

## 1 무엇 때문에 이문서가 있는가?

Jargon File과 기타 유사하고 유명한 문장의 작가로서, 나는 가끔씩 열정에 차 넘치는 인터넷 초보자들의 질문을 받곤 한다. 요지는 한마디로 다음과 같다. "어떻게 하면 훌륭한 해커가 될 수 있죠?" 이상한 것은 어느 하나의 FAQ 혹은 Web으로 된 문서가 이렇게 중요한 문제에 대해 설명 해 놓은 것이 없다. 그래서 나의 생각을 적어 보려고 한다.

만약 당신이 지금 오프라인에서 이 문서를 보고 있다면 최신버전을 유의해주시길 바란다. 영문버전은 <http://www.catb.org/~esr/faqs/hacker-howto.html>에서 얻을 수 있다.

주의: 이 문서의 마지막에 FAQ가 있다. 나한테 이 문서에 대한 물음을 제기하기 전에 두 번은 읽어보길 바란다.

현재 이문서는 여러 언어로 된 버전이 있다. 한국어에서 찾아볼 수 있다. 이 문서는 수시로 번역이 되기 때문에, 현재 당신이 읽고 있는 것은 낡은 버전일 수도 있다.

## 2 해커란?

"해커 사전(Jargon File)" 에는 "해커"에 대해서 많은 정의를 해 놓았다. 그 대부분은 기술적으로 정통한 사람과 문제를 해결하고 한계를 극복하면서 즐거움을 구하는 사람들과 깊은 연관을 맺고 있는 정의들이다. 만약 여러분이 해커가 되는 방법을 알고 싶다면, 이러한 정의 중 단 두 가지 만이 관련된 것이다.

몇 십 년 전 첫 번째 소형컴퓨터가 만들어지고 ARPAnet 프로젝트가 금방 시작되었을 때, 이 때 전문 프로그래머와 인터넷 마법사들로 구성된 공유된 문화공동체가 있었다. 이런 문화의 구성원들이 "hacker"란 단어를 만들어냈다. 이들 해커들은 인터넷을 만들었다. 그리고 해커들은 UNIX 운영체제를 오늘날의 모습으로 가꾸었다. 해커들은 유즈넷을 운영하고 있고, 월드 와이드웹이 제대로 돌아가게 만들었다. 만약 여러분이 이러한 문화의 일부이고, 또한 기여하고 있다면, 이 문화의 사람들은 여러분을 해커라고 부른다. "당신은 해커입니다"라고 말이다.

해커정신은 소프트웨어해커에만 국한되지 않는다. 일부사람들은 해커정신을 다른 일에 사용하곤 한다. 예를 들면 전자공학이나 음악 사실 당신은 기타 과학이나 음악의 최 상위 활동에서도 이것을 발견할 수 있다. 프로그래머들은 이런 사람들에게 동류의식을 느끼고 그들도 해커라고 부른다. 그리고 혹자들은 해커들의 천성은 해커들이 종사하고 있는 특정 매체에 구애되지 않는다고 한다. 그러나 이 글에서 우리는 소프트웨어 해커의 기법과 자세, '해커'라는 말을 창시한 공유된 문화의 전통에 초점을 맞춰 얘기를 전개시킬 것이다.

자신들을 해커라고 소리 높여 부르는 또 다른 집단이 있다. 과연 그럴까? 대답은 "아니오"다. 이들은 주로 10대의 청소년들인데, 스릴을 느끼려고 컴퓨터 시스템과 전화 시스템을 프리킹(phreaking)하는 자들이다. 진정한 해커들은 이들을 '크래커(cracker)'라고 부르고 있으며, 자신들과는 아무런 상관관계가 없기를 바라고 있다. 진정한 해커들은 대체로 크래커들을 게으

르며, 무책임하고, 그다지 현명하지 못하고 실력 없는 자들이라고 여기고 있다. 전문적으로 다른 사람의 컴퓨터보안을 파괴 하는 행위는 당신을 해커로 만들지는 못한다. 쇠사슬로 자동차를 도둑질했다고 해서 자동차 엔지니어가 될 수 없는 것처럼 컴퓨터 보안 시스템을 깬다고 해서 당신이 해커가 될 수 있는 것은 아니다. 불행히도, 거의 대다수의 저널리스트들과 작가들은 크래커들을 묘사하는 말로 "해커"라는 말을 사용하여 스스로를 웃음거리로 만들고 있다. 이들의 이러한 행위는 진정한 해커들을 정말 화나게 만드는 꼴이다.

해커와 크래커와의 근본적인 차이는 무엇인가? 해커들은 사상(事象, things)을 건설하지만, 크래커들은 사상(事象)을 파괴하는 자들이다.

아직도 해커가 되고 싶다면, 계속 읽어주기 바란다. 그런데 크래커가 되고 싶다면 ALT.2600 뉴스그룹에 가라. 그리고 자신이 생각했던 것만큼 똑똑하지 않음을 깨달았다면, 이 뉴스그룹에 올라온 게시물을 그대로 따라 하라. 이것이 크래커들에 관해서 말할 수 있는 것의 전부이다.

### 3 해커의 마음가짐

해커들은 문제를 해결하고, 사상(事象)을 건설한다. 그리고 자유와 자발적인 상호 부조(扶助)의 신념을 믿는다. 해커로서 인정을 받으려면, 여러분은 이러한 유형의 마음가짐으로 자신을 무장해야 할 것이다. 단순히 마음가짐만으로는 안 된다. 반드시 실천이 따라야 한다. 그렇다고 결과 속이 달라서는 않될 것이다. 해커 문화에서 인정을 받기 위한 방편으로 해커식 마음가짐을 배양하려 한다면, 여러분은 주제의 초점을 잃은 것이나 진배없다. 이러한 것을 믿는 유형의 사람이 된다고 함은 여러분 스스로 끊임 없이 동기 부여를 하고 스스로 학습해야 된다는 것이다. 이것이 중요한 점이다. 모든 창조적인 예 술과 마찬가지로, 명인(名人)이 되기 위한 가장 효과적인 방법은 거장(巨匠)들의 사고방식과 심적 경향을 모방하는 것이다. 지적으로 뿐만 아니라 정서적으로도 말이다.

가능하게 아래의 시조가 위에 뜻을 잘 표현 했을 것이다.

To follow the path: (이 길을 따르라)  
look to the master, (달인을 찾으라)  
follow the master, (달인을 따르라)  
walk with the master, (달인과 함께 걸으라)  
see through the master, (달인을 분석하라)  
become the master. (달인이 되라)

그래도 해커가 되고 싶다면, 다음의 문구들을 깨달을 때까지 암송하십시오.

