

운영체제에 무관한 컴퓨터 그래픽 구현

김경석 · 한창민

본 논문에서는 서로 다른 운영체제(Operating System)와 컴파일러(Compiler)에서 V 공용 라이브러리(Library)를 사용하여 원시 코드(Source Code)의 수정이 필요 없이 동일하게 구현되는 그래픽 프로그램을 작성하였다. 윈도우즈에서 Visual C++ 컴파일러와 리눅스에서 g++ 컴파일러를 사용하여 그래픽 프로그램을 시험한 결과 양쪽 모두 동일하게 작동됨을 확인하였다. 이와 같이 해서 우리는 운영체제에 무관한 컴퓨터 그래픽을 구현할 수 있었다.

I. 서 론

컴퓨터의 발달로 인해 이론과 실험 중심에서의 물리학에서 전산물리 분야가 새로 생겨났으며 과거에는 엄두도 내지 못했던 일들을 이제는 컴퓨터를 사용하여 연구가 가능하게 되었다. 이러한 전산 연구의 결과를 그래픽으로 표현하는 것은 그 결과를 설명하고 이해하는 데 있어서 매우 효과적이며, 현재 컴퓨터 그래픽 프로그램은 모의 실험, 원자 구조, 함수 해석 등 많은 부분에서 응용되고 있다. 그러나 운영체제와 컴파일러마다 그래픽을 구현하는 방식이 다르기 때문에 새로운 운영체제나 컴파일러를 사용할 때에는 그에 관한 많은 것을 다시 배워야만 하고, 기존에 작성한 원시 코드 중 그래픽에 관련된 많은 부분을 수정해야만 한다.

이러한 문제점을 해결하기 위해서는 여러 운영체제와 컴파일러에서 쓸 수 있는 공용 그래픽 라이브러리를 사용하는 방법을 생각할 수 있다. 각각의 운영체제와 컴파일러에 이 공용 그래픽 라이브러리를 설치하고 이를 사용해서 그래픽 프로그램을 작성하면 운영체제와 컴파일러마다 원시 코드를 수정해야 하는 문제가 해결된다. 또한 사용자는 이 라이브러리의 설치방법과 사용법만을 배우게 되면 동일한 방법으로 그래픽을 구현할 수 있기 때문에 각각의 운영체제와 컴파일러에 대한 그래픽 구현 방법을 모두 배워야 할 필요가 없게 된다.

우리는 이러한 기능을 하는 공용 그래픽 라이브러리로서 V를 사용하였다. 이 V는 미국 표준안 협회(ANSI: American National Standard Institute)의 표준 C++을 지원하기 때문에 호환성이 높고 정의된 함수가 간단하기 때문에 사용하기 쉽다. 또한 현재 많은 운영체제와 컴파일러에서 사용할 수 있도록 지원되고 있다 [1].

이 V 라이브러리를 이용해서 1차원 본뜨기에서 쌍갈림 도표 [2]를 출력하는 프로그램을 작성하였고, 일반적으로 많이 사용하는 운영체제와 컴파일러인 윈도우즈의 MS Visual C++ 컴파일러와 리눅스의 g++ 컴파일러에서 각각 그 원시 코드를 실행하였다. 그 결과 원시 코드를 전혀 수정하지 않고도 운영체제에 무관한 그래픽을 구현할 수 있었다.

이것은 다양한 운영체제와 컴파일러를 사용하는 사람들에게 많은 시간을 절약하게 하며 그래픽의 구현을 보다 편리하게 해줄 것이라고 생각한다. 또한 물리학에서도 그레픽을 보다 쉽고 편리하게 응용할 수 있을 것이다.

II. 본 론

1. 공용 그래픽 라이브러리 V의 소개

V는 C++ 그래픽 사용자 인터페이스를 사용하여 응용프로그램을 가능한 쉽게 만들기 위해 개발되었으며 X 윈도우, Microsoft 윈도우즈(Win 3.1, Win 95/98, NT), OS/2 등의 환경에서 사용할 수 있다.

V는 표준 C++ 그래픽 사용자 인터페이스를 사용하기 때문에 각 운영체제에서 동일하게 사용할 수 있으며, 사용자가 원하는 다른 라이브러리를 더 첨가하여 응용할 수도 있고, 정의된 C++ 클래스(Class)가 간단하기 때문에 프로그램하기 쉬운 장점이 있다. 또한 V는 GNU Library General Public License 하에서 배포되는 무료 소프트웨어 이므로 경제적일 뿐만 아니라 사용자가 원하는 형태로 수정을 할 수 있고 임의로 배포할 수도 있다 [2].

2. V의 설치방법

우리는 일반적으로 많이 사용하는 윈도우즈와 리눅스를 운영체제로 선택하였고 컴파일러로는 각각 MS Visual C++ 컴파일러와 g++ 컴파일러를 선택하여 작업을 하였다. V는 gzip, tar로 압축된 상태로 배포되며 다양한 운영체제와 컴파일러에서 V 라이브러리를 만들기 위한 각각의 소스(Source)와 헤더 파일(Header File), V에 대한 자세한 정보를 담은 문서(Reference)들을 포함하고 있다.

