프로그래밍 언어 논술 에세이 2

넓고 넓은 열린 세상에의 편안한 중독을 꿈꾼다.

서울대학교 공과대학 컴퓨터공학부
최 종 식

차 례

- 1 언어의 기반은 이론이다.
- 2 관성을 경계한다.
 - 2.1 언어의 큰 물결을 비판한다.
 - 2.2 열린 시각으로 바라보자.
- 3 오만을 경계한다.
 - 3.1 과거를 돌아본다.
 - 3.2 비판하는 것은 무엇인가?
 - 3.3 하지만 자유로운가?
 - 3.4 우수한 언어에 대한 종교논쟁
- 4 아는 만큼 보인다.
 - 4.1 익숙함이 주는 편안함에 취하다.
 - 4.2 좋은 언어는 있다.
 - 4.3 언어는 중요한 것이 아니다.
- 5 특성별 언어의 통합이 필요하다.
 - 5.1 좋은 설계의 역설
 - 5.2 통합 언어 출범의 역사를 쓰자
- 6 프로그래밍 언어학은 창조의 학문이다.
 - 6.1 프로그래밍 언어학은 창조의 학문이다.
 - 6.2 패러다임의 변화는 충분한 준비가 필요하다.
- 7 넓은 열린 세상에의 편안한 중독을 꿈꾼다.

1 언어의 기반은 이론이다.

대전제. 언어의 기반은 이론이다. 지난 글에서는 프로그래밍 언어를 디자인 하는데 이론에 기반을 둔 설계를 해야 하는 이유를 이야기 했다. 현실의 벽과 싸우면서 발달한 C 언어의 과정과 대비되는, 엄밀한 정의를 통해 발달해 온 언어에 대한 이론들도 살펴보았다. 이번 글에서는 관성적인 언어선택을 비판하는 몇 가지 글을 읽고, 앞으로 언어에 대해서 취해야 할 관점을 어떻게 발전시켜야 할지 생각해보고자 한다.

2 관성을 경계한다.

2.1 언어의 큰 물결을 비판한다.

모든 글에서, 대부분의 사람들이 다른 사람들이 많이 쓰는 언어를 관성적으로 선택하는 현실을 비판하고 상황에 따라 적합한 프로그래밍 언어를 선택해야 하는 필요성에 대해 이야기 한다. 그리고 대부분의 글에서 Lisp과 mL을 좋은 예로 언급하고 있는데 이들은 수학을 기반으로 발달한 언어이기 때문에 기계에서 유래된 타언어들에 비해 해당 방향의 강점을 필요로 하는 여러 상황에서 유리한 위치를 가진다. 따라서 적합한 프로그래밍 언어의 선택은 그 언어에 익숙하지 않은 프로그래머라면 초기에 불편을 겪기는 하겠지만 그 언어의 강점이 살려지게 된다면 경쟁력을 가질 수 있게 된다는 내용이다.

2.2 열린 시각으로 바라보자.

언어는 적합한 곳에 적합한 디자인을 할 수 있는 것으로 선택해야 한다. 하드웨어를 기술하는 언어의 선택이 lisp일 수는 없을 것이다. 또한 구조적인 프로그래밍이 필요한 곳에 C언어를 쓸 수는

없을 것이고 고속의 연산을 필요로 하는 곳에 스크립트 언어를 쓸 수는 없을 것이다. 목적에 맞는 좋은 디자인을 할 수 있는 언어를 선택해야 한다.

중요한 것은 열려있는 시각이다. 앞서서도 말했듯이 많이 쓰는 언어, 자신이 익숙한 언어를 관성적으로 사용하고, 타 언어들에 대해서 배타적인 시각을 가지는 것은 좋지 않다. 물론 그러한 선택이 항상 선택 성공을 보장하는 것은 아니지만, 다른 언어의 장점을 인정할 수 있는 열린 시각은 꼭 필요하다.

3 오만을 경계한다.

3.1 과거를 돌아본다.