#### 3.1 이 세계에는 해결을 기다리는 매력적인 문제들이 산적해 있다.

해커로 가는 길은 상당한 즐거움을 안겨다 준다. 그러나 엄청난 노력이 동반되는 즐거움이다. 노력을 통해서 동기 유발이 된다. 성공적인 운동선수는 자신들의 신체를 단련하고, 과거의 자신의 육체적 한계를 극복하는 과정에서 일종의 육체적 즐거움으로부터 동기 유발이 된다. 이

와 같이 해커가 되어간다는 것은 문제를 해결하고, 자신의 기술을 예리하게 가다듬고, 자신의 지능을 함양 하는 과정에서 원초적인 전율을 느끼는 것이다. 만약 당신이 천성적으로 이런 종류의 사람이 아니고 또 해커가 되고 싶다면, 당신은 모든 방법을 동원해서 이러한 사람이 되기에 노력을 해야 한다. 그렇지 못하다면 결국에는 여러분의 해킹기력(氣力)은 돈 섹스, 사회적 인정과 같은 일탈적인 것에 침탈당하게 될 것이다.

(그리고 자신만의 학습 능력에 따라 일종의 신념을 개발해야만 한다. 다시 말해 비록 문제를 해결 하는데 필요한 온갖 것들을 알고 있지 못한다 할지라도, 과감히 도전하고, 이로써 또 배우고 배워 문제를 완전히 해결할 때까지 도전과 학습을 해나가는 것과 같은 신조와 성실성 따위를 자기 철학으로 세워야 할 것이다.)

### **3.2 그 누구도 한 문제에 두 번씩이나 해결하려들면 안 된다.**

창조적인 두뇌 활동은 가치 있는 것이자 유한의 자원이다. 해결되어야 할 새로운 문제들이 산적해 있는데, 또 다시 바퀴를 재발명하는 것과 같은 일에 지력과 시간을 낭비해서는 안 된다. 해커처럼 행동하려면, 여러분은 다른 해커들의 사고와 시간도 매우 귀중한 것이라는 믿음이 있어야 한다. 즉, 여러분은 정보를 공유하고, 문제를 해결하고, 그리고 나서 그 해법을 널리 알려, 다른 해커들이 예전의 문제를 한도 끝도 없이 재차 다루기보다는 새로운 문제를 해결할 수 있도록 하는 배려는 일종의 해커의 도덕률과도 같은 것이다. 그렇다고 여러분이 행한 창조적인 작품 모두를 나누어주라는 얘기는 아니다. 비록 이렇게 철저한 공유의 원칙을 지키는 해커들은 다른 해커들에게서 최대의 찬사와 존경을 받고 있기는 하지만 말이다. 최소한의 생활비와 컴퓨터 임대비용을 지불할 만큼으로 자신의 창조적인 작품을 파는 것은 “해커 가치”와는 전혀 모순되는 것은 아니다. 그리고 가족을 부양하거나 부자가 되는데 자신의 해킹 기술을 이용하는 것보다도 모순되지 않는다. 단 여러분이 해킹을 하는 동안에 자신이 해커임을 잊지 않는다면 말이다.

### **3.3 권태와 단조로운 일은 악(惡)이다.**

해커들(일반적으로 창조적인 사람들)은 절대로 지루함을 느껴서는 안 되며, 어리석고 반복적인 작업에 매달려서도 안 된다. 이러한 일의 발생은 할 수 있는 일을 하지 못한다는 뜻이기 때문이다. 즉, 새로운 문제를 해결하는데 들여야 할 시간과 노력을 이런 지분한 일에 빼앗겼기 때문이다. 이러한 낭비는 모든 사람들에게 상처를 입히는 결과를 빚는다. 따라서 권태와 단조롭게 반복되는 일은 불쾌한 일이기도 하지만 엄밀히 말해서 악인 셈이다. 해커처럼 행동하기 위해서는, 여러분은 따분한 정보를 가능한 한 많이 자동적으로 날려 보낼 정도로 이것을 신조로 삼아야 한다. 당신 자신을 위해서뿐만 아니라 모든 사람(특히 다른 해커들)을 위해서도 말이다. 한 가지 예외는 있기는 하다. 해커들은 관조자의 눈에는 반복적이고 따분하게 보이는 것을 아무 생각 없이 실습하기도 한다. 이때는 특정 기술의 습득 또는 여러분에게는 없는 특정의 경험을 하려는 목적이다. 그러나 생각이 있는 사람이라면 그 누구라도 이러한 권태의 타성에 굴복해서는 안 된다.

### **3.4 자유는 좋은 것.**

해커들은 타고난 반 권위주의자들이다. 여러분에게 지시나 명령을 내릴 수 있는 사람이라면 여러 분이 매료된 온갖 문제의 해결을 저지시킬 수 있다. 그리고 권위주의적인 마음가짐의 소유자들이 하는 일의 방식은 황당할 정도로 어리석기까지 하다. 그러니 권위주의적인 자세와 직면하게 될 때마다 투쟁해야 한다. 적어도 당신과 다른 해커들이 질식사하지 않도록 말이다. 이는 모든 권위와 싸우라는 얘기는 아니다. 아이들은 지도를 받을 필요가 있고, 범법자들은 어느 정도 구속이 필요할 수 있다. 해커들은 명령을 좇아 허비하는 시간 이상으로 그가 원하는 것을 얻기 위해서는 일부 권위를 받아들일지도 모른다. 그러나 이러한 수용은 제한적이고 의식(意識)적 인거래(去來)이다. 권위주의자들이 원하는 일종의 개인적 굴복 같은 것은 제공할 성질이 아니다.(이런 권위에 굴복을 조장하는 것은 매스미디어도 그 한몫을 한다. 그들에게는 기사 거리이자 수익을 창출할 수 있는 또 다른 이벤트이기 때문이다--역주) 권위주의는 검열과 비밀주의로 변성한다. 그리고 이들은 자발적인 협력과 정보 공유를 믿지 않는다. 즉, 이들은 자신들이 “통제할 수 있는 협력”을 좋아한다. 해커처럼 행동하려면 여러분은 이러한 검열 및 그 제도, 비밀주의, 그리고 책임감있는 성인들을 구축(驅逐)하는 권력과 기만 수단에 본능적인 적의를 노정시킬 수 있도록 자신을 개발(開發)해야 한다. 그리고 이 신념에 따라 기꺼이 행동할 수 있어야만 한다.

