

Hanbit eBook

Realtime 58

BACK TO THE BASIC

뿌리부터 이해하는

C 언어

Vol I

이승미, 이희정 지음

 한빛미디어
Hanbit Media, Inc.

BACK TO THE BASIC

뿌리부터 이해하는 C 언어
Vol.1

BACK TO THE BASIC 뿌리부터 이해하는 C 언어 Vol. I

초판발행 2014년 4월 30일

지은이 이승미, 이희정 / 펴낸이 김태현

펴낸곳 한빛미디어(주) / 주소 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부

전화 02-325-5544 / 팩스 02-336-7124

등록 1999년 6월 24일 제10-1779호

ISBN 978-89-6848-690-6 15000 / 정가 12,000원

책임편집 배용석 / 기획 김병희 / 편집 안선화

디자인 표지 여동일, 내지 스튜디오 [맘], 조판 김현미

영업 김형진, 김진불, 조유미 / 마케팅 박상용, 서은옥, 김옥현

이 책에 대한 의견이나 오타자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주십시오.

한빛미디어 홈페이지 www.hanbit.co.kr / 이메일 ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea

Copyright © 2014 이승미, 이희정 & HANBIT Media, Inc.

이 책의 저작권은 이승미, 이희정과 한빛미디어(주)에 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다.

책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요.

한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

저자 소개

저자 이승미

서울여자대학교 컴퓨터공학과/정보보호공학과를 졸업했다. 모바일 관련 벤처기업에서 8년간 근무하며 C와 C++ 기반의 브라우저 개발과 국내 및 해외 모바일 솔루션 상용화에 참여했다. 모바일 환경이 안드로이드로 옮겨가는 시기에 안드로이드 환경에 적합한 오피스 관련 제품군 상용화에 참여하였으며, 현재는 작은 모바일 앱 개발 업체인 **소프트씨드**의 대표이자 개발자로 근무 중이다.

• seungmil@gmail.com | <http://androbook.tistory.com/>

저자 이희정

국민대학교 컴퓨터응용학과 졸업 후 한양대학교 컴퓨터공학과에서 석사과정을 마쳤다. 모바일 관련 벤처기업에서 7년간 근무하며 C와 C++ 기반의 브라우저 개발을 비롯해 국내뿐만 아니라 동남아시아 여러 국가의 모바일 솔루션 상용화에도 참여했다. 현재는 C와 C++, 웹 기술들에 관심이 있으며 다양한 분야의 컴퓨터 서적을 번역하는 등 프리랜서로 활동 중이다.

• heejoung80@gmail.com | <http://ichbbol.blogspot.com/>

스페셜 코드 에디터_ 김태희

현재 캐나다 몬트리올에 있는 Warner Bros Games Montreal에서 서버 팀의 팀장으로서 크고 작은 온라인 게임을 제작하고 있다. 이 책에 있는 예제 리뷰에 참여하였다.

• steeple@naver.com | <http://blog.eodl.com/>

저자 서문

같은 회사를 7년 정도 다니고 비슷한 즈음 그 회사를 그만둔 인연으로 어찌어찌 함께 C 언어에 관한 책을 써 보기로 했을 때, 과연 우리가 책을 쓸만한 그릇이 되나 하고 잠시 고민했었다. 생초짜 개발자 시절부터 무시무시한 프로젝트에 눌러 꾸역 꾸역 버텨내긴 했지만, 남에게 알려줄 만한 지식을 쌓아놓긴 했는지 스스로 의심스러웠기 때문에. 하지만 둘 다 뭔가 결정하는 데 오래 고민하는 스타일은 못 돼서, “일단 해보자”고 결정을 내린 후 그해 겨울 줄곧 커피숍에서 만나 의논해가며 나를 즐겁게 원고를 진행했다. 책을 써 내려가며 우스갯소리로 “우리가 C를 하긴 했구나”라는 말을 했다. 뭣도 모르고 개발을 시작해서 여전히 아는 것은 하나도 없는 것 같은데, 막상 누군가에게 C를 알려줘야겠다고 생각하니 전해 줄 말이 너무 많았다.

원고를 완성하고 사실 우리의 원고가 묻힐지도 모르겠다는 생각을 했다. 다행히(?) 나를 열심히 만든 원고가 책이 되어 나온다고 하니 심장이 덜컥거린다. “제대로 책을 썼나?”라는 걱정이 앞선다. 이 책이 개발자로서의 첫걸음을 내딛는 많은 이들에게 조금이나마 도움이 되길 바라는 마음이다.

Thanks to

승미

첫 입사 이후 줄곧 무시무시한 프로젝트를 통해 ‘스파르타’식으로 저를 이끌어주신 구용구 수석님(가끔 저를 비웃다가 눈물 흘리신 건 잊어드릴게요.), 8년여 세월 동안 같은 회사에서 개발자로서 서로 보탬이 되었던 동료들(은경, 미주, 정훈, 광균……, 너무 많아 생략), 무심한 파트장(?) 만나서 고생한 우리 애기들(선욱, 범삼, 상훈, 기현, 관구, 애기라곤 할 수 없지만, 함께 힘들게 일했던 재성, 계형, 동진)을 비롯해서 함께 일했던 모든 분께 감사드립니다. 생애 첫 집필에 함께 기뻐해 준 우리 가족들, 친구들 감사드립니다. 언제나 내 삶에 기쁨에 되어주는 우리 조카들(예원, 예나) 이모가 무지 사랑해. 책을 출판할 기회를 만들어 주신 엄진영 님, 출판에 힘써 주신 김병희 님 및 한빛미디어의 많은 분께 감사드립니다. 함께 원고를 집필하느라 고생한 희정, 정말 수고했어.

희정

집필하는 동안 많은 지원을 해주신 가족들께 감사드립니다. 저에게 끝없는 용기를 불어넣어 준 때이님 감사합니다. 출판 업계에 뛰어들게 해주신 김병부 님께 감사드립니다. 그리고 이 책을 출판하기까지 정말 많이 도와주신 한빛미디어의 김병희 님과 스마트미디어팀 여러분께 감사드립니다. 함께 책 쓰고 먹으러 다니느라 고생한 승미 언니, 소프트씨드 성공하길 빌어요.

대상 독자 및 예제 파일

초급

초중급

중급

중고급

고급

이 책은 '프로그래밍을 해보자'에 초점이 맞춰져 있다.

