

実践を通じて学ぶ

今日は「Learn By Doing」について話しましょう。このことは非常に重要です。学習は大切です。そして、どのように学ぶかを学ぶことの重要性は言うまでもありません。私は、何かを作り出すためには、実践しながら学び、学びながら実践することが最善だと気づきました。例えば、Android 開発を学びたいなら、実際に Android 開発を実践し、挑戦的なプロジェクトを一つずつこなしていくことが必要です。

Paul Graham は、コンピュータサイエンスの大学生がどのように学生生活を過ごすべきかについて述べています。それは、大量の困難な問題を解決することです。まず、最初から最後まで困難な問題を解決する過程で、多くのことを学ぶことができます。例えば、非常に挑戦的なプロジェクトに取り組むこと、WeChat クライアントを開発することなどです。もしすべて自分で行うのであれば、最終的には Photoshop を使って画像を処理できるようになり、Android のあらゆる側面を理解するべきです。もしサードパーティのプッシュサービスを使用しないのであれば、自分でリアルタイム通信サーバーを書くことで、さらに多くのことを学ぶことができるでしょう。

本を一冊読み終えると、大体理解できたと感じことがあります。しかし、それだけでは十分ではありません。『疯狂 Android 讲义』を読み終えたとき、私は Android プログラミングについてかなり理解したと思いました。本の内容は大体把握できたと感じました。約 10 日間で数百ページの本を読み終えました。その後、4 ヶ月かけてその知識を習得しました。いくつかの小さなプロジェクトと、少し大きなプロジェクトをこなして、ようやく Android 開発に慣れてきたと感じました。

つまり、本を読んで得た知識だけでは十分ではありません。本を読むことは表面的なものです。もちろん、私が 10 日間だけ本を読んだのであれば、それほど時間をかけていないとも言えます。思いついたんだけど、高校教育、中学教育、小学校教育のほとんどの時間は無駄にされているのかもしれない。これらはすべて本に基づいた教育だからだ。

前回、伯父と石灰水がどうやって壁に固定されるかについて話し合いました。実は今でもよく分かっていません。CaO と O₂ が酸化して CaCO₃ になるのか、それとも CaCO₃ が固体で比較的硬いからなのか。日常生活では、Cao と CaCO₃ を区別するのが難しいです。練習や問題を解くことで、彼らが何に変換できるか、どんな特性があるかを熟知できるようになりましたが、すぐにまた忘れてしまいます。大学入試からもう 1 年半が経ち、自発的に思い出せる知識はもうあまりありません。私は大学入試で 590 点を取りましたが、今すぐもう一度受けるとしたら、300 点くらいでしょうか。

幸いなことに、私は高校の 3 年間、多くの時間をアルゴリズムやプログラミングの学習に費やしました。すべての時間を退屈な教科書や試験といった紙の上のゲームに費やしたわけではありませんでした。しかし、残念なことに、当時はもっとプログラミングを学んでいませんでした。

確かに、物理、化学、生物、国語、数学、英語を学ぶ過程で、私は鍛えられ、何度も完璧を追求することで、忍耐力や粘り強さを身につけました。しかし、その弊害も明らかです。それは、学習が苦痛であり、知識が退屈で、全てが順位や高得点のためだと思わせてしまうことです。それはさらに、生まれつきの興味や好奇心に影響を与え、プログラミングの道においても、順位や自分がxxより優れているかどうかに気を取られ、知識そのものの楽しさを味わうことができなくなってしまいます。

TJ (vision media) は、自分は学校に行ったことがなく、本も読まないとっています。彼はプログラミングを学ぶために、他人のコードを見て、常にその理由を知りたがり、好奇心を持って学ぶのだそうです。

Paul Graham は、野心的な大人とは、好奇心を一つの小さな領域に集中させる人だと言っています。

知識が複雑になればなるほど、名声や利益、自制心に頼って習得することは難しくなり、自発的な好奇心に依存するしかありません。

私は小さい頃から知識を習得するのは、名声や利益、順位、他人よりも優れるためでした。私は外部の認可をとても必要としています。もちろん、プログラミング自体も比較的好きです。

私の原動力となっているのは、いくつかの友人と比べて、私はまだ未熟だということです。彼らはとても優秀で、私と同じ年齢、大学2年生で20歳ですが、一人は趣味でコンパイラを書いており、普段は外注会社でとても良い仕事をしています。もう一人の友人は2つの会社を持ち、フルタイムの仕事でバックエンド開発を請け負っています。また、知り合いの清華大学の学生たちは、さまざまな世界大会で賞を受賞しています。

明らかに、自分でも多くの差を感じています。彼らが私よりも早く学び始めたとかいうことではなく、彼らの効率が高く、しかも毎日私よりも多くの時間を費やしているのです。

同世代のプレッシャーに基づいて、同世代の人々よりも優れたいという動機は非常に大きいと言わざるを得ません。同僚にも非常に優秀な人たちがいます。しかし、無意識のうちに彼らは私より年上だと思い、私が彼らの年齢になったら彼らと同じくらいになれるかもしれませんとれます。そう考えると、特に気にすることはありません。

ですから、私たちはできるだけ自分と同じ年齢層で、自分よりも優れた人々と知り合いになるように努めます。それは自分にとって残酷なことかもしれません。常に自分が未熟であることを認めなければならぬからです。しかし、その結果は良いものになります。ただ、あなたが強く全てに立ち向かい、自暴自棄にならず、差を見つけて静かに努力すれば、結果は非常に良いものになるでしょう。

さらに多くの利点があるかもしれません。彼らは優れていて、特定のスキルに熟練しているため、そのことを本当に好きな人である可能性が高いです。プログラミングを愛している人であ

る可能性が高いです。そのため、自分自身にポジティブな影響を与え、プログラミングの楽しさを感じることができるでしょう。

プログラミングの良いところは、一度没頭すると、自然とそれが好きになり、順位や自分がどれだけ優れているか、名声や利益といったものを忘れてしまうことです。

私は認めます、私がプログラミングに時間を費やすのは、一つにはそれが私の仕事であり、上司に何かを報告しなければならないからです。二つ目は、自分がまだまだ未熟で、もっと努力しないと遅れてしまうと感じているからです。三つ目は、プログラミングの達人になってフルスタックで起業するという夢を持っているからです。しかし、私が時間を費やしたことで、手を動かして何かを創造する楽しさをすぐに感じることができました。一つまた一つと問題を解決していく達成感を味わうことができたのです。

手を動かすことで、知識をより良く学ぶことができ、またより楽しくもなります。