### 3.5 마음가짐이 되었다고 능력이 없어도 된다는 얘기는 아니다.

해커가 되려면, 여러분은 이러한 자세를 개발해야 한다. 자세만 가지고 뛰어들면 해커가 될 수 없을 것이다. 그 이상의 어떤 것이 있어야 챔피언 또는 록스타가 될 수 있다. 해커로 가는 길은 지력과 실천, 헌신(獻身), 그리고 상당한 고역이 동반되는 길이다. 그러므로 여러분은 마음가짐만으로 안주하지 말고, 온갖 종류의 능력을 존중해야 한다. 해커들은 시험관들이 자신들의 시간을 어영부영 보내도록 내버려두지 않을 것이다. 해커들은 능력을 귀하게 여긴다. 특히 해킹 능력을 말이다. 어떤 것에 대한 능력은 좋은 것이다. 그 누구도 거장(巨匠)이 될 수 없는 기술에 관한 능력은 훌륭한 것이라 말할 수 있다. 정신적 예리함과 교묘한 재주, 고도의 집중력이 포함된 기술에 대한 능력은 더 말할 나위도 없다. 만약 여러분이 능력을 숭배한다면, 여러분은 능력 개발 자체를 즐길 것이다. 처절한 실천적 노력과 헌신은 단조로운 일보다는 일종의 철저한 유희가 될 것이다. 해커가 되기 위해서는 이 점이 가장 중요한 요소이다.

## 4 해커가 알아야 할 기술들

해커의 자세는 매우 중요하다. 그러나 그 기술은 절대적으로 필요한 것이다. 자세만으로 무능을 보상할 수 없다. 그리고 자신이 해커로 불리기를 꿈꾸기 전에 갖추어야 할 어떤 기본적인 기술이라는 도구(toolkit)가 필요하다. 이러한 수단은 테크놀로지가 새로운 기술을 창출하고 옛 것을 쓸모없는 것으로 만들면서 서서히 변하고 있다. 기계어 프로그래밍과 최근의 HTML을 놓고 볼 때 격세지감을 느끼지 않을 수 없다. 현시점에서는 다음에 기술하는 내용을 분명히 포함할 것이다.

## 4.1 프로그래밍 하는 방법을 배워라.

물론 이것은 가장 기본이 되는 해킹 기술이다. 1997년까지 여러분이 반드시 배워야 할 프로그래밍 언어는 C이었다. 어떤 컴퓨터 프로그래밍 언어도 모른다면, 나는 Python 을 추천한다. 이 언어는 산뜻하게 설계되었으며, 문서화 또한 잘되어 있고, 비교적 초보자들이 접근하기 쉬운 프로그래밍 언어이다. 비록 처음 접근하는 언어이기는 하지만, 장난감 같은 언어는 아니다. 이 언어는 매우 강력하고 유연하며, 대형 프로젝트에도 적합하다. 나는 python에 대한 구체적인 평가서를 작성했다. 그리고 학습서는 파이썬 웹사이트에서 구할 수 있다.

java도 프로그래밍에 입문하기 좋은 언어다. python에 비해 어렵지만 생성된 코드의 실행속도는 많이 빠르다. 또 java는 프로그래밍에 입문하기만을 위한 것이 아닌 아주 훌륭한 언어이다.

여러분은 단 한두 가지의 컴퓨터 언어만 알고 있다면 해커가 아니다. 심지어는 프로그래머의 수준도 안 된다. 여러분은 어떤 한가지 언어에 편향되지 않는 보편적 방식으로 프로그래밍 문제에 관해 사고하는 방법을 배울 필요가 있다. 진정한 해커가 되려면, 매뉴얼에 기술된 내용과 당신이 알고 있는 것을 연관시켜, 며칠 이내에 새로운 언어를 배울 수 있는 수준까지 오를 필요가 있다. 이는 여러분이 최소한 매우 다른 몇 가지 언어를 익혀야 됨을 뜻하는 말이다.

진지한 프로그래밍을 하고 싶으면, C 언어를 익혀야 할 것이다. C 언어는 유닉스에서 핵심이 되는 언어이다. C와C++ 는 아주 흡사하다, 만약 둘 중에 하나를 익혔다면 다른 하나를 배우는것도 쉬울것이다. 그러나 이 두언어 모두 초보자가 프로그래밍을 시작하기에는 적합하지않다.. 실제로 C언어를 적게 쓰면 쓸수록 당신의 작업효율은 높아질것이다.

C는 아주 효율적이고 자원을 절약한다. C언어의 고효율은 당신이 컴퓨터최하층(예: 메모리)에 대한 직접적인관리를 토대로 하는 것이다. 이러한 코드는 아주쉽게 버그를 만들어낼수있고 많은 시간을 디버깅에 투자하여야한다. 요즘 컴퓨터는 점점 빨라지고있다. 이는 얻는 것보다 잃는 것이 더많은 것이다. 비교적 총명한선택은 실행속도가늦고 효율이낮은 하지만 대폭적으로 당신의 시간을 절약할수있는 언어를 선택하는 것이다. 그러기에 python 을 선택한다.

해커에게 있어 특히 중요한 여타 언어들은 LISP 과 Perl 이다. Perl은 역동적인 웹 페이지를 작성하고 시스템을 관리하는 데 폭넓게 쓰 이는 언어라는 매우 실질적인 이유에서 배울 가치가 있다. Perl로 작성하지는 못한다 할지라도 이를 읽고 이해할 수는 있어야 한다. 많은 사람들이 perl를 사용하는 이유는 내가 여러분들에게 python을 추천하는 이유와 같다. 모두 C가 하지않아도될 고효율적인 작업을 하기위해서다. 당신은 그 코드들의 내용을 이해하는것을 필요할것이다.

리스프(LISP)는 마침내 도달하게 될 심오한 계발 경험을 위해 배울 가치가 있다. 그 경험은 여러분의 앞으로의 프로그래머로서의 기간 동안 더욱 훌륭한 프로그래머가 되게 해주는 경험일 것이다. 비록 여러분이 리스프 자체를 실제로 그다지 많이 사용하지 않는다 할지라도 말이

다.

5가지 언어(Python, JAVA, C/C++, Perl, 그리고 LISP) 모두를 배운다면 더할 나위 없겠다. 이 언어들은 가장 중요한 해킹 언어가 될 뿐더러, 이것들 각각은 프로그래밍에 대한 접근 방법은 서로 다르며, 여러분에게 매우 소중한 접근법으로 가르침을 줄 것이다.

여기서 어떻게 프로그래밍 언어를 익혀야 하는가를 구체적으로 기술할 수는 없다. 매우 복잡한 기술(skill)이기 때문이다. 그렇지만 시중의 서적이나 강좌들은 그렇지 못하다는 것을 밝히지 않을 수 없다(거의 대다수의 해커들이 독학한다). 굳지 말하라면, (1) 코드를 읽고 (2) 코드를 작성하는 것이다.