C 프로그래밍을 시작하기 위해 알아야 할 것은 그리 많지 않다. 필자는 기본적으로 컴퓨터를 열어서 프로그램을 설치하고, 키보드를 사용해 글자를 타이핑할 수 있는 사람이라면 모두 프로그래밍을 할 수 있다고 본다. C 프로그래밍을 할 수 있는 프로그램을 자신의 컴퓨터에 설치하고, 그 프로그램을 실행해서 몇 줄의 코드를 작성한 뒤, 컴파일을 시작하는 버튼을 클릭한다. 그리고 실행 버튼을 클릭하여 방금 작성한 프로그램을 눈으로 확인한다. 이 과정 자체가 프로그래밍이며, 위에 단 몇 줄로 적은 바와 같이 이것은 그다지 어려운 과정이 아니다.

이러한 과정을 따라해 보고 코드라는 것을 작성하기 위해 필요한 기초적인 것들을 배우면서 몇 줄의 코드를 더 작성해본다. 프로그래밍을 시작하기 위해 모든 이론을 완벽히 알고 있을 필요는 없다. 어려운 부분이 있다면 일단 넘어가자. 그 부분을 알아야만 프로그래밍을 할 수 있는 것은 아니다. 일단 프로그래밍이 어떤 것인지 경험해보길 권한다.

영어와 한국어 등 세상에 서로 다른 언어가 많듯이, 컴퓨터가 알아듣는 언어에도 여러 가지가 있다. C, C++, Java 등이 컴퓨터가 알아들을 수 있는 언어이며, 이 책에서는 이 중 C라는 언어를 사용해서 컴퓨터와 소통하는 방법을 익혀갈 것이다.

영어를 처음 배울 때 “I’m a girl.”을 배우고 “I’m a beautiful girl.”을 배우듯, 초반에 일단 프로그래밍이란 것을 경험해보고, 각 단락 단락에서 기본적인 코드에 추가적으로 붙여나갈 수 있는 것들에 무엇이 있는지를 확인할 것이다. 영어선생님은 영어를 하는 방법을 가르쳐주는 사람이지만, 외국인을 만나 나눌 대본을 만들어주는 사람이 아니다. 또 언어를 배운 후에는 언어를 사용해봐야 잊어버리지 않는다. 마찬가지로 이 책을 통해 C언어를 사용하는 방법을 익히 뒤, 독자 여러분도 한 번쯤 꼭 그 언어를 사용해 각자 만들어보고 싶었던 프로그램을 작성해보고 바란다.

■ 예제 파일 및 책에서 사용한 기호의 의미

이 책에 포함된 소스 코드는 다음 주소를 통해 다운받을 수 있다.

- <http://www.hanbit.co.kr/exam/2690>

이 책에 사용한 기호는 다음과 같은 의미로 사용하였다.



표시(error) : 컴파일 시 에러가 발생하는 경우



표시(warning) : 컴파일 시 에러가 발생하진 않지만, 바람직하지 않은 경우

한빛 eBook 리얼타임

한빛 eBook 리얼타임은 IT 개발자를 위한 eBook입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾는 것도 쉽지 않습니다. 또한 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1. eBook Only - 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다.

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 수 있는 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수 분에게는 보다 쉽게 집필할 수 있는 기회가 될 수 있으리라 생각합니다. 또한 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

2. 무료로 업데이트되는, 전자책 전용 서비스입니다.

종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나 저자(역자)와 독자가 소통하면서 보완하여 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3. 독자의 편의를 위하여 DRM-Free로 제공합니다.

구매한 전자책을 다양한 IT 기기에서 자유롭게 활용할 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

4. 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다.

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 힘든 전자책이 아니라, 전자책의 환경에 가능한 한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가도록 하겠습니다.

지금 보시는 전자책에 소유권한을 표시한 문구가 없거나 타인의 소유권한을 표시한 문구가 있다면 위법하게 사용하고 있을 가능성이 높습니다. 이 경우 저작권법에 의해 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한 한빛미디어 사이트에서 구입하신 후에는 횡수에 관계없이 다운받으실 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오타자 교정이나 내용의 수정·보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려드리며, 구매하신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 한빛미디어 사이트에서 구입하신 독자에게만 제공되며, 다른 사람에게 양도나 이전은 허락되지 않습니다.

차례

01	C 프로그래밍 시작하기	1
	1.1 컴퓨터, 그리고 프로그램	2
	1.2 컴퓨터의 언어_고급 언어와 C 언어	7
	1.3 컴퓨터의 기억력_프로그램은 누가 가지고 있나?	14
	1.4 프로그래밍 환경 구성하기	18
02	변수	43
	2.1 변수 선언하기	44
	2.2 변수 데이터 타입	64
	2.3 사용자 입력받기와 출력하기	79
03	조건문	94
	3.1 if - 만약에	95
	3.2 숫자 비교하기, 문자 비교하기	102
	3.3 여러 개의 조건 사용하기	110
	3.4 변수의 유효성이란?	127
04	순환문	132
	4.1 순환문이란?	133
	4.2 순환문, 어떻게 써야 할까?	137
	4.3 순환문, 영원히 돌 순 없다	150
	4.4 순환중첩 사용하기	157

5.1 배열, Array와 친해지기	169
5.2 배열 뜯어보기	172
5.3 다차원 배열 만들기	178
5.4 배열, 정말로 많이 쓰이나?	190

1 | C 프로그래밍 시작하기

1장의 목표

- 컴퓨터의 기본 구조와 동작을 이해한다.
- 프로그램의 의미와 만들어지는 과정을 이해한다.
- C 언어의 역할을 이해한다.
- 개발에 필요한 환경을 구성하고 간단한 프로그램을 만들어본다.

1장의 구성

- 1.1. 컴퓨터, 그리고 프로그램
 - 1.2. 컴퓨터의 언어_고급 언어와 C 언어
 - 1.3. 컴퓨터의 기억력_ 프로그램은 누가 가지고 있나?
 - 1.4. 프로그래밍 환경 구성하기
-

요리를 하려면 요리의 재료와 도구가 필요하고, 미적분을 이해하기 위해서는 종이와 연필을 들고 기초적인 숫자부터(1,2,3...) 먼저 배워야만 한다. 마찬가지로 프로그래밍을 하려면 컴퓨터와 컴파일러가 필요하며, 이것들을 어떻게 동작시켜 프로그래밍해야 하는지에 대해 한 단계씩 익혀나가야 한다. 영화 속 천재적인 주인공들은 순식간에 멋진 프로그램을 만들어 컴퓨터를 조종하지만, 그들 역시 처음에는 컴파일러가 무엇이며 컴퓨터는 어떻게 생긴 장치인지를 먼저 배웠을 것이다. 1장에서는 컴퓨터란 무엇이고 프로그래밍을 하려면 무엇이 필요한지 확인한 뒤, 그 어떤 천재 프로그래머라도 처음에는 거쳐 갈 수밖에 없는 유명한 프로그램, "HelloWorld!"를 만들어볼 것이다.

