eForth and Zen

Dr. C. H. Ting
Offete Enterprises
1993

eForth and Zen

Contents

1. Forth and Zen

Forth is often mentioned not only as a computer language but also a religion because of its fervorish followers. Among religions Zen is considered to be the closest to Forth. As popularly known Zen is understood as a synonym of simplicity, brevity, light, understanding, wisdom and enlightenment. These are also attributes to Forth as a language and a philosophy. Indeed, Zen and Forth also bear striking similarity in their historical development and evolution. This aspect of similarity between Forth and Zen has not been well documented, and this is one purpose of this book.

Zen and Forth both started as revolutions towards well entrenched establishments--the priesthood. Zen and Forth both started as oral traditions because of the lack in supporting

literature and the high concentration of expertise. Zen and Forth both stressed that their matters of subject were simple and straightforward. However, it was in the interests of the entrenched establishment to make things complicated and inaccessible to common folks. Zen and Forth both stressed that the true enlightenment and understanding were within the grasp of individuals. The rituals and the accepted practices exercised in the establishments had nothing to do with these goals.

Here I would like to compare Zen and Forth in greater details so that we all have a better appreciation of these two seemingly unrelated topics. Forth programmers may be encouraged to press on using this language with a higher hope that their work will be recognized by their peers in the future.

Comparing Forth and Zen

The most important contributions of Huineng, the Sixth Patriarch of Zen in China, was that he had his lectures recorded by his disciples. He also had the lectures printed and distributed as identifications of discipleship to his teach and his philosophy. In striking contrast to the Buddhist Sutras translated from Sanskrit, the lectures were plain, easy to read and to comprehend. This collection of lectures was the only book written by a Chinese to be granted the status of a Sutra in the Buddhist literature, and is commonly call the Platform Sutra. (Sutras were the teaching of Buddha himself.)

People compared Forth and Zen in a very superficial manner, mostly without the understanding of either. I feel that there is a need to treat both subjects fully in a single treatise. There is no better way than to present the very original text of Zen and the source code of Forth side by side. In the course of preparing the complete documentation for eForth, I thought it would be very useful to lay down the eForth text in parallel with the original text of the Platform Sutra. Of course, it is impossible to correlate the text of eForth directly with the text of the Platform Sutra. However, one should be able to see the common threads in them, as the themes of simplicity, personal understanding, enlightenment, and the struggles against prevailing established doctrines recurring time and again in both texts.

As I intended this book for both English and Chinese readers, I would like to include texts and discussions in both English and Chinese. In effect, we would have four threads running in parallel: the original Platform Sutra in Chinese, its English translation, the text of eForth, and its Chinese translation. This is how this book is laid out.

Historically, Zen was a grand synthesis, combining the essence of Buddhism, Confucianism, and Taoism after about one thousand years of inter fertilization. It was the results of the many Chinese minds, struggling for emancipation after a thousand years of conflicts between the traditional humanistic Confucianism, the nihilistic Taoism and the imported anti-materialistic Buddhism. It laid dormant for a hundred years, surviving through five generations of Zen masters, passing the doctrine orally from heart to heart. Eventually, during the reign of the Sixth Patriarch, Huineng (638-714 AD), it blossomed in

full and became the dominating religious philosophy in China ever since. Its influence spread into Japan, Korea, and Southeast Asia. Lately, it also became fashionable in Western Europe and America.

The history of Forth is too short for meaningful comparison with that of Zen. It was virtually unknown to the world in its first decade of existence until the late 70's. It was invented by a lone programmer, Charles H. Moore, outside of the main stream of computer industry and computer sciences. In the early 70's, it was only used in astronomy, as he helped programming minicomputers to automate the telescopes and the observatories. Then it blossomed with the microcomputer revolution, promoted by the Forth Interest Group in the early 1980's. Since 1985, the Forth Interest Group has been in steady decline, as C became the dominant programming language. The advantages of Forth, such as the elegance in its architecture, the simplicity in its syntactic construction, and the economy in the memory utilization, seem irrelevant in the age of mega-resources, where MIPS, megabytes of RAM memory, gigabytes of disk storage are commodities easily affordable.