프로그램을 배운다는 것은 일상의 언어로 매우 좋은 글을 쓰는 것과 같다. 이렇게 하기 위한 최선의 방법은 명인(名人)들이 작성한 자료를 읽고 스스로 습작을 해보고, 많은 글을 읽고 작성해보고, 더욱 많은 글을 읽고 좀 더 많은 글을 작성하고.... 한마디로 당신이 작성한 글들이 여러분들만의 틀로 일종의 힘과 경제성을 개발하기 시작할 때까지 반복에 반복을 거듭하는 길 뿐이다. 읽기 좋은 코드를 찾기란 매우 어렵다. 그 이유는 소스 상태로 애송이 해커들이 알고 어설프게 다룰 수 있는 대형 프로그램을 구할 수가 없기 때문이다. 이러한 상황은 극적으로 변화를 보고 있다. 지금은 소스 공개(open-source) 소프트웨어, 프로그래밍 툴(tool), 운영 체제(이 모든 것들은 해커들이 만든 것이다)들을 어디서든 구할 수 있다.

#### 4.2 소스 공개 유닉스들 중 하나를 구해서 사용법과 운용법을 배워라.

여러분이 개인용 컴퓨터를 갖고 있거나 이에 접근할 수 있다는 전제 하에 이 글을 쓴다. 어떤 신참(newbie) 해커들이 해커 기술들을 습득할 수 있는 가장 중요한 그 첫 단계는 리눅스 배포본이 나 BSD 유닉스 계열의 운영체제를 구하여, 여러분의 컴퓨터에 설치하고 이를 돌리는 일이다. 이 세상에는 유닉스 이외의 무수히 많은 운영체제들이 있다. 그러나 이것들은 바이너리(binary) 형태로 배포되기 때문에 소스 코드를 읽고 분석하여, 이를 수정할 수 없다. 도스나 윈도우, 또는 매킨토시 운영체제에서 해킹하는 법을 배우려는 것은 몸 전체에 깁스를 하고 춤추는 법을 배우는 것과 같다. 게다가 유닉스는 인터넷의 운영체제이다. 유닉스를 모르고 인터넷을 배울 수 있지만, 인터넷을 이 해하지 못하고 인터넷 해커가 될 수는 없다. 이러한 연유로 오늘날의 해커 문화는 매우 강한 유닉스 중심적인 문화이다.(이 얘기는 언제나 진실이지는 않다. 그리고 일부 원로 해커들은 이러한 얘기에 불편한 심기를 노정한다. 그러나 유닉스와 인터넷과의 공생(共生) 관계는 마이크로소프트의 힘으로도 심대한 타격을 줄 수 없을 만큼 매우 강력한 것이 되었다.

유닉스를 양육하시오 - 내 자신은 리눅스를 좋아 하지만, 다른 방법도 있다. 유닉스를 공부하고 사용하고 설정하라. 유닉스를 통해서 인터넷과 대화하라. 코드를 읽어라. 코드를 수정하라. 여러분은 마이크로소프트의 운영 체제가 꿈도 꿀 수 없는 것 이상의 더 말할 나위없는 프로그래밍 툴들을 얻게 될 것이다(C를 포함하여, Lisp과 Python 및 Pears도). 그리고 즐거움을 누릴 것이며, 거장 해커가 되어 회고하면서 끝없이 배워야 한다는 사실을 깨닫는 것보다도 더욱 풍부한 지식에 흠뻑 빠지게 될 것이다.

유닉스 학습에 대해서 좀 더 많은 것을 알려면, The Loginataka를 참조하시오.

리눅스를 연습하여 익숙해지려면, 리눅스를 입수할 수 있는 곳을 참조하시오.

BSD 유닉스 도움말 및 자원을 구하고자 한다면, <http://www.bsd.org/>를 참조하시오.

나는 "Unix와Internet기초" 라는 문장을 쓴 적이 있다.

(주: 나는 여러분이 이제 갓 시작한 초보(newbie)라면 단일 프로젝트로써 리눅스나 BSD를 설치하기를 권하지 않는다. 리눅스의 경우라면 각 지역별 리눅스 사용자 모임에 도움을 청하거나, LISC(Linux Internet Support Co-Operative)와 접촉하라. LISC는 여러분이 도움을 얻을 수 있는 IRC채널을 운영하고 있다.)

#### **4.3 월드 와이드 웹(WWW)의 사용법과 HTML 작성법을 배워라.**

해커 문화가 건설한 대부분의 것들은 정말로 훌륭한 업적으로 가득 차 있다. 非해커들이 삶의 방식에 어떤 현시적인 충격을 주지 않고 공장, 사무실, 그리고 대학이 돌아갈 수 있게 하였다. 웹은 유일한 예외이다. 정치인들마저도 인정했던 거대한 해커 장난감이 이제는 세상을 변화시키고 있다. 이러한 이유 하나만으로도 여러분은 웹이 어떻게 작동하는지를 배워야 한다.

이 말은 단지 누구나 할 수 있는 브라우저 사용법을 익히는 것뿐만 아니라, HTML를 작성하는 법도 배우라는 말이다. 만약 여러분이 프로그램 하는 법을 모른다면, HTML 작성은 프로그램 작성에 도움이는 되는 정신적인 기질(habits)을 여러분에게 가르쳐 줄 것이다. 자! 우리 모두 홈페이지를 구축합시다.(인터넷에는 좋은 자료들이 많다)

그러나 홈페이지를 구축하는 것이 여러분을 해커로 만들어 줄 수 있는 충분조건은 아니다. 웹에는 홈페이지로 가득 차 있다. 이들 대부분은 핵심과 중심 과제가 없는 진흙창과 같은 것이다. 그 것도 여러분의 마음을 빼앗을 수 있는 매우 호화로운 진창들이다. 그렇다고 파리똥이 똥이 아니라고 말할 수 있겠는가.(좀 더 자세한 것을 알고 싶으면 The HTML Hell Page를 참조하라)

여러분의 웹페이지가 가치 있는 것이 되려면 알찬 내용으로 가득 차야 한다. 재미있을 뿐더러 다른 사람(해커)들에게도 유용한 것이야 한다.