1.1 컴퓨터, 그리고 프로그램

요즘은 컴퓨터가 뭘지 모르는 사람을 찾아보기 힘들다. 웬만한 집에는 컴퓨터가 한 대쯤 있으며 대형마트의 계산대라든지 도서관 입구에서도 쉽게 찾아볼 수 있다. 요즘 인기 있는 스마트폰이나 태블릿 역시 컴퓨터의 한 형태다. 컴퓨터는 말 그대로 생활 곳곳에 스며들어 우리들의 일상을 돕고 있다. 그런데 정확히 무엇을 컴퓨터라고 부르는 것일까?

그림 1-1 컴퓨터의 한 형태인 스마트폰



1.1.1 컴퓨터의 구조

컴퓨터의 의미를 사전에서 검색해보면, ‘정보를 입력하여 그것을 정해진 과정대로 처리하고 그 결과를 제공해주는 기기’라고 정의되어 있다. 우리가 흔하게 보는 컴퓨터의 구조를 간단히 도식화하면 ‘입력 장치’, ‘본체’, ‘출력 장치’로 구분할 수 있다. 키보드, 마우스 등 사용자에게 어떤 동작을 입력받아 정보를 전달하는 장치들이 입력 장치다. 그리고 컴퓨터 내부 본체에서 처리된 정보를 사용자에게 다시 보여주는 모니터, 프린터, 스피커 등의 장치는 출력 장치에 포함된다.

최근에는 물리적인 키보드를 대신하여 모니터(화면)에 띄운 가상 자판기를 활용하거나 마우스 대신 터치 센서를 이용하기도 한다. 요즘의 스마트폰을 생각해 보면 이러한 추세가 이해될 것이다. 입력과 출력을 대표하는 키보드나 모니터 등은 눈으로 확인되는 장치이므로 큰 설명이 필요 없겠지만, 본체는 이와 달리 케이스에 가려져

일반 사용자는 그 내부를 잘 모르고 있는 경우가 많다. 컴퓨터 본체의 내부는 어떻게 구성되어 있을까?

그림 1-2 컴퓨터의 구성



입력되는 정보와 출력되는 정보는 모두 본체를 거쳐 흐른다. 정보가 흐른다는 것은 이 정보를 전달해주는 길이 있다는 의미다. 어릴 때 한 번쯤 만들어보았던 종이컵 전화기를 생각해보자. 한쪽에서 종이컵에 대고 말하면 이 음성이 상대편 종이컵으로 전달된다. 음성은 종이컵에서 종이컵으로 꺾충 뛰어가는 것이 아니며, 종이컵 사이에 연결된 실을 통해 흘러 전달된다. 이렇게 정보를 전달하기 위해 만들어진 길이 컴퓨터 내부의 회로판이다.

그림 1-3 컴퓨터 내부의 회로판은 정보 전달 통로다.

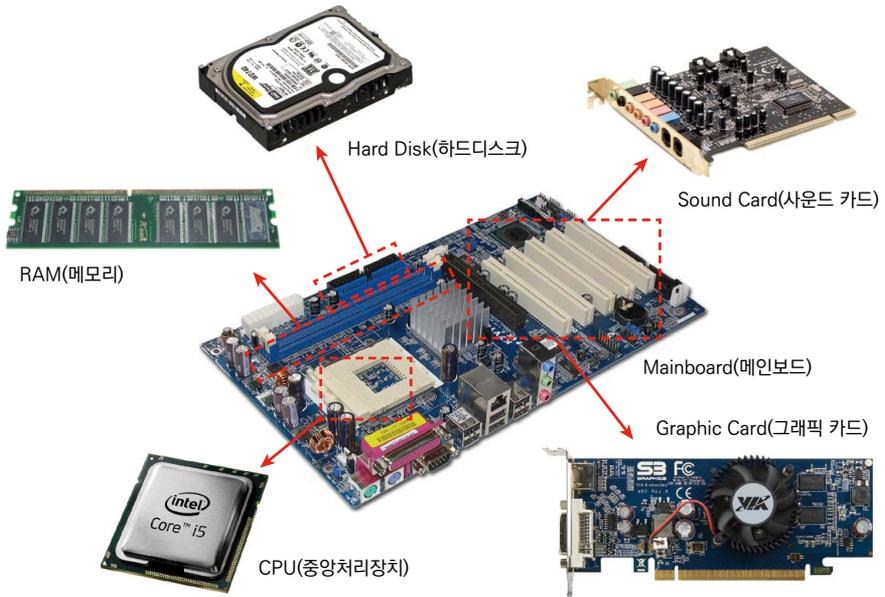


즉, 컴퓨터 본체의 회로판이 입력된 정보를 처리하고 다시 출력 장치로 그 정보를 흘려보내 준다. 이 회로판을 가리켜 ‘메인보드’라고 부르며, 메인보드에는 다양한 정보를 처리하기 위한 칩^{chip}들이 곳곳에 꽂혀 있다. 메인보드에 결합하여 정보를 처리하는 칩이나 부품에는 대표적으로 다음과 같은 것들이 있다.

- CPU(중앙처리장치): CPU는 컴퓨터의 머리와 같다. 컴퓨터에 전원이 들어오면 CPU는 내장된 프로그램을 통해 컴퓨터 전반의 정보 흐름을 처리·관리한다.
- RAM(메모리): 입력되는 정보를 처리하기 위해 정보들을 임시로 저장할 수 있는 메모리다. 말 그대로 ‘임시’ 저장소기 때문에 한 번에 저장할 수 있는 크기는 크지 않다. 최근에는 1GB부터 4GB 크기의 RAM이 일반적으로 사용되고 있는데, 사용자들에 따라서는 8GB 등으로 업그레이드하여 사용하기도 한다.
- Hard Disk(하드디스크): 컴퓨터의 전원이 나간 후에도 정보를 저장해 둘 수 있는 지속성 있는 메모리(저장소)다. 최근에는 동영상이나 게임 등의 크기가 커지면서 사용자가 원하는 하드디스크의 크기도 점점 커져서, TB(테라바이트) 단위의 하드디스크도 쉽게 찾아볼 수 있게 되었다.