헤더 파일은 V에 포함된 것을 변형하지 않고 그대로 사용하여 라이브러리 파일은 사용하고자 하는 각각의 운영체제와 컴파일러에서 사용자가 직접 컴파일하여 만들어야 한다. MS Visual C++에서는 배포된 Project file을 열고 난 후에 V의 헤더 파일을 불러올 수 있도록 경로를 설정하여 컴파일하였다. g++에서는 Makefile의 형태로 배포된 것을 이용하는데 Makefile은 경로 설정을 수정하고 make의 명령으로 컴파일하였다. 이러한 컴파일을 통하여 라이브러리 파일이 만들어지는데 Visual C++ 버전 4.0에서는 vmsvc32.lib, Visual C++ 버전 6.0에서는 libV.lib가 생성되며, g++에서는 Vx.a의 파일이 생성된다. 이렇게 생성된 라이브러리 파일과 다운로드한 V에 포함되어 있는 헤더 파일을 컴파일러가 찾을 수 있는 경로에 각각 복사해 넣어 V의 설치를 완료하였다.

V의 설치를 확인하기 위한 가장 간단한 방법은 V에 포함된 예제를 실행하는 것이며, 예제나 직접 작성한 프로그램을 컴파일과 링크할 때 설정하는 옵션으로 V 라이브러리를 첨가해야 하는데 그 옵션의 설정은 MS Visual C++에서는 /Lvmvc32.lib(v4.0), /LlibV.lib(v6.0)이고, g++에서는 -lVx이다. 그리고 각각의 컴파일러가 필요로 하는 인터페이스에 관계된 라이브러리를 첨가시키는 것은 각각 /Lcomctl32.lib과 -lXaw -lXmu -lXt -lXext -lX11이다.

3. 운영체제에 무관한 컴퓨터 그래픽 구현 예제

공용 라이브러리 V를 사용하여 그래픽이 동일하게 구현되는 것을 확인하기 위해 혼돈 거동을 보이는 가장 간단한 동력학계인 다음과 같은 1차원 본뜨기를 예로 선택했다 :

$$x_{t+1} = f(x_t) = 1 - Ax_t^2$$

(x_t : 이산시간 t 에서의 상태).

그리고 비선형 맷음변수 A 를 바꿔가면서 계의 접근상태인 꿀개(attractor)를 보여주는 쌍갈림 도표를 출력하는 그래픽 프로그램을 작성했다.

그림 1과 2는 Visual C++ 컴파일러와 리눅스 g++ 컴파일러 각각에서 위의 1차원 본뜨기 그래픽 프로그램의 원시 코드를 수정없이 실행해서 얻은 쌍갈림 도표들이다. 각

그림들은 A 가 증가함에 따라 무한연속 주기배가가 일어나고, 임계점을 지나면서 혼돈전이가 일어나는 동일한 쌍갈림 도표들을 보여주고 있다. 이와 같이 해서 서로 다른 두 운영체제와 컴파일러에서 동일한 그래픽이 구현됨을 확인할 수가 있었다.

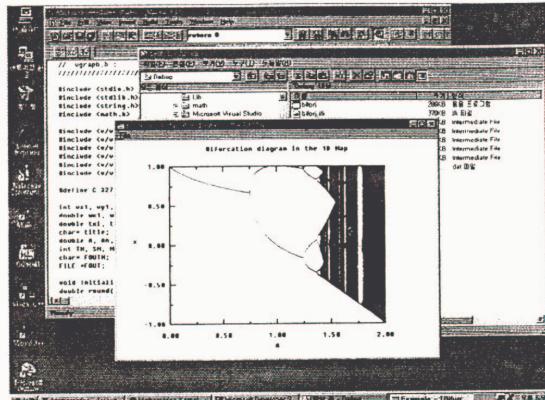


그림 1. 1차원 본뜨기에서의 쌍갈림 도표를 윈도우즈(Windows)의 Visual C++에서 V를 사용하여 실행한 결과.

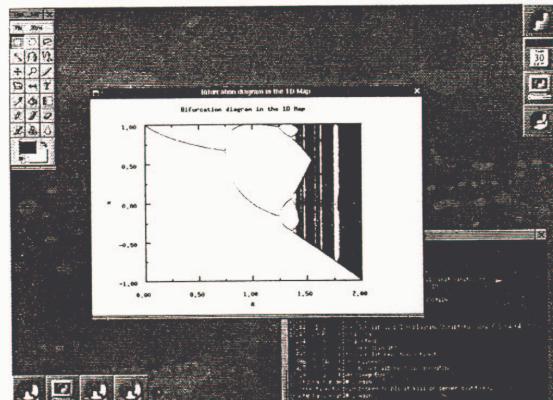


그림 2. 1차원 본뜨기에서의 쌍갈림 도표를 리눅스(Linux)의 g++에서 V를 사용하여 실행한 결과.

III. 결 론

우리는 V를 사용하여 1차원 본뜨기에서 쌍갈림 도표를 출력하는 프로그램을 작성하였고 윈도우즈의 Visual C++ 컴파일러와 리눅스의 g++ 컴파일러에서 각각 실행을 시켰다. 그 결과 운영체제에 무관하게, 즉 원시 코드의 수정없이 동일한 그래픽이 구현됨을 확인하였다. 이것은 다양한 운영체제에서 물리학을 연구하기 위해 필요한 그래픽 프

로그램의 상호 교환을 수월하게 하여 유용하게 사용될 것이다.

감사의 글

이 논문을 지도해 주신 김상윤 교수님과 처음부터 끝까지 주의를 기울여 도와주신 박사 과정을 밟고 있는 임우창 선배님께 많은 감사를 드립니다. 그리고 함께 동고동락하면서 도움을 준 여러 동기들에게도 고마움을 전합니다. 끝으로 물리학과의 많은 발전이 있기를 바랍니다.

참 고 문 현

- [1] URL이 <http://www.objectcentral.com/> 인 웹 사이트를 방문하면 공용 라이브러리 V를 받아올 수가 있다.
- [2] S. H. Strogatz, *Nonlinear Dynamics and Chaos* (Addison-Wesley, New York, 1997), Chap. 10.