#### **4.4 만약 아직 실용성 영어를 모른다면, 배워라.**

미국사람으로서 또 영어를 모국어로 하는 사람으로서 예전에 나는 이민을 꺼내기 싫었다. 문화상의 제국주의가 될 수 있기 때문이다. 하지만 상당수의 기타언어를 모국어로 하는 분들이 이 문제를 제기할 것을 요구해왔다. 그것은 바로 영어는 인터넷에서의 공용언어이라는 것이다. 여러분들이 순리롭게 해커사회에서 활동하려면 반드시 알아두어야 한다.

이것은 확실한 것이다. 대개 1991년 당시 나는 해커사회에서 해커들이 기술을 서로 주고받을 때 영어를 사용한다는 것을 알았다. 심지어는 그들이 사용하는 언어가 갖고 영어는 그냥 제2언어인 경우에도 말이다. 내가 알아본바 에 의하면 현재 영어는 다른 언어들보다 더욱 많은 기술적용어를 보유하고있다. 그러기에 작업하기에는 매우좋은 도구이다. 같은원인 으로 하여 영어로된 기술문서번역은 보통 이상적 이지 못하다. (만약 번역이 되었다면)

핀란드인인 Linus Torvalds. 영어로 자신의 코드에 주석을 달고(분명한 것은 이 행동이 그가 한 우연한행동은 아닐것이다) 유창한 영어가 그가 전세계범위내의 리눅스개발자들을 관리하는 중요한 요소였다. 이것은 본받아야할 예제이다.

## 5 해커 문화에서의 신분

화폐 경제가 아니었던 대부분의 문화처럼, 해커왕국은 명성으로 영위되고 있다. 여러분이 흥미로운 문제를 풀려고 한다. 그런데 그 문제들이 얼마나 재미있는 것인지, 여러분의 해법이 진정 훌륭한 것인지는 오로지 여러분의 동료 또는 선배들의 판단 여하에 달려 있다. 따라서 여러분이 해커 게임을 할 때, 우선 점수를 유지하는 법을 배워야 한다. 이것으로 다른 해커들이 여러분의 기술이 어떠한지를 판단한다(이것이 다른 해커들이 계속해서 여러분을 해커라고 부를 때까지는 여러분이 해커가 될 수 없는 이유이다). 이러한 사실로 미루어 볼 때 해킹이 고독한 작업이라는 이미지는 정확히 사실을 반영한 것이라고는 볼 수 없다. 또한 자아(自我)와 외부의 확인이 각자의 동기부여 에 포함되어 있다는 사실을 인정하지 않는 해커 문화적 금기(지금은 점점 퇴색되어 가고 있지만, 여전히 잠재적인 영향을 끼치고 있다)라는 것도 현실의 반영은 아니다. 특히, 인류학자들은 해커왕국을 “선물 문화”라고 부르고 있다. 여러분은 해커 왕국에서 신분과 명성을 얻는다. 이는 타인을 지배함으로써가 아니라, 또한 아름다워지거나 타인이 원하는 것을 소유

함으로써 이루어지는 것은 아니다. 오히려 여러분이 갖고 있는 것을 나누어줌으로써 얻어지는 신분과 명성인 것이다. 특히, 여러분의 시간, 창의력, 체득된 기술의 결과물을 나누어줌으로써 말이다.

해커들에게서 존경받는 일을 할 수 있는 것 중 가장 기본이 되는 5가지를 제시한다.

### 5.1 소스 공개(open-source) 소프트웨어를 작성하라.

우선(가장 핵심적이고 전통적인 것), 다른 해커들이 생각하는 프로그램이 재미있고 유용하도록 작성한다. 그리고 그 프로그램 소스를 전(全)해커 문화가 사용할 수 있도록 나누어 준다.

우리는 이것을 “공개(무료, free) 소프트웨어”라고 부른다. 그러나 “무료(free)”라는 말이 정확히 어떤 의미를 내포하는지를 알지 모르는 대다수의 사람들에게는 혼동만 가중시키는 꼴이 되었다. 우리 대다수의 해커들은 이 용어를 “open-source” 소프트웨어라고 부르기를 좋아한다.



해커왕국에서 가장 존경받는 인물(demigod)들은 포괄적인 필요에 걸맞은 대형·성능이 우수한 프로그램을 작성하여, 모든 사람들이 이용할 수 있도록 이 프로그램을 나누어 주는 사람이다.

## 5.2 소스 공개 소프트웨어의 분석(test)과 디버그(debug)를 할 수 있도록 도와라.

또한 소스 공개 소프트웨어를 유지하고 디버그 하는 것에 봉사한다. 우리가 사는 이 불완전한 세 상에서 어쩔 수없이 우리는 디버깅하느라고 우리들의 대부분의 소프트웨어 개발시간을 허비한다. 소스 공개 저작자들이 여러분에게 훌륭한 베타테스터(beta-tester, 소프트웨어 증상을 명확하게 기술하고, 문제점들을 정확히 집어내고, 졸속적으로 내놓은 판이기 때문에 어느 정도 버그들이 있을 수 있다는 것을 수용하고, 그리고 아주 간단한 진단 루틴들을 기꺼이 적용할 수 있는 사람들)는 루비라는 보석에 그 가치를 덧붙이는 사람이라고 말하는 이유가 여기에 있다. 심지어 이들 테스터들은 지지부진하고 소모적인 악몽과 같은 디버깅 국면과 그저 유익한 뉘앙스가 있는 디버깅 국면과의 차이점을 변별할 수 있는 사람들이다.

만약 여러분이 초보자(newbie)라면 여러분이 흥미를 갖고 훌륭한 베타테스터가 될 수 있는 현재 개발 중인 프로그램을 찾아보기 바란다. 이는 프로그램을 테스트하는 것을 돕는 것에서 디버깅할 수 있도록 돕고, 이를 수정할 수 있도록 돕는 자연스럽게 진보적인 절차이다. 이를 통해서 많은 것을 배울 것이며, 후에 여러분을 돕게 될 사람들과 좋은 인연을 만드는 것이기도 하다.

## 5.3 유용한 정보를 출간하라.

또 다른 선행(善行)으로는 유용하고 재미있는 정보를 수집하고 여과하여, 웹페이지나 FAQs(Frequently Asked Questions lists)와 같은 문건으로 만들어, 누구든 이용할 수 있게 만드는 일이다.

주요 기술적인 FAQs를 유지·개선하는 사람들은 소스 공개 저작자들만큼이나 존경을 받는다.