- Lan Card(랜카드): 인터넷 정보를 처리하는 부품이다. 랜선 등 인터넷과 연결되는 선이 이 카드와 연결된다. 최근에는 무선 랜카드도 출시되어, 무선으로 연결된 인터넷 정보를 처리할 수 있다.
- Graphic Card(그래픽 카드): 화면 출력에 관련된 정보를 처리한다. 내부 정보를 처리하여 모니터로 화면을 전달한다.
- Sound Card(사운드 카드): 소리 출력에 관련된 정보를 처리한다. 내부 정보를 처리하여 스피커, 혹은 이어폰으로 소리를 전달한다.

그림 1-4 컴퓨터의 구성



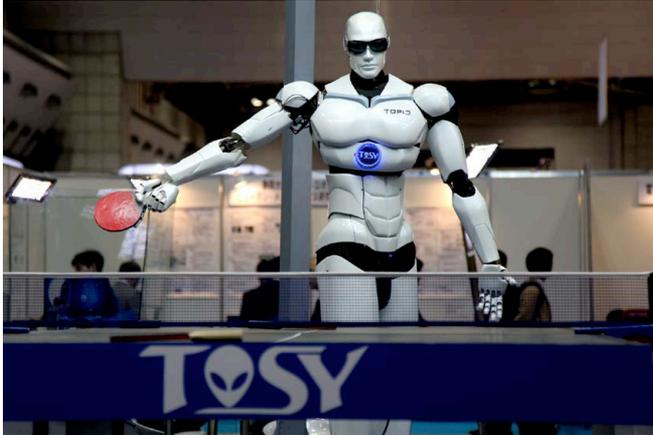
모든 상품이 그렇듯, 컴퓨터 부품들 역시 계속해서 더 좋아지고 더 빨라진 버전으로 업그레이드되어 시장에 등장하고 있다. 흔히 컴퓨터 사양을 나타낼 때에는 부품들에 관련한 정보(제조사, 버전, 사양)를 표시하며, 사용자는 필요에 따라 최근에

나온 더 좋은 것으로 부품을 업그레이드한다.

1.1.2 프로그램이란?

컴퓨터의 기계적인 구조에 대해 대략 둘러보았으니, 이 기계가 어떻게 동작하는지 확인해볼 차례다. 컴퓨터의 동작은 만든 사람이 결정한다. 스스로 해야 할 일을 생각하고 몸을 움직여 일을 하나씩 수행하는 사람과 달리, 컴퓨터의 머리인 CPU는 사람이 디자인하여 명령한 동작만 수행할 수 있다. 스스로 생각하고 움직이는 안드로이드에 대한 연구가 진행 중이긴 하지만, 그 궁극적인 목적을 보면 결국 사람이 명령한 일을 좀 더 편리하게 수행하는 것이다.

그림 1-5 사람이 명령한 일을 좀 더 편리하게 수행하기 위해 만든 안드로이드



컴퓨터는 ‘전원이 들어오는 순간 특정 저장소에 저장된 프로그램을 읽어, 프로그램에 적혀 있는 동작을 수행하도록 만들어진 기계 장치’라고 정의할 수도 있다. 이때 프로그램이란, 컴퓨터가 어떤 동작을 해야 할지를 기입한 명령어의 집합이다. 프로그램을 다른 말로 소프트웨어라고도 하는데 ‘프로그래머’, ‘프로그램 개발자’, ‘소프트웨어 개발자’, ‘소프트웨어 엔지니어’는 모두 프로그램을 만드는 사람을 이야

기한다.

많은 사람이 알고 있는 대표적인 프로그램으로 ‘윈도우^{MS Windows}시리즈’를 들 수 있다. 컴퓨터를 포맷해본 적이 있는가? 본인이 직접 해본 적은 없더라도 집이나 회사, 학교에 있는 컴퓨터가 바이러스로 인해 버벅거릴 때 “컴퓨터를 밀어버리고(포맷하고) 윈도우를 새로 깐다”라는 말을 들어본 적이 있을 것이다. ‘윈도우를 깐다’라는 표현은 윈도우라는 프로그램을 컴퓨터의 저장소에 설치하는 것을 의미한다.

윈도우는 ‘운영체제 프로그램Operating System, OS’의 하나로, 컴퓨터가 처음 켜지고 나서 수행해야 할 일을 기입해둔 명령어의 집합이다. 이러한 운영체제에는 윈도우 시리즈 외에도 리눅스Linux, 유닉스UNIX, Mac OS 등 다양하다. 다만 현재 가장 일반적으로 사용되는 운영체제가 윈도우기 때문에 대부분 윈도우에 익숙할 것이다.

운영체제 이외에도 포토샵, 워드 등 컴퓨터에 설치할 수 있는 프로그램이 많은데, 이러한 프로그램은 일반적으로 운영체제 프로그램을 통해 설치하거나 접근할 수 있다. 컴퓨터의 전원 버튼을 누르면, 컴퓨터는 운영체제를 읽어 운영체제가 보여주는 화면을 사용자에게 보여준다. 그리고 운영체제는 컴퓨터에 설치된 여러 프로그램을 모아 사용자가 사용할 수 있는 방법을 제공한다. 예를 들어 윈도우에서는 아이콘 형식으로 설치된 프로그램을 보여주고 더블 클릭을 통해 사용자가 선택한 프로그램을 사용할 수 있게 해준다. 프로그램의 설치 역시 운영체제의 화면에서 설치 파일을 동작시켜 이루어진다. 즉, 운영체제는 컴퓨터라는 기계장치를 동작시키는 데 기본적으로 필요한 기초 프로그램이라 할 수 있다.

1.2 컴퓨터의 언어_고급 언어와 C 언어

앞서 설명한 바와 같이 프로그램이란, 컴퓨터가 어떤 동작을 수행해야 하는지를 미리 기재해놓은 명령어의 집합이다. 컴퓨터는 프로그램에 있는 명령어대로만 동작하며 이를 벗어나는 동작은 스스로 수행할 수 없다.

사람이라면 아주 어린 아이조차도 자율적인 프로세스에 의해 행동을 처리한다. 예를 들어 엄마가 아이에게 “은우가 가지고 논 자동차니까, 은우가 치워야죠?”라고 말한다면, 아이는 가지고 놀던 자동차를 집어 들고 있던 곳을 찾아 정리하는 과정을 스스로의 의사 결정을 통해 진행한다는 것이다. 하지만 컴퓨터에 이와 같은 동작을 명령하기 위해서는 “네가 치우세요”라는 한마디로 끝나지 않는다. 컴퓨터가 이와 같은 동작을 수행하도록 하기 위한 명령은 [팔을 오른쪽으로 x 각도로 뻗는다 → 뻗은 팔을 아래로 x 각도로 내린다 → 자동차를 잡는다 → 자동차를 x 각도로 들어올린다 → 좌로 x 센티미터, 우로 x 센티미터 이동한다 → 팔을 x 각도로 들어올린다 → 팔을 앞으로 x 각도 뻗는다 → 쥐고 있는 자동차를 내려놓는다]와 같이 동작 하나하나를 지정해줘야만 하는 복잡한 집합이 된다.

