

지금의 정보기술을 보는 눈높이

이광근*
KAIST

[도올고신], 제 38신, pp.25-26, 2001년 6월

정보통신의 화려한 기술은 현란하기 이룰데 없지만, 실상은 불안하고 위태롭습니다. 이 불안의 한 원인은 소프트웨어들이 가지고 있는 많은 수의 알수없는 “버그”(bug)들입니다. 영어의 버그는 “버러지”를 뜻하지만, 전산분야에서는 컴퓨터의 오동작을 일으키는 소프트웨어의 틀린점을 일컫습니다.

우리가 쓰던 컴퓨터가 갑자기 아무 반응이 없어지고 모든걸 다시 시작해야 하는 경우, 이것은 소프트웨어의 버그 때문입니다. 컴퓨터를 함부로 사용해서가 아니고, 컴퓨터를 움직이는 소프트웨어가 틀린 명령을 컴퓨터에게 내렸기 때문입니다. 핸드폰 전화를 쓰다가 단추작동이 제대로 되지 않아서 대리점에 들어서 핸드폰의 시동을 다시 걸어야 했던 일, 그것도 핸드폰을 작동하는 소프트웨어가 틀렸기 때문입니다. 자동차의 급발진 사고들, 이것또한 자동차의 엔진 시동장치를 운영하는 소프트웨어의 문제일 거라고 저는 90% 이상 확신합니다. 인천공항의 수하물 자동 처리시스템의 오동작, 이것도 수하물 처리 기계장치들을 운영하는 소프트웨어의 문제입니다. 1996년 6월 4일 유럽의 아리안 로켓 501호가 발사후 약 40초만에 폭발했습니다.¹ 이것도 로켓을 운영하는 소프트웨어의 사소한 문제, 딱 하나였던것으로 밝혀졌습니다.² 1999년 12월 화성에 도착해서 여러실험을 하기로 한 Mars Polar Lander 모듈이 화성궤도에서 화성표면으로 내려가는 도중에 사라져 버렸습니다.³ 하강중에 역추진 로켓이 점화되지 않아 추락해 버린것으로 추정하고 있는데, 그 원인도 소프트웨어의 문제였던 것으로 좁혀졌습니다.

*현재 서울대학교 컴퓨터공학부 소속

¹www.cnn.com/WORLD/9606/04/rocket.explode/index.html

²www.esrin.esa.it/tidc/Press/Press96/press33.html

³mars.jpl.nasa.gov/msp98/lander/

소프트웨어의 버그가 특히 불안해 지는 것은, 공기중의 산소와 같이 세상의 모든 구석구석에서 우리의 일상에 없어서는 안될 존재로 스며드는 컴퓨터때문입니다. 지금도 이미 전화/전기/금융/교통의 인프라를 운영하는 것이 컴퓨터들이기는 하지만, “지구 = 컴퓨터”라는 슬로건이 과장이 아니게 컴퓨터가 세상을 덮어가고 있습니다. 언젠가는 (혹시, 10년내에는) 지구상의 모든 컴퓨터들은 서로의 거리를 느끼지 못할만큼 빠른 네트워크로 모두 연결될 것이고, 전 세계 모든 컴퓨터들은 각자 하나의 부품으로서 지구위에 거대한 스케일과 역동적인 모습의 네트워크 컴퓨터를 구성하게 될 것입니다. 지금의 굳어있고 느린 인터넷과는 달리 시시각각 한없이 그 연결이 변화하면서 눈 깜짝할 사이의 소통능력으로 하나의 컴퓨터 같이 살아가는 것입니다. 지구의 대기권에서 소용돌이 치면서 생성되고 사라지는 구름들의 동영상을 일기예보 시간에 본적이 있다면, 그 구름속의 물방울들 하나하나가 우리의 몸이나 주변에 있는 컴퓨터가 되는 것이고, 수시로 변하는 그 구름의 모습이 바로 컴퓨터끼리의 소통을 통해서 무언가를 운영해가는 지구 컴퓨터의 모습이라고 생각할 수 있습니다. 이러한 환경에서는, 모든 소프트웨어들은 지구위를 광속으로 돌아다니면서 필요한 곳에서 필요한 자원(소프트웨어 부품, 데이터, 마이크로프로세서, 메모리등)을 적시에 동원하여 작업을 끝마치는 것이 일상적이 됩니다. 손바닥 위의 컴퓨터, 장난감 속의 컴퓨터, 책상위의 컴퓨터, 가전제품들에 숨은 컴퓨터, 응접실/침실/화장실 가구의 컴퓨터, 자동차의 컴퓨터, 은행의 컴퓨터등이 전세계 불특정 다수의 컴퓨터와 서로 정보와 소프트웨어를 주고받으며 필요한 작업을 진행하게 되는 것입니다.

이러한 세상을 운영하면서 지구위를 쏘다니는 무수히 많은 소프트웨어들이 “버그”로 우글된다면, 그 세상은 컴퓨터가 없느니만 못할 것입니다. 내 핸드폰으로 내려받은 게임소프트웨어가 실행중에 내 전화번호부를 삭제버린다; 아침에 일어나 불일을 보고 났더니, 변기의 컴퓨터가 내 건강상태에 대한 잘못된 데이터를 담당 의사에게 전달한다; 약국의 약장을 휘집고 다니며 처방전대로 약을 제조하는 로봇팔의 작동이 오류가 생겨서 처방과 다른 알약을 섞는다; 누군가의 맛나는 요리법이라고 오븐에 내려받아 재료를 넣고 굽기 시작했더니 오븐이 과열로 화재가 난다; 입고 있는 컴퓨터가 오동작으로 내 허리를 심하게 압박해 온 이후로 변비로 고생한다.

소프트웨어의 버그는 어쩔 수 없이 생기는 천재지변이 아니고 사람이 만들어내는 실수입니다. 소프트웨어를 만드는 것은 사람이기 때문입니다. 작성하는 중에 도구의 도움으로 많은것이 자동화 된다고 해도, 소프트웨어를 만들어주는 그 도구들도 모두 소프트웨어이고 이것 역시 사람의 손으로 만들어진 것들 뿐입니다. 사람에 의해 기획되고 사람에 의해 만들어지지 않는 소프트웨어는 없습니다. 따라서, 소프트웨어의 버그는 외부에서 스며든 “버러지”때문이 아니고, 소프트웨어를 만드는 사람이 잘못생각해

서 잘못된 명령을 컴퓨터에게 전달하도록 프로그램을 구성해서 생기는 문제입니다.

소프트웨어의 버그란 한마디로, 프로그램이 생각대로 작동하지 않는 것입니다. 이것은 컴퓨터가 프로그램 한대로 충직하게 실행하지 않아서가 아니고, 그 프로그램 자체가 생각한 대로 짜여져있지 않기때문입니다. 말도 안 되는 프로그램, 이것이 소프트웨어의 버그입니다.

전산학의 중요한 한 과제는 프로그램이 생각한대로 작동하는지를 실행시키기 전에 자동으로 검증하는 기술을 이룩하는 것입니다. 프로그램의 버그를 자동으로 잡아주는 기술입니다. 소프트웨어가 기획한대로 (생각대로) 구현되었는지, 그 소프트웨어가 작동하기 전에 엄밀하고 안전하게 자동으로 확인할 수 있는 기술, 이것을 이룩하기 위해서 지난 40년간 전산학에서는 어떤일이 있어왔는지, 그리고 궁극적인 위의 목표를 위해서 어떤 공부들이 필요한지는 다음기회에 계속 논의해 보겠습니다.