이전의 글에서 우리가 비판했던 것은 중구난방으로 기계에 맞춰서 디자인되면서 발전한 언어들이었다. 그 비판과 더불어 궁극적으로 프로그래밍 언어는 이론에 기반을 두는 곳으로 가야 한다는 이야기를 했다. 하지만 어떤 이론이 프로그래밍 언어 설계에 도움이 되는 것인지에 대한 이야기는 아직까지 논란도 많고 존재하지도 않는 것 같다. 이전에도 말했듯이 지금도 C 언어와의 전투는 계속되고 있고, 그들의 의견도 일견 타당한 점이 있는 것은 사실이다. 하드웨어의 측면에서 발전시킨 이론 위에서 C언어는 훌륭한 언어임에 틀림이 없다. 아마도 올바른 개념적 프로그래밍 언어를 위해서는 프로그래밍 언어 분야뿐만 아니라 하드웨어 분야도 바뀌어야 할지도 모르는 일이다. 하지만 기계적인 기반의 이론이든 수학적 기반의 이론이든 어떤 이론적 배경을 바탕으로 언어가 설계되어야 한다는 점은 주지의 사실이라고 감히 말할 수 있을 것이다.

3.2 비판하는 것은 무엇인가?

이번 글들에서 비판하는 것은 앞에서 언급되었듯 프로그래밍 언

어 선택에 있어서 사람들의 관성이다. 메이저 언어들에 오직 그 이유만으로 선택하는 사람들을 비판하고 좀 더 넓은 시각으로 언어를 바라보고, 받아들일 장점이 있다면 받아들이라는 것이다.

3.3 하지만 자유로운가?

이 글을 쓴 글쓴이들은 위에서 말한 논지에서 전혀 자유롭지 않다. 논지전개의 방법이나 그 논리가 옳지 않다는 말은 아니지만 'Lisp와 mL만이 그런 자신의 논지에 맞는 언어다'라는 생각이 기저에 깔려있는 상태에서 글을 읽는 사람들에게도 그러한 선택을 하도록 유도하고 있는 것이다. 한 글에서는 자신의 목적이 절대로 생각을 바꾸려는 것이 아니라고 말하고 있지만, 사실 그런 것이 아니라는 것은 글의 내용상 쉽게 알 수 있다. 어떤 논지를 전개하면서는 그런 특성을 가지기 위한 필요조건들을 이야기해야하는데 어느 글에서건 그런 조건을 이야기 하는 것이 아니라 mL을 이야기 하고 Lisp을 이야기 한 뒤에 이 언어들에 그 조건을 가지고 있다고 이야기 한다.

3.4 우수한 언어에 대한 종교논쟁

'언어는 반은 기술이고 반은 종교다'라는 말은 정말로 현재의 상황을 획기적으로 표현한 말이라고 생각한다. 주변에도 각각의 언어를 대변하는 사람들이 있는데 각자의 논리들을 듣고 있으면 그런 생각이 든다. 개인적으로 항상 경계하는 것들 중 하나가 내가 남들보다 높은 식견을 가졌다고 생각하는 우를 범하는 것이다. 물론 이런 생각조차 스스로의 비판의 틀에서 자유로울 수는 없지만 그래도 경계해야 할 만한 것이라고 생각한다. 그런 면에서 분명한 것은 이 글들에서 글쓴이들이 Lisp 종교에 빠져서 다른 언어를 사용하는 대부분의 사람을 생각이 없는 사람 혹은 고급언어의 장점을 모르는 사람으로 생각하고 있다는 것이다. 하지만 Lisp을 통해서

성공을 했다는 글의 내용에 역설적으로 일부의 성공 사례를 이렇게 찾아서 봐야 할 정도의 언어라면 이미 그 경쟁력에서 패퇴해 있는 것에 다름 아니다. 성공의 조건은 프로그래밍 언어에 있는 것이 아니라는 현대의 기본적인 사실을 간과해선 안 될 것이다. 글쓴이들은 Lisp보다 더 좋은 언어가 나온다면 자신이 익숙한 Lisp 따위 버리고 그 언어를 선택할 수 있을 것인가?

4 아는 만큼 보인다.

4.1 익숙함이 주는 편안함에 취하다.