따라서 컴퓨터에 쓸모 있는 어떤 동작을 명령하려면, 우선 해당 동작을 컴퓨터가 이해할 수 있는 각 단계로 분류하고 이를 명령어로 입력해줄 프로그래머가 필요한 것이다. 컴퓨터가 이해할 수 있는 명령어를 입력하기 위해서는 프로그래머 역시 컴퓨터가 알아들을 수 있는 언어를 알고 있어야 한다. 프로그래머는 이 언어를 통해 각 단계를 명령어로 입력하여 하나의 동작을 위한 프로그램을 작성한다.

1.2.1 컴퓨터에 말 걸기

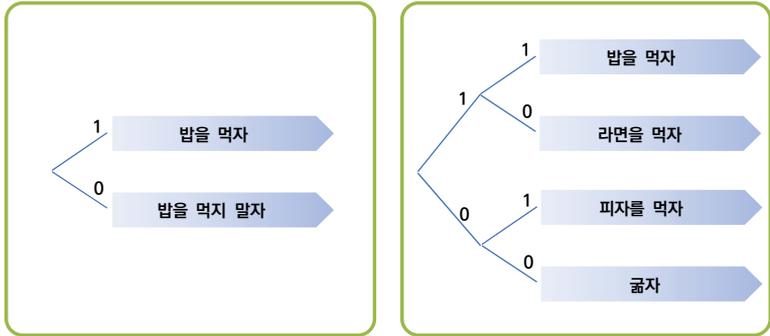
컴퓨터는 전기로 동작하는 기계 장치다. 애초에 기계는 언어를 알아들을 수 있는 유기체가 아니다. 그렇다면 어떤 언어를 써서 이 기계에 말을 걸어야 할까?

기본적으로 전기를 사용하는 기계는 두 가지 상태만 있다. 이 두 가지 상태란 즉 전기가 들어오거나 나간 상태로, 이것은 ‘on/off’, ‘1/0’, 혹은 ‘true/false’로 표시한다. 이처럼 전기가 들어오거나 나오는 두 가지 상태밖에 없는 기계가 두 가지 이상의 동작을 할 수 있도록 하는 방법이 바로, 두 가지 상태를 여러 개 이어서 인식하도록 하는 것이다.

이것이 기계 장치인 컴퓨터에서 2진수 체계를 사용하게 된 이유라고 할 수 있다.

on/off 두 가지 상태만 존재한다면
오직 두 가지의 동작밖에 할 수 없다.

on/off 상태를 중복하여 연결하면
더 많은 수의 상태를 만들 수 있다.



Special Tip 1-1 2진수 체계란?

수학을 공부하다 보면 10진수가 무엇인지 배우게 될 기회가 생기는데, 10진수는 우리가 현재 사용하는 0~9까지의 숫자로 세상의 모든 숫자를 표현하는 방법이라고 배울 것이다. 이와 달리 '2진수 표기법'이라는 것이 있는데, 이것은 세상의 모든 숫자를 0과 1, 두 가지 숫자로 표기하는 방법이다.

10진수 표기에서는 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 등 총 10개의 숫자로 모든 숫자를 표시한다. 이에 따라 9 다음의 숫자는 10, 99 다음의 숫자가 100이 된다. 반면 2진수에서는 0과 1 오직 두 가지의 숫자를 사용하기 때문에 1 다음의 숫자는 10이 되고 11 다음의 숫자는 100이 된다.

2진수와 10진수의 대응값을 표시하면 다음과 같다.

표 1-1 10진수와 2진수의 대응

10진수	0	1	2	3	4	5	6	7	8
2진수	0	1	10	11	100	101	110	111	1000

10진수와 2진수 중 어느 것을 사용하든 숫자를 붙여나가면 끝없이 무한한 숫자 표현이 가능하다. 그 때문에 둘 중 무엇을 사용하든 상관없지만, 우리는 표기의 편의성 때문에 10진수를 사용하고 있다. 2진수와 10진수 표기에 이어 16진수 표기법도 있는데, 16진수 표기에서는 알파벳 F까지 사용하여 ‘0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f’ 등 16개의 문자로 숫자를 표기한다. 가령 10진수의 숫자 ‘16’을 2진수로 표현하면 ‘10000’이 되며 16진수로 표현하면 ‘10’이 된다.

2진수 표기법에서는 모든 것을 0과 1의 숫자들이 이어진 값으로 표현하는데, 컴퓨터의 회로판과 칩셋이 바로 0과 1로 표현되는 상태를 받아 어떤 동작을 하게끔 설계된 장치다. 이렇게 0과 1로 작성되어 기계와 소통하는 언어를 ‘기계어’라고 한다.

1.2.2 C 언어는 고급 언어다. 고급 언어가 뭔데?

그렇다면 프로그래머는 기계에 말을 걸기 위해 모든 말을 0과 1로 구성된 값으로 외워야 하는 걸까? 컴퓨터가 만들어진 초창기에는 그랬다. 생각만 해도 머리 아픈 일이다. 그때만 해도 프로그래머는 0과 1로 이루어진 기계어를 통해 프로그래밍해야 했다. 그래서 프로그래밍은 그야말로 고된 노동이었고, 프로그래머를 하겠다는 사람도 많지 않았다.

하지만 곧 사람들은 이러한 고된 노동에서 벗어나기 위해 ‘어셈블리어’라는 언어와 함께 ‘어셈블리’를 만들었다. 어셈블리어는 “mov ax, 10”과 같이 무슨 말인지 모르겠지만 0과 1은 아닌, 좀 더 알아볼 수 있을 것 같은 문자로 작성되는 언어다. ‘어셈블리’는 이러한 어셈블리어에서 작성된 “mov”, “ax”, “10” 같은 단어를 읽은 다

음, 이에 대응하는 기계어로 번역해주는 장치다. 그 때문에 이 장치만 있으면 프로그램머는 0과 1로 만들어진 기계어가 아니라 조금 더 쉬운 어셈블리어로 프로그램을 작성할 수 있었다.