## 5.4 하부구조가 견실히 작동할 수 있도록 도와라.

해커 문화(그리고 인터넷의 공학적 개발)는 자원자들에 의해 운영된다. 메일링 리스트 관리, 뉴스 그룹의 조정 작업, 대형 소프트웨어 아카이브 사이트(archive site)를 유지하고, RFC와 기술 표준을 개발하는 것같이 반드시 유지·지속되어야 하는 것이 있지만 그리 매력적인 일은 아니다. 이러한 일을 제대로 하는 사람들은 대단한 존경을 받는다. 왜냐하면 이러한 작업은 엄청나게 많은 시간이 소요되며, 코드를 갖고 노는 것처럼 그다지 흥미로운 것이 아니기 때문이다. 철저한 헌 신적 봉사가 아니라면 이를 수 없는 일이다.

## 5.5 해커 문화 그 자체에 봉사하라.

마지막으로, 여러분은 해커 문화 자체에 봉사하고, 그 사상을 전파할 수 있다(예로, 해커가 되

는 법에 관한 정확한 입문서를 작성하는 것 따위). 이러한 위치에 도달하기까지는 앞에서 기술한 것 들을 제대로 이해하고 실천하고 나서야 비로소 이루어질 수 있는 것이다.

엄밀히 말해서 해커 문화는 지도자들은 없다. 그러나 그 문화의 영웅들, 부족 원로들, 그리고 역 사가와 대표자들은 있다. 여러분이 이러한 최전선의 진지를 지켜왔을 때야 비로소 해커 문화의 일원으로 성장할 것이다. 경고 : 해커들은 그들의 부족 원로들 중에서 떠들썩한 자들을 믿지 않는 다. 일종의 명성을 위한 가시적인 접근은 위험한 것이다. 오히려 이러한 속 빈 강 정들과 싸워라. 여러분은 여러분 자신들만의 지위를 획득해야 한다. 그래야 명예를 얻을 수 있다. 그리고 난 다음 에 여러분의 신분에 고나 해서 겸손과 자비로운 마음을 항시 마음에 새 겨 실천해야 한다.

## 6 해커(hacker)와 숙맥(nerd)간의 관계

대중적인 신화와는 달리, 여러분은 해커가 되려고 숙맥이 될 필요는 없다. 그러나 도움은 된다. 사실상 많은 해커들이 숙맥들이기는 하다. 사회적으로 의지할 것이 없다는 생각은 해킹처럼 정령 중요한 일들에 집중할 수 있는 여건을 마련해주는 셈은 된다.

이러한 연유로, 많은 해커들은 자궁(自矜)의 상징으로써 '숙맥(nerd, 샌님, 꾀)'이라는 딱지가 붙는 다. 심지어는 좀 더 험한 용어인 '괴짜(奇人, geek)'라는 말도 사용한다. 이는 정상적인 사회적 기대감에서 자신들은 독립된 존재임을 선언하는 방식이다. 좀 더 확장된 논의를 원하면 The Geek Page 를 방문하시오.

여러분이 해킹에 익숙해질 만큼 충분히 어떻게 해서든 전념할 수 있고, 여전히 삶을 영위하고 있다면, 다행이다. 지금은 내가 1970년대의 열쫓이(newbies) 시절을 보냈을 때보다는 훨씬 수 월하게 보낼 수 있다. 다시 말해 주류 문화가 이전보다는 기술 숙맥(techno-nerd)들에게 훨씬 호의를 보이 고 있기 때문이다. 해커들은 종종 관찮은 연인이자 배우자임을 깨닫는 사람이 점점 많아지고 있다.

생활을 영위할 수 없다는 이유로 해킹에 끌린다면, 그것도 괜찮다. 적어도 전념에 곤란을 겪 지 않을 테니까 말이다. 아마도 여러분은 언젠가는 분명 해커가 될 것이다.

## 7 해커 품격 갖추기

해커가 되기 위해서는, 여러분은 해커의 사고방식과 태도를 체득해야 한다. 여러분이 컴퓨터로 도움을 줄 수 없을 때, 몇 가지 할 수 있는 일이 있다. 그렇다고 이것도 해킹의 대체물일 수는 없다. 그 어떤 것도 해킹과 대체할 수 있는 것은 없다. 그러나 많은 해커들은 다음과 같은 노력을 기울이고 있으며, 일부 기본적인 처세술과 해킹의 본질은 그 맥을 같이하고 있다고 느낀다.

과학 소설을 읽어라. 과학 소설 집회에 참석하라(해커와 원형 해커들을 만날 수 있는 좋은 방

법이다).

선(禪)을 익히거나 병법(兵法)을 익혀라.(정신 수양은 무엇보다도 중요하다. 즉, 노자의 도덕경(道德經), 장자(莊子) 등.)<sup>1)</sup>

음악을 통해 분석적 청각을 개발하라. 특별한 음악을 감상하는 법을 배워라. 또한 악기 연주 법 과 창법(唱法)을 배워라.

이음동의(異音同義)의 익살과 어휘적 유희의 변별력을 개발하라.

여러분의 모국어로 좋은 글을 쓰는 법을 배워라.(거의 대다수의 해커들은 훌륭한 작가가 될 소질이 있다.)

상기 언급한 것을 이미 행하고 있다면, 여러분은 타고난 해커적 재목감이라 할 수 있다. 언급한 내용들은 단독적으로 적용되는 것은 아니다. 서로 얽기 설기 얽혀져 있다. 우리의 두뇌 좌/우반구를 모두 개발해야 한다(해커들은 논리적인 추론과 문제의 명확한 논리 외의 단계로까지 끌어 올 릴 수 있는 능력이 필요하다).

마지막으로 해서는 ‘안 될 것’을 적어둔다.

어리석고 뽐내는 듯한 ID나 스크린 명을 사용하지 마라.

유즈넷 상에서 flame wars에 끼어들지 마라(그 어디서든지).

여러분 자신을 ‘cyberpunk’라고 부르지 말 것이며, 쓸데없이 시간을 낭비하는 사람들에게 여러분의 시간을 할애하지 마라.

오자투성이에 문법에 어긋나는 글을 게시하거나 전자우편으로 보내지 마라.

이러한 짓을 하게 된다면, 명성은 하루아침에 가을에 떨어지는 낙엽과 같은 꼴이 된다. 해커들은 상당히 좋은 기억력의 소유자들이다. 어떤 것을 받아들이면, 최소한 수년 동안 잊지 않는다. (그리고 한국철학과 사상을 익히는 것이 우리들에게는 좀 더 직접적이고, 시사하는 바가 크다.)

인터넷아이디가 갖는 문제에 대해서 다소 부연 설명하고자 한다. 직함의 裏面에 자신의 정체성을 숨기는 행위는 청소년이나 크래커, warmer d00dz, 그리고 여타 저열한 생명체의 어리석은 행동 특성이다. 해커들은 이러한 짓을 하지 않는다. 해커들은 자신이 한 일에 자긍심을 갖으며 그러 한 일들이 자신의 실명과 연관되기를 원하다. 그러니 아이디가 있다면, 버려라. 해커 문화에서 직함 이란 패자의 표식일 뿐이다.