자신이 편한 언어를 계속해서 쓰는 사람은 그 언어가 편한 것이 당연하다. 그리고 그 언어의 틀 속에서 요구하는 모든 기능들을 만들어 낸다. 생각의 폭은 해당 언어의 틀에서 갇혀버린다. 이것은 개인적으로도 실제로 느끼고 인정하는 사실이다. 베이직을 가장 먼저 배웠기 때문에 재귀호출에 대한 명확한 개념이나 메모리 주소값, 객체 같은 다른 언어의 요소들에 대해서 적응하는데 시간이 오래 걸렸다. 그런 개념들을 새로이 배웠을 때의 충격이란 이루 말할 수 없었다.

어떤 언어에 익숙해진다고 하더라도 최소한 다른 언어가 가지는 장점, 그 언어를 대표하는 개념들은 알아야 할 필요가 있다. 언어의 선택에서도 생각의 틀을 넓히는데도 도움이 된다. 좁고 넓은 세상에의 편안한 중독에 빠져서 헤어 나오지 못하면 곤란하다. 모든 언어가 튜링완전하다는 점에서는 동등하기 때문에 튜링머신으로 프로그래밍 하는 것이 익숙하다고 튜링머신으로 프로그래밍 하고 있을 수는 없는 노릇 아닌가?

우리가 관심을 가지는 것은 언어의 표현한계가 아니라 프로그래머의 고생한계인 것이다. Blub 역설에서 이야기 되었던 재귀를 지원하지 않더라도 원하는 기능을 결국에는 다 구현할 수 있고, 그래픽 가속기와 같이 어떤 분야에서 특정하게 자주 쓰이고 속도가 중

요한 명령어들이 있다면 분명히 기존 명령어 셋으로 다 표현할 수 있다. 그렇다고 하더라도 재귀를 쓸 수 있다면 쓰고, 가속기를 쓸 수 있다면 쓰는 것이 이득인 것이다.

4.2 좋은 언어는 있다.

좋은 언어는 있다. 분명한 사실이다. 당장 같은 언어에서도 버전이 계속해서 업데이트 되고 표준위원회에서 표준이 새로 생긴다. 그런 과정을 통해서 더 좋은 언어를 만들 수 없다면 그 많은 사람들의 노력은 헛수고가 될 것이다. 지금도 많은 사람들이 더 좋은 언어를 위해서 연구하고 있다. 그리고 언어간에서도 각각의 장점이 있고, 우월한 방향이 있다. 좋은 언어란 해당 프로그램을 작성하는데 좋은 표현을 할 수 있는 언어이다. 프로그래머는 사람이 읽을 수 있는 코드를 작성해야 한다.

4.3 언어는 중요한 것이 아니다.

프로그램의 작성에는 언어의 종류가 중요한 것이 아니라 언어가 가지고 있는 표현력, 개념적인 구조가 중요하다. Python이니 ASP니 싸우고 있을 것이 아니라 해당 언어가 가지고 있는 표현력이 내가 만들 프로그램에 어떤 의미가 있을지를 생각해 보아야 할 것이다. 이론에 기반을 두었다고는 하나 Lisp, mL같은 언어들도 어떤 분야에서나 쓸 수 있는 완벽한 언어라고 볼 수는 없다. 완벽한 언어로는 가야할 길이 멀다.

5 특성별 언어의 통합이 필요하다.

5.1 좋은 설계의 역설

좋은 설계는 단순하고, 좋은 설계는 문제를 제대로 해결한다. 우

리의 문제는 무엇인가? 이것은 명백하게도 우리가 원하는 프로그램을 제대로 된 설계를 해서 쉽게 만드는 것이다. 하지만 우리의 환경은 어떠한가? 어떤 프로그램을 만들고자 할 때 고려해야 할 언어의 다양함은 단순함을 이미 넘어서고 있고, 조금만 큰 프로그램에 다다르면 그 어떤 언어도 그 프로그램을 작성하는데 모든 방면에서 장점을 가지는 것이 없다. 문제를 제대로 해결하는 언어가 없다는 말이다. 이것은 각각의 언어를 연구 개발하는 사람들이 스스로 내뱉는 좋은 언어와 좋은 설계의 특징에도 스스로 역설적이다.