시간이 지나면서 사람들은 어셈블리어보다 좀 더 편한 언어는 없을까 고민했고, 자연히 어셈블리어보다 사람이 사용하는 언어에 더 가까운 고급 언어들이 만들어지기 시작했다. 이 고급 언어 중 하나가 'C 언어'다.

'고급'이란 표현은 일상에서 사용되는 의미와 동일하다. 어떤 제품이 고급화된 버전으로 다시 시장에 등장할 때에는 기존의 제품에 비해 편리성이 증가하게 마련이다. 예를 들어, 청소할 때 빗자루를 사용하다가 빗자루 대신 청소기가 등장했고, 최근에는 다시 사람이 들고 다닐 필요도 없이 알아서 돌아다니며 바닥을 청소해주는 자동 청소기가 나왔다. 고급 언어 역시 이러한 제품의 고급화와 다르지 않다. 기계어를 사용하며 불편함을 느낀 개발자들이 사용하기 더 편한 어셈블리어를 만들었으며, 이도 불편하자 그보다 더 편한 다른 언어를 만들었다. 이 최종 단계의 고급화된 언어들을 '고급 언어'라 부르는데, 이 중 'C 언어'가 포함되는 것이다.

그림 1-6 청소기의 고급화



이런 과정을 거쳐 현재 프로그래밍을 배우는 우리는 고급 언어를 공부한다. 물론 원하는 사람은 기계어나 어셈블리어를 공부할 수도 있다. 하지만 고급 언어를 통해 프로그램을 만드는 편이 수고를 훨씬 덜는 방법이다. 이 책을 통해서도 역시 고급 언어 중 하나인 C 언어를 통해 프로그램을 만드는 방법을 익히게 될 것이다.

고급 언어란 프로그램을 만들기 위해 개발된 최신의 언어들을 이야기하며, 고급 언어로 작성된 프로그램은 컴파일러를 통해 기계어로 번역될 수 있다. 고급 언어에는 C 언어 이외에도 여러 언어가 있는데 대표적인 언어를 간단히 확인해보고 넘어가기로 한다. 각 언어의 장점에 관해서는 일반적으로 알려진 사항을 언급하겠지만, 사실 이러한 장점을 제대로 이해하려면 각 언어를 공부하고 해당 언어를 이용해 프로그램을 작성해보는 것이 가장 정확한 방법이다.

FORTRAN

- 1954년 IBM 704에서 과학적인 계산을 하기 위해 시작된 컴퓨터 프로그램 언어다.
- 복잡한 계산 수행 성능이 뛰어나 공학과 자연과학 등 특정분야에서 사용되고 있다.

COBOL

- 1959년 미국 ‘데이터 시스템 언어 회의(CODASYL, Conference On Data System Language)’에서 발표한 사무 처리용 고수준 프로그래밍 언어의 하나다.

BASIC

- 초보자를 대상으로 한 프로그래밍 언어로, 1963년 미국의 다트마스 대학교에서 TSS(Time Sharing System(시분할시스템))용으로 개발했다.
- 초보자가 익히기 쉬운 대표적인 언어다.

PAS-CAL

- 1970년경 스위스 취리히 대학의 니클라우스 위스(N. Wirth) 교수가 만든 프로그래밍 언어다.
- 학생들에게 프로그램 작성을 가르칠 목적으로 개발되었기 때문에, 전산학 교육과 연구에서 널리 사용했다. 실무적인 용도로도 많이 사용된다.

C

- 1971년 미국 벨 연구소의 리치D. Ritchie가 개발한 프로그래밍 언어다.
- 현재 주로 사용되는 C++, Java 등은 모두 C 언어를 기초로 하여 확장된 것이라고 할 수 있다. 컴퓨터의 구조에 밀착한 기초 기술이 가능하고, 간결하게 표기할 수 있다는 것이 특징이다. 애초에 시스템 프로그램을 개발하기 위해 연구된 언어로, 유닉스 / 리눅스 / 윈도우 / Mac OS의 근간이 되었다.

C++

- C 언어를 확장한 객체 지향 프로그래밍 언어로, 1983년 AT&T 벨 연구소의 스트로스트룹B. Stroustrup이 발표했다.
- C 언어와 호환하면서도 객체 지향을 따르고 있어 많은 분야에서 채택되는 언어다.

Objective C

- C 언어에서 파생된 객체 지향 언어로, 1984년 브래드 콕스Brad Cox가 발표했다.
- Mac 사의 운영체제에서 동작하는 프로그램을 만들기 위해 이 언어를 사용하고 있다.

Java

- 미국의 썬마이크로시스템즈(2009년 4월 2일, 오라클에 인수됨)사가 개발한 객체 지향 프로그래밍 언어로, 1995년 5월 발표 후 1996년 1월부터 정식 버전이 배포되었다.
- C++를 바탕으로 객체 지향의 개념이 더욱 분명한 최신의 객체 지향 언어다.

위에서 언급한 언어들은 고급 언어의 역사상 저마다 큰 역할을 차지했던 언어들이다. 최근 많이 사용되는 고급 언어에는 다음과 같은 것들이 있다.

- C#, Object-C 등의 네이티브 랭귀지
- Python, Perl, Ruby, Lua 등의 스크립트 랭귀지

- PHP, ASP, JSP, JavaScript 등 웹 관련 스크립트 랭귀지

네이티브, 스크립트 언어가 무엇인지는 지금 몰라도 좋다. C 언어 외에도 이렇게 많은 언어가 있다는 것 정도만 알아두자.

1.3 컴퓨터의 기억력_ 프로그램은 누가 가지고 있나?

고급 언어로 작성한 코드는 컴파일러에 의해 번역되어 기계어로 이루어진 명령어 집합, 즉 ‘프로그램’이 된다. 그렇다면 이 프로그램은 어디에 존재할까? 누가 이 명령어를 컴퓨터에 전달해주는 것일까?

‘샤워’, ‘식사’, ‘독서’ 등의 단어는 하나의 동작처럼 보이지만, 실제로는 여러 개의 동작을 묶어놓은 ‘행동 단위’다. 아주 어릴 때에는 이 단어의 의미를 이해하지 못하지만, 자라면서 ‘샤워’라는 것이 옷을 벗고-물을 틀고-비누칠을 하고-물로 몸을 헹구고-수건으로 물기를 닦아내는 과정이라는 것을 알게 되어 기억해두게 된다. 이처럼 ‘샤워’라는 단어가 어떤 행동을 의미하는지 알게 된 사람은 필요할 때 머릿속에서 ‘샤워’를 위해 필요한 일련의 동작을 찾고 수행하게 된다. 컴퓨터에서 어떤 일련의 동작은 프로그램이라는 명령어 집합으로 묶여 컴퓨터의 기억 장치(저장 장치)에 저장되고, 저장된 프로그램은 사용자의 필요에 따라 아이콘 더블 클릭 등을 계기로 동작한다.

