출력 설계 단계

- 문제 해결을 위해 입력되는 데이터의 입력 형태와 장치, 처리 방법, 출력 형태와 장치 등을 결정한다.
- 입력 설계 : 입력 데이터의 형식과 분량, 데이터 항목과 순서, 크기와 성격, 입력 매체 등을 결정한다.
- 출력 설계 : 출력 매체, 출력 내용의 형식과 방법 등을 결정한다.

③ 순서도 작성 단계

- 문제를 처리하려는 방법과 순서를 단계적으로 일정한 기호를 사용하여 논리적으로 도표화한 것이다.
- 설계 도구에는 순서도, 나시-슈나이더만(NS, Nassi-Schneiderman) 차트, 의사 코드(pseudo code) 등이 있다.
- 순서도는 프로그램을 코딩하거나 검증 및 수정할 때 사용한다.

④ 원시 프로그램 작성 단계

- 입·출력 설계와 순서도에 따라 프로그래밍 언어를 선택하여 처리할 내용에 따라 명령문을 기술하는 과정이다.
- 코딩 과정에서 프로그래밍 언어의 문법과 규칙에 맞지 않거나 틀린 글자가 있으면 프로그램 번역과 실행 과정에서 오류가 발생하거나 올바른 결과를 얻지 못하므로 주의가 필요하다.

⑤ 번역과 오류 수정 단계

작성된 원시 프로그램을 언어 번역 프로그램을 사용하여 컴퓨터가 이해할 수 있는 목적 프로그램으로 바꾸는 과정과, 이 과정에서 발생하는 오류를 수정하는 단계이다.

⑥ 모의 실행 단계

번역된 목적 프로그램에 모든 문장이 한 번 이상 실행되도록 모의 데이터를 입력하여 실제 데이터 처리 과정에서 생길 수 있는 논리적 오류를 찾아 수정하는 단계이다.

⑦ 프로그램 실행 단계

실제 처리할 데이터를 입력하여 처리하고 원하는 결과를 얻는 과정이다.

⑧ 문서화 단계

프로그램의 실행이 성공적으로 끝나면 프로그램에 관계되는 모든 자료를 문서화하여 보관한다.

● 문법적 오류

프로그래밍 언어의 문법을 잘못 사용하여 프로그램의 번역 과정에서 발생하는 오류이다. 이 오류는 번역 프로그램이 번역하는 과정에서 검사하여 오류 메시지를 출력하여 줌으로 비교적 쉽게 수정할 수 있다.

● 논리적 오류

문법 상의 오류는 없었다고 하더라도, 계산 결과가 나오지 않거나 틀린 결과가 발생하는 오류를 말하며, 이때에는 문제 분석 단계에서부터 원인을 찾아내어 원시 프로그램을 수정해야 한다.



기출 모의고사

정답 및 해설 p. 2

1

2006학년도 대수능

프로그래밍 언어의 특징을 표로 작성하였다. 각 내용(ㄱ~ㄴ)의 참 또는 거짓을 다음의 논리 게이트에 입력하였을 때 결과값(Z)이 참인 것은? (단, 내용이 옳으면 참이고 틀리면 거짓이다.)

	구분	입력 조건	
		컴파일러 언어(X)	인터프리터 언어(Y)
ㄱ	분류	고급 언어	저급 언어
ㄴ	디버깅	불가능	가능
ㄷ	번역 단위	일괄 번역	한 문장씩 번역
ㄹ	해당 언어	FORTTRAN	C
ㅁ	목적 프로그램	생성 안함.	생성함.



- ① ㄱ ② ㄴ ③ ㄷ
④ ㄹ ⑤ ㅁ

2

2009학년도 대수능

다음은 프로그래밍 언어 번역기에 대하여 설명한 것이다. (가)~(다)에 들어갈 내용으로 옳은 것은?

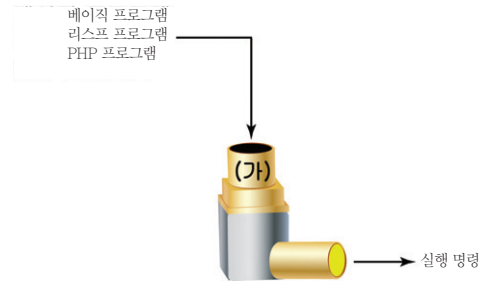
프로그래밍 언어 번역기의 종류

- (가) 저급 언어로 작성된 프로그램을 번역할 때 사용한다.
기계어와 1:1로 대응되는 기호화 언어를 번역한다.
- (나) 고급 언어로 작성된 프로그램을 번역할 때 사용한다.
원시 프로그램에 대한 목적 프로그램을 생성한다.
- (다) 고급 언어로 작성된 프로그램을 번역할 때 사용한다.
원시 프로그램을 명령문 단위로 하나씩 번역하여 바로 실행한다.