## 8 기타 자원

피터 시바흐(Peter Seebach)는 해커와 함께 일하는 방법을 모르는 관리자들을 위해서 훌륭한 Hacker FAQ를 작성해 두고 있다. 피터의 사이트가 반응이 없다면, Exite 검색엔진을 이용하면 그 사본을 찾아볼 수 있을 것이다.

Loginataka에서는 유닉스 해커들의 적절한 훈련법과 자세에 관해 알려준다.

나는 해커왕국의 소사(小史)(A Brief History Of Hackerdom, <http://www.tuxedo.org/~esr/writings/hacker-history/hacker-history.html>)에 관한 글을 썼다.

나는 리눅스와 소스 공개 문화가 어떠한 것인지에 관해서 묘사한 글인 성당과 시장 (우리나라의 경우는 <http://kldp.org>나 각 PC 통신사의 관련동회에서 동일 문건에 대한 번역본을 찾아볼 수 있다.) 라는 글을 썼고, 그 후 상기 글의 속편으로 認智圈의 택지화 에서 좀 더 구체적으로 개진했다.

나와 Rick Moen이 함께 "질문의 지혜"라는 글을 썼다. 당신이 효율적으로 도움을 얻을 수 있을 것이다.

만약 당신이 PC, UNIX와 INTERNET의 기본개념과 동작원리를 알고 싶다면 The Unix and Internet Fundamentals HOWSO를 참고하라.

만약 당신이 프로그램이나 기타팩스를 배포하려고한다면 "프로그램배포조례"를 따르도록 노력하라.

## 9 빈번한 질문들(FAQ)

질문 : 나에게 해킹하는 법을 가르쳐 줄 수 있소?

이 웹페이지가 개설되고 난 후에 나는 "해킹에 관해 모든 것을 가르쳐 달라"라는 요청을 받아왔다. 불행히도, 나는 이러한 일에 투입할 여력이 없다. 나는 내 해킹 프로젝트를 수행하는데 내 능력과 시간에 거의 110%를 쏟아 붓고 있다.

내 처지가 이렇다. 그러나 해킹이란 본질적으로 독학으로 얻어지는 자세이자 기술인 것이다. 만약 진정한 해커들이 당신을 돕기를 원할 때, 여러분이 그들이 알고 있는 모든 것을 어린 아이 음식 먹여주듯 일일이 떠 먹여 준다면, 그들은 당신을 존경하지 않을 것이다. 존경받지 못하는 해커가 무슨 의미가 있는가!

작은 것부터 시작하라. 스스로 학습할 수 있는 능력이 있음을 보여라. 그리고 나서 해커들에게 의문나는 점을 질문하라.

질문 : 그러면 어떤 식으로 시작할 수 있는가?

시작하기 가장 좋은 방법은 아마도 리눅스 사용자 그룹 모임(LUG)에 참석하는 길일 것이다. 그런 그룹은 LDP의 리눅스 일반 정보 페이지 에서 찾을 수 있다. 아마도 이곳이 우선적으로 접근할 수 있는 곳 중 하나이기도 하다. 주로 대학과 연관되어 있다. LUG 구성원들에게 질문을 하면 아마도 리눅스를 언급할 것이다. 그리고 반드시 이들은 리눅스를 설치 및 간단한 운영법에 관해서 도움을 줄 것이다.

질문 : 당신은 언제 시작했소? 배우기에는 내가 너무 늦은 나이는 아닌지?

시작하고자 하는 동기를 부여 받은 나이가 최적의 나이이다. 보통은 15세에서 20세 사이에 관심을 갖지만 예외 없는 법칙이란 없다.

질문 : 해킹하는 법을 배우려면 얼마나 걸리나?

이는 배우고자 하는 사람이 얼마나 재능이 있으며, 얼마나 열정적으로 매달리는지에 따라 다르다. 대부분의 사람들은 18개월에서 24개월 정도면 남부럽지 않은 기술을 획득할 수 있다. 물론 이 일에 전적으로 집중해야 한다는 전제 하에서 이다. 이것으로 모든 해킹 기술을 습득했다라고 여기면 큰 오산이다. 당신이 진정한 해커라면 끊임없이 학습하고 그 기교를 완벽하게 하는데 나머지 인생을 투자해야 할 것이다.

질문 : 비주얼 베이직(Visual Basic)과 델파이(Delphi)로 시작하면 괜찮겠소?

오! 천만에 말씀. 이 언어들은 이식성이 없다. 이들 언어들은 오픈 소스를 구현할 수 없기 때문에 판매자들이 지원하는 플랫폼에 갇히게 된다. 일종의 독점적 상황을 받아들인다는 것은 해커로서 의 길이 아니다.

특히 비주얼 베이직은 끔찍하다. 독식적인 마이크로소프트 언어라는 사실만으로도 결격사유가 된다. 그리고 여타 베이직 프로그램도 졸렬한 설계의 언어이기 때문에 나쁜 프로그래밍 습관을 배태시킨다.

질문 : 어떤 시스템을 크랙(crack)하려는데 도움 좀 주겠소? 아니면 크랙 하는 법좀 가르쳐 주시오?

싫소. 이 FAQ를 읽은 후에도 그런 질문을 하는 자라면, 너무 어리석어 비록 내가 시간이 많아 개인 교습을 한다 해도 불가능하다. 이러한 유형의 요구를 전자우편으로 받는다면 나는 무시해 버리거나, 극도의 무례함으로 회신할 것이다.

질문 : 어떻게 하면 다른 사람의 아이디와 패스워드를 얻어낼 수 있는가?

이건 크래커가 하는 짓이야! 꺼져버려~! 바보.

질문 : 어떻게 하면 다른 E-MAIL을 해킹할 수 있는가?

이건 크래커가 하는 짓이야. 내 앞에서 꺼져버려, 더러운 자식.

질문 : 어떻게 하면 IIC에서 OP권한을 도둑질할 수 있을까요?

이건 크래커가 하는 짓이야. 뒤져버려~! 구제 불능한 자식~!

질문 : 계속 크랙 당하는데, 이들을 강력하게 되받아 칠 수 있도록 도와줄 수 없겠소?