컴퓨터의 부품 중에는 저장 장치라고 분류되는 것이 몇 가지 있다. 메모리(RAM)와 하드디스크(HDD), 이 두 가지는 컴퓨터의 대표적인 저장 장치다. 조립 혹은 업그레이드를 위해 컴퓨터 사양을 비교해본 사람이라면 이 두 가지 저장 장치의 크기와 속도 등 그 사양을 확인해본 경험이 있을 것이다.

컴퓨터는 기계어로 동작한다. 기계어로 동작하는 컴퓨터는 모든 언어가 궁극적으로는 2진 체계로 처리된다. 따라서 데이터를 저장하는 저장 장치 역시 2진 형식의

값을 저장한다. 예를 들어 우리가 흔히 저장 장치로 잘 알고 있는 하드디스크를 열어보면, CD처럼 생긴 동그란 판이 여러 장 겹쳐 있는데, 이 판의 겉면을 덮은 물질은 자성을 띠고 있다. 초등학교에서 배우듯 자성은 N/S 두 극을 가질 수 있는데, 하드디스크는 이 판을 아주 작은 크기로 쪼갠 뒤 그 단위 하나하나의 자성을 바꾸어 0/1 둘 중의 하나의 성질을 갖도록 변경하는 형식으로 데이터를 저장한다. 하드디스크를 비롯한 모든 저장 장치는 이렇듯 2진 형태의 값을 저장할 수 있는 구조로 되어 있다. 그리고 이러한 ‘2진 형태의 값 한 개를 가질 수 있는 단위’를 컴퓨터에서는 ‘1비트(bit)’라고 칭한다.

그림 1-7 하드디스크의 내부 모습



단위 체계는 작은 것에서 큰 것으로 가며 용어가 확장된다. 무슨 말이나 하면, 길이는 밀리미터(mm)-센티미터(cm)-미터(m) 단위로 확장되며, 무게는 밀리그램(mg)-그램(g)-킬로그램(kg) 단위로 확장된다는 것이다. 저장 장치에는 얼마나 많은 데이터를 저장할 수 있는지를 나타내는 단위가 적용되는데, 방금 말한 1비트는 데이터 저장의 최소 단계며, 1비트가 8개 모인 것을 1바이트(byte), 1바이트가 1024개 모인 것을 1킬로바이트(kilobyte), 다시 킬로바이트가 1024개 모인 것을 1메가바이트(megabyte)라고 표현한다.

표 1-2 용량의 증가 단위

단위	호칭	약자	bit 환산
1 bit	비트	-	1 bit
1 byte	바이트	-	1 * 8 bit
1 kilobyte	킬로바이트	KB	1 * 8 * 1024 bit
1 megabyte	메가바이트	MB	1 * 8 * 1024 * 1024 bit
1 gigabyte	기가바이트	GB	1 * 8 * 1024 * 1024 * 1024 bit
1 terabyte	테라바이트	TB	1 * 8 * 1024 * 1024 * 1024 * 1024 bit

저장 장치의 가장 중요한 사양은 저장할 수 있는 데이터의 용량이다. 최근에는 하드디스크 시장에서 테라바이트 단위를 쉽게 찾아볼 수 있을 정도로 저장 장치의 사양이 급격하게 고급화되었다.

Special Tip 1-3 RAM과 HDD, 왜 저장 장치가 여러 개 필요할까?

RAM과 HDD는 가장 대표적인 컴퓨터의 저장 장치다. 두 가지 장치 중 일반 사람들에게 더욱 익숙한 저장 장치를 고르라면 아마 HDD(하드디스크)일 것이다. 다운 받은 영상물(드라마나 영화)이나 mp3파일, 사진 등을 모아놓는 주 저장 장소가 바로 하드디스크기 때문이다. 컴퓨터를 구입할 때 본체 내에 내장된 것을 ‘내장하드’라 부르며, USB 연결 등으로 사용할 수 있는 분리된 모양의 하드를 ‘외장하드’라 부른다.

그림 1-8 내장하드



그림 1-9 외장하드와 USB 저장 장치



RAM은 일반 사용자에게 조금 덜 친숙하지만, 게임이나 업무를 하면서 컴퓨터 속도를 업그레이드하려 했던 사람이라면 한 번쯤 들어봤을 저장 장치다. ‘컴퓨터의 메모리’라고 할 때의 메모리란 주로 이 RAM을 뜻한다. RAM이 일반 사용자에게 덜 친숙한 이유는 이 저장 장치가 ‘어떤 프로그램의 동작 시 일시적인 데이터를 저장하는 데’ 주로 사용되기 때문이다. 즉, 단순히 데이터를 저장하고 다시 확인하는 데에만 익숙한 사용자는 RAM이라는 저장 장치에 관심을 가질 일이 없으므로 낯설 수밖에 없다.

하드디스크와 RAM 둘 다 데이터를 저장할 수 있는 장치이지만, 그 구조 및 메인 모드와의 소통 방법의 차이 때문에 ‘데이터 전달 속도’ 및 ‘저장된 정보의 지속성’이 서로 다르다.

하드디스크는 전원이 off 되어도 기록된 데이터가 지워지지 않으며 용량 증가에 따른 제작 단가가 저렴하다. 이러한 장점에 때문에 하드디스크는 데이터를 영구적으로 보관하고자 할 때 많이 사용된다. 반면 RAM은 다른 저장 장치에 비해 CPU와의 소통 속도가 빠르지만, 전원이 off 되는 순간 데이터 역시 지워지는 특징이 있다. 그래서 RAM은 주로 전원이 공급되어 컴퓨터가 동작하는 동안 임시로 가지고 있어야 하는 정보 저장에 사용된다.

이처럼 정보의 특성(용량, 지속성)에 적합한 저장 장치를 사용하기 위해 컴퓨터에서는 여러 형태의 저장 장치 사용을 허용하고 있다.