- (가) (나) (다)
- ① 어셈블러 컴파일러 인터프리터
② 어셈블러 인터프리터 컴파일러
③ 컴파일러 어셈블러 인터프리터
④ 컴파일러 인터프리터 어셈블러
⑤ 인터프리터 어셈블러 컴파일러

3

그림 (가)의 언어 번역 프로그램에 관련된 것을 <보기>에서 고른 것은?



보기

- ㄱ. 컴파일러를 언어 번역 프로그램으로 사용한다.
ㄴ. 한 문장씩 독립적으로 실행하므로 문장 단위로 실행 결과를 알아볼 수 있다.
ㄷ. 목적 프로그램을 생성하지 않는다.
ㄹ. 논리적 오류가 발생하지 않는다.

- ① ㄱ, ㄴ ② ㄱ, ㄷ ③ ㄴ, ㄷ
④ ㄴ, ㄹ ⑤ ㄷ, ㄹ

4

다음은 프로그램 작성과 번역 과정에서 사용하는 용어에 대한 설명이다. 설명과 용어가 서로 바르게 짝지어진 것은?

- (가) 원시 프로그램을 목적 프로그램으로 변환한다.
(나) 문법적으로는 오류가 없으나 계산 결과가 나오지 않거나 틀린 결과가 발생하는 오류
(다) 여러 개의 목적 프로그램을 연결하여 하나의 실행 프로그램으로 만드는 과정

- (가) (나) (다)
- ① 컴파일 문법적 오류 링킹
② 컴파일 논리적 오류 디버깅
③ 컴파일 논리적 오류 링킹
④ 로딩 문법적 오류 컴파일
⑤ 로딩 문법적 오류 링킹

5 다음은 세 사람이 공동으로 해결해야 할 과제와 역할 분담에 대한 내용이다. 프로그래밍 작성 절차 중 영탁이의 역할로 바른 것은?

〈과제〉 몇 년도의 몇 번째 일을 입력으로 받아 그 해의 월, 일로 출력하는 프로그램을 작성하시오.

〈역할 분담〉

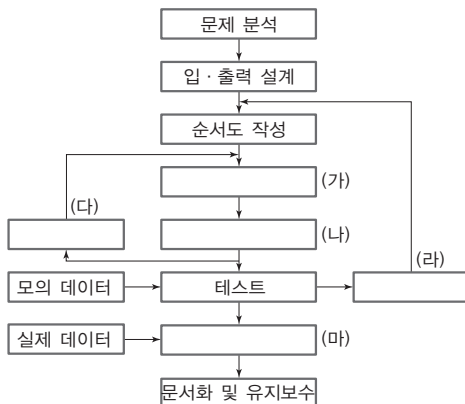
진옥 : 컴퓨터로 처리 가능한 것인지를 확인한다. 문제 해결을 위한 방법을 그림으로 표현한다.

영탁 : 진옥이가 제시한 문제 해결 방법을 이용하여 C 언어로 프로그램을 작성하고, 목적 프로그램으로 바꾸는 과정을 수행한다.

수진 : 영탁이가 작성한 프로그램에 논리적인 오류가 있는지 검사한다. 실제 처리할 데이터를 입력하여 실행해 보고, 모든 자료를 문서화한다.

- ① 문제 분석과 입출력 설계
- ② 순서도 작성과 프로그램의 코딩
- ③ 프로그램 코딩과 번역
- ④ 프로그램 번역과 오류수정 및 모의 실행
- ⑤ 입출력 설계와 프로그램 실행

6 다음은 프로그래밍 순서를 단계적으로 나타낸 그림이다. (가)~(마)에 대한 설명으로 옳은 것을 <보기>에서 고른 것은?



보기

- ㄱ. (가)에서 C 언어로 작성된 프로그램은 바로 실행할 수 있다.
- ㄴ. 포트란 언어로 작성된 프로그램은 (나)에서 컴파일러를 사용하여 목적 프로그램으로 변환된다.
- ㄷ. 명령어를 잘못 사용하여 발생하는 문법적 오류는 (다)에서 수정할 수 있다.
- ㄹ. 논리적 오류는 (라)에서 수정할 수 없다.
- ㅁ. (마)에서 실행하는 프로그램에서 문법적 오류가 발생할 수 있다.

- ① ㄱ, ㄴ ② ㄴ, ㄷ ③ ㄷ, ㄹ
- ④ ㄷ, ㅁ ⑤ ㄹ, ㅁ

7 다음은 게임 프로그램 개발 절차에 대하여 협의한 내용이다. 발표 학생을 프로그램 작성 순서대로 바르게 나열한 것은?

영희 : 우리가 개발하려는 프로그램에 대한 정확한 분석을 해야 해.

영탁 : 나는 비주얼베이직으로 코딩할게.

정희 : 자, 그럼 이 게임에 대한 순서도를 작성해보자.

수진 : 사용할 입력장치를 정하자.

혜진 : 우리 우선 시범 게임을 해볼까?

- ① 영희-영탁-정희-혜진-수진
- ② 정희-영탁-영희-수진-혜진
- ③ 수진-영희-정희-영탁-혜진
- ④ 정희-혜진-수진-영희-영탁
- ⑤ 영희-수진-정희-영탁-혜진