5.2 통합 언어 출범의 역사를 쓰자

과거의 언어 발달과정과 비슷한 역사를 가진 것들 중 하나는 아마도 운영체제를 비롯한 컴퓨터 플랫폼일 것이다. 표준이 없었기 때문에 기업체에서 컴퓨터 시스템을 구축할 때마다 어느 회사의 어느 운영체제는 어떤 장점을 가지는지 분석하는 것이 중요한 선택이었다. 시스템을 업그레이드 할 때에는 새로운 시스템을 도입하기에는 이미 익숙해져 있는 시스템을 버리지 못하고 관성적으로 한 회사에 종속되는 등의 문제가 있었다. 오랜 암흑 시절을 보내고 POSIX 와 같은 표준이 제정되고 나서야 사용자들이 비로소 자신의 권리를 찾고 하나의 좋은 운영체제를 발전시켜 나가면서 자신의 목적에 맞게 사용할 수 있는 토대가 마련이 되었다.

난립하는 언어를 통합하자. 이 말은 서로간의 특성을 무시하는 획일화된 통합을 의미하는 것이 아니다. 프로그래밍 언어가 갖추어야 할 특성들을 합의하고 표준을 제정하자. 그리고 운영체제가 그러했고 그러하듯이 어떤 특성이 필요한 분야를 구분하고 특성별로 확장 가능한 언어를 구축하자. 우리가 쓰는 언어 개발을 언제까지나 언어 공급자에게 맡길 수는 없는 노릇 아닌가.

6 프로그래밍 언어학은 창조의 학문이다.

6.1 프로그래밍 언어학은 창조의 학문이다.

프로그래밍 언어는 사람들이 새로 만들어 나갈 수 있는 언어이다. 인간의 언어는 함부로 통합하자는 말을 할 수 없고, 오랜 역사와 함께 만들어진 언어를 어느 한순간 사람들이 표준을 제정한다고 새로 만들 수 있는 것은 아니다. 하지만 프로그래밍 언어는 그렇지 않다. 사람들이 노력한다면 올바른 언어를 만들 수 있다. 일시적으로 기존언어 사용자들의 종교적인 믿음을 기반으로 한 반발이 있을 수 있겠지만 그들이 주장하는 언어의 장점들을 대부분 포괄하고 있는 좋은 언어라면 머지않아 통합 표준 프로그래밍 언어가 그 자리를 대신할 수 있을 것이다.

표준 언어를 제정하자. 필요할 수도 있겠지만 일단은 기존의 세상을 완전히 뒤엎는 변화를 요구하는 것은 아니다. 현대의 여러 언어 서로의 장점을 인정하고 수용하자.

6.2 패러다임의 변화는 충분한 준비가 필요하다.

좋은 설계는 무더기로 등장한다고 했듯이 좋은 언어 또한 언어에 대한 담론이 성숙해진 분위기 속에서 갑작스럽게 찾아올지도 모른다. 분위기 뿐만 아니라 우리도 많은 준비가 필요하다. 버릇을 버리고, 종교를 버리자. 각각의 장점을 인정하고 무교가 되자. 적절한 목적에 적절한 언어를 사용하자고 하는 것은 과거의 적절한 목적에 적절한 컴퓨터시스템과 운영체제를 구입하자는 것에 다름 아니다. 프로그래밍 언어가 가져야 할 특성에 대한 공감대를 만들고, 정리하고, 표준을 제정하자.

넓고 넓은 세상에의 편안한 중독을 꿈꾸자. 보다 나은 환경에서 프로그래밍 언어를 만들어 나가고 사용하자. 물론 이것이 그 종교에 귀의하자는 말은 아니다. 우리는 항상 열린 마음을 가지고 또

다른 새로운 패러다임의 변화에도 놀라지 않을 준비를 하면서 살아 갈 것이다.

7 넓은 열린 세상에서의 편안한 중독을 꿈꾼다.

서로 다른 언어를 하는 사람들끼리 벌이는 자존심 싸움도 없고, 이것도 저것도 공부해야 적응할 수 있는 척박함도 없고, 하나의 언어에 매몰되어 사고의 틀이 갇히는 좁은 세상도 아닌 넓은 세상에서, 그 세계에 안주하는 것만이 아닌 열린 세상에서, 익숙하고 편안한 중독을 꿈꾼다.

끝