이것도 싫소. 지금까지 이런 질문은 수도 없이 받았다. 그런데 대개 윈도우를 운영하는 사람들이 하는 얘기다. 크래커들의 공격에 윈도우 시스템을 효율적으로 안전하게 운용한다는 것은 불가능 하다. 그 코드와 아키텍처는 너무 단순해 너무 많은 결함을 갖고 있다. 이는 체로써 배에 핀 물을 퍼내는 꼴이다. 신뢰할 수 있는 유일한 방어책은 운영체제를 리눅스로 전환하거나 진정한 보안을 탐지할 수 있는 다른 운영체제를 이용하는 길이다.

질문 : 저의 Windows프로그램이 문제 생겼어요. 도와주시겠어요?

물론이죠, DOS모드에 들어가신 다음 "format c:" 라고 쳐보세요, 모든 문제가 몇분 사이에 해결됩니다.

질문 : 어디가면 대화를 나눌 수 있는 진정한 해커를 만날 수 있을까요?

여러분 지역의 유닉스나 리눅스 사용자 그룹을 찾아보고, 그들의 회합에 참여하는 것이 가장 좋은 방법이다.[메타랩(Metalab, 이전의 sunsite) 소재의 LDP 사이트에서 유용한 사용자 그룹 목록을 찾을 수 있을 것이다].

"IRC(Internet Relay Chat)가 아닌 것만은 확실하다. 여기는 눈을 씻고 봐도 flamer들과 크래커들이 들끓는 곳이다"라고 말한 적이 있다. 그러나 상황은 변했다는 것을 알고 있다. 분명 진정한 해커 공동체(GIMP나 Perl 등에 지속적으로 어떤 의미를 부여하는)에서는 IRC 채널을 갖고 있다.

질문 : 해킹 주제와 관련해 유용한 서적을 추천해 줄 수 있는가?

나는 유용한 리눅스 관련 문건 하우투 목록 Linux Reading List HOWTO를 운영하고 있다. 그리고 Loginataka도 흥미를 가질 만 하다.

python에 관한 자료는 파이선 사이트의 초보 입문 자료를 참고하시라.

질문 : 해커가 되려면 수학을 잘 해야 하는가?

필요 없다. 해커는 수학을 사용할 기회가 매우 적다. 하지만 꼭 논리적으로 사고하고 정밀적으로 추리하는 법을 알아야 한다.

질문 : 가장 먼저 익혀야 할 언어는 무엇인가?

HTML. 이 언어를 모른다면은 말이다. 걸만 번지려 하고 과장으로 가득한 불량 HTML 서적들이 산재해 있다. 서글프게도 HTML에 대한 양서는 드물다. 내가 좋아하는 서적 중 하나는 HTML : The Definitive Guide(O'reilly 사 간)이다.

그러나 HTML은 완벽한 프로그래밍 언어는 아니다. 프로그래밍 할 만반의 준비가 되어 있다면, Python을 먼저 시작할 것을 권한다. 그러나 많은 사람들은 Perl을 추천할 것이다. Perl은 Python보다 대중적인 인기를 끌고 있다. 개인적 소견이긴 하지만 Python보다는 배우기에 다소 어렵고, 그리 잘 설계된 것은 아니다. 웹에는 Python을 이용해 프로그래밍하려는 초보자들을 위한 사이트가 있다.

C언어는 중요하다. 그러나 Perl과 Python보다는 익히기가 좀 더 어렵다. 굳이 C로 먼저 시작하려고 는 하지 말라.

윈도우 사용자라면 Visual Basic에 안주하지 마라. 이는 잘못된 프로그래밍 습관을 배태시킨다. 윈 도우 운영체제를 떠나선 생존할 수 없는 것이다. 피하라.

질문 : 어떤 하드웨어가 필요한가?

개인용 컴퓨터는 다소 성능이 떨어지고 충분한 메모리를 탑재할 수 없기 때문에 해커들의 학습 과정에서는 어떤 인위적 한계를 갖고 있다는 생각이 일반적이었다. 그러나 이는 몇 년 전에나 해 당되는 얘기이다. 486DX50이상의 머신은 X 개발 작업과 인터넷 통신, 그리고 을 하기에 충분할 정도로 강력하다. 그리고 요즘은 구입할 수 있는 가장 적은 용량의 디스크만으로도 대부분의 작 업을 하기에 충분할 정도로 크다.

학습을 하기 위한 머신을 선택할 때 중요한 기준은 그 하드웨어가 리눅스(또는 BSD 호환, 반드시 그러한 기준에 의해 선택해야 한다)를 설치하는데 적합한 것인가이다. 이는 요즘 나오는 거의 대 부분의 머신에 적용된다. 유일하게 까다로운 부분은 모뎀과 프린터이다. 어떤 모뎀과 프린터는 리눅스와는 궁합을 맞출 수 없는 윈도우에서만 작동하는 하드웨어이다. 하드웨어 호환성에 대한 FAQ의 최근 버전은 아래 기술한 사이트에서 확인하기 바란다.

질문 : 마이크로소프트를 미워하거나 비난할 필요가 있는가?

그럴 필요는 없다. 마이크로소프트가 혐오스럽지 않다는 얘기는 아니다. 해커 문화는 마이크로 소프트가 있기 오래 전부터 있어왔다. 마이크로소프트가 지금은 승승장구하고 있다 할지라

도, 해커 문화는 여전히 존재하고 그 저류를 흐르고 있다. 마이크로소프트를 미워하는데 허비할 에너지가 있다면 자신의 기교를 사랑하는데 더 많은 시간을 쏟아라. 멋진 코드를 작성하라. 그러면 그대의 업보를 훼손하지 않고도 마이크로소프트를 충분히 한 대 매길 수 있을 것이다.

질문 : 소스 공개 소프트웨어를 작성하여 나누어 준다면, 프로그래머들은 어떻게 생활을 꾸려 나갈 수 있는가?

지금까지는 그런 것 같지 않다. 소스 공개 소프트웨어 산업은 프로그래머들을 내몰기보다는 일 (job)을 창출하는 것 같다. 작성된 프로그램을 갖고 있다는 것은 전혀 작성하지 않았을 때 보다는 순경제적 이득이라고 한다면, 프로그래머들은 그 프로그램이 완성된 후에 그것이 무료가 되었든 그렇지 않던 간에 보상을 받게 될 것이다. 그리고 아무리 “무료”로 소프트웨어를 작성한다 하더라도, 언제나 새롭고 주문적인 응용프로그램에 대한 좀 더 많은 수요가 창출된다고 볼 수 있다. 나는 오픈소스 페이지에 본인의 생각을 구체적으로 개진했다.

질문 : 어떻게 시작할 수 있는가? 어디서 공개 유닉스를 구할 수 있는가?

어디서든 구할 수 있다. 해커가 되기 위해서 여러분은 자신을 교육할 수 있는 동기, 창의력과 능력이 필요하다.