1.4 프로그래밍 환경 구성하기

어셈블리로 작성된 프로그램을 기계어로 번역하기 위해 어셈블리가 필요한 것처럼 고급 언어로 작성된 프로그램도 기계어로 번역하기 위한 장치가 필요하다. 이 장치를 ‘컴파일러’라 부르며, C 언어 프로그래밍을 시작하면서 제일 먼저 해야 할 일이 바로 내 컴퓨터에 컴파일러 프로그램을 설치하는 것이다. C 언어를 위한 가장 대표적인 컴파일러는 Microsoft Visual Studio가 있다. 실제 현업에서 개발하는 경우라면 먼저 이 프로그램을 구입해야 하겠지만, 가격이 만만치 않으니 프로그래밍을 공부하는 현재의 시점에서는 해당 프로그램의 ‘평가판’을 사용한다.

다른 프로그램들처럼 Visual Studio도 해를 거듭할수록 업그레이드된 최신판이 발매되고 있으며, 현시점에서 Visual Studio의 가장 최신판은 “Visual Studio 2013” 버전이다. 먼저, 해당 제품의 홈페이지를 방문하여 설치 파일을 다운받자.

Microsoft사가 제공하는 다음 홈페이지를 통해 Visual Studio 제품군을 다운로드할 수 있다.

- <http://www.microsoft.com/visualstudio/kor/downloads>

접속 시 다음과 같은 페이지가 표시되는데, “Ultimate, Premium, Professional, Test Professional, Team Foundation Server”는 각각 Visual Studio의 기능을 필요한 범주를 달리하여 묶어 제공하는 링크들이다. 우리는 그중에서 ‘Professional 버전’을 사용할 것이다.

NOTE_ Visual Studio 2013에는 기능은 약간 제한적이지만 무료로 Key도 받을 수 있고, 계속 공짜로 사용 가능한 패키지도(Express 버전) 있다. 원한다면 해당 버전을 설치하는 것도 나쁘지 않다. 설치 방법은 다르지 않으며, 인증 방법은 많은 웹 페이지를 통해 알 수 있으니 원한다면 스스로 검색하여 인증해보기 바란다.

그림 1-10 Microsoft사의 Visual Studio 다운로드 안내 페이지

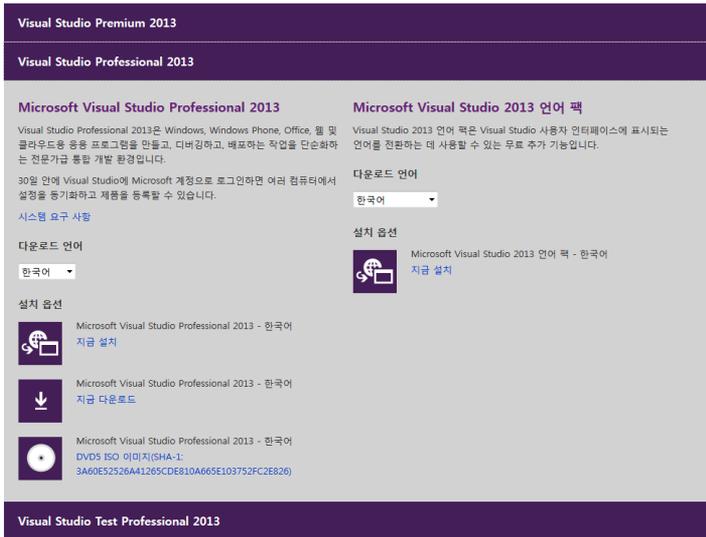


Visual Studio 2013 90일 무료 평가판

Visual Studio Ultimate 2013
Visual Studio Premium 2013
Visual Studio Professional 2013
Visual Studio Test Professional 2013
Visual Studio Team Foundation Server 2013 업데이트 2

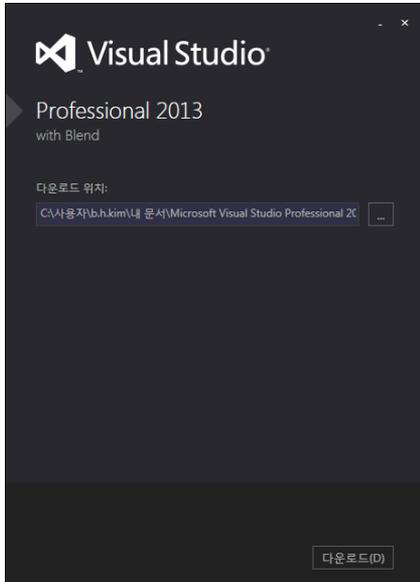
목록 중 [Professional]을 클릭하여 펼치면 다음과 같은 화면이 나타난다.

그림 1-11 Microsoft사의 Visual Studio Professional 다운로드 안내 페이지

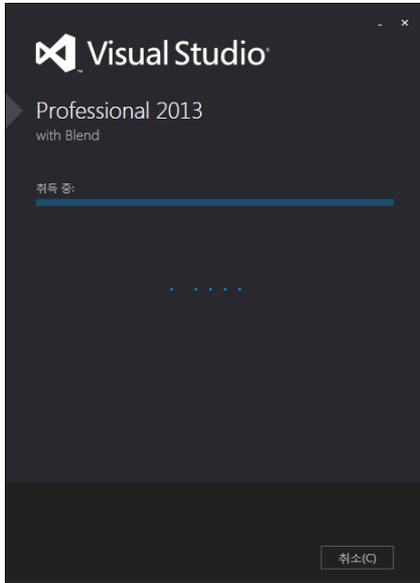


먼저, 한국어로 된 프로그램을 사용하기 위해 [다운로드 언어] 부분을 모두 [한국어]로 변경한 뒤, [지금 설치] 혹은 [지금 다운로드]를 선택한다. 다운로드된 경로에서 'vs_professional_download.exe' 프로그램을 실행하면 다음과 같은 화면이 나타난다.

NOTE 프로그램을 다운받으려면 Microsoft사의 계정이 필요하다. 계정 생성은 무료이므로, 계정이 없다면 만든 후 프로그램을 다운받도록 한다.



‘다운로드 위치’를 확인한 후 [다운로드] 버튼을 눌러 설치 프로그램을 다운받는다. 다음 화면을 보고 있다면 실제로 Visual Studio 설치 파일이 정상적으로 다운로드 되고 있다는 의미다.



화면이 다음과 같이 바뀌었다는 것은 설치 파일을 모두 다운받았다는 것이다. 마치 설치가 완료되었다는 것처럼 표시되지만, Visual Studio 판매 업체인 Microsoft 사가 외국계열이다 보니 한국어 번역 문제로 이처럼 표현된 것일 뿐이다. 이 화면은 단지 프로그램의 다운로드가 완료되었다는 것을 나타낸다.