As the operating systems and applications grow to fill the available RAM and disk storage, at some point people will ask the question whether these huge programs are worthy of the resources they consume. People will have to ask whether the direction we are heading will lead us to better lives and better environment. When we stop equating bigger to better, and more to happier, then we can re-evaluate the computer technology in a new light. Then, maybe Forth will shine again.

Zen as an Oral Tradition

Buddhism was founded by Gautama Siddhartha, a religious philosopher and teacher who lived in India (~560-480 B.C.) He was called Buddha which means the enlightened one. He attracted a large following. His teaching in essence was that one could reach Nirvana, a divine state of release from earthly pain, sorrow, and desire, by the right living, the right thinking and self denial. He left no writing behind, while Buddhism flourished in India for a long time.

In the two hundred years after Buddha's death, many schools formed after different personalities and there were great arguments and debates concerning what were his true teaching. Great conventions were held to debate the issues and codified his teachings as Sutras in Sanskrit.

The earlier dominant school was Hinayana, which spread to the south and is still flourishing in Sri Lanka, Burma, and Thailand. The later dominant school was Mahayana, which spread into China, Tibet, Korea, and Japan.

The Hinayana School emphasize the mystic power of Buddha and the personal salvation through one's own efforts. The Mahayana School emphasized eclecticism and in common search for salvation.

After the introduction of Mahayana into China in 200 AD, it arose great interests in the intellectuals as well as the peasants. Many emperors and their courts were converted and spent great efforts in building temples and spreading the Buddhism. A continuing effort over three hundred years was devoted to translate SanskritSutra and related literature into Chinese. In Tang Dynasty (618-907 AD), more than 5000 volumes of Buddhist literature were translated and assembled.

Most of the Sutra translations were done poorly and required a priesthood for the interpretation and dissimulation. The vast amount of literature also caused sectarian divisions and arguments among the priesthood, continuing the Hindu tradition.

Zen was introduced into China by a legendary Indian monk Buddhidharma in 527 AD. He stayed at the Shaolin Temple for 9 years, spending all his time meditating in front of a stone wall. He was known as the 'Indian Monk Looking at a Wall'. He didn't use any Sutra, and he didn't write anything. He taught a few students and encouraged them to find enlightenment in themselves. His teaching was summarized as:

> *My teaching is outside Buddhist tradition,*
> *As truth cannot be conveyed by writings;*
> *Cleanse your mind to reveal your true nature;*
> *One can reach Nirvana directly.*

Buddhidharma passed his garment and bowl tohis student Huiko as evidence of the discipleship and commanded him to do the same for five generations. He then returned to India. Huiko passed the teach with the garment and the bowl to Sengtsan. Sengtsan passed them to Taohsin. Taohsin passed them to Hungjen. Finally, Hungjen passed them to Huineng (638-714 AD).

For a hundred years, Zen was passed from mouth to mouth, and from heart to heart. Very few people knew of its existence. Even fewer knew its philosophy and teachings. In China, Buddhism flourished when supported by the emperors and by high officers. A number of times the Buddhism was almost completely destroyed when the country was in turmoil and when the Confucian officers could convince the emperor that Buddhists were threats to the state. All the while, the Zen masters orally passed their teachings from one generation to the next.

Forth as an Oral Tradition

Forth was invented by Charles (Chuck) H. Moore who was trained as a physicist in MIT but wandered into programming. In early days, he built an interpreter so that it would execute words on punched cards. Later he found that these words could be more conveniently compiled into lists, which could be executed by the computer more conveniently. The interpreter with very small modification, could be made to compile anything and everything, and the whole scheme evolved into a programming language. It was named Forth, as abbreviated from Fourth, meaning the fourth generation of

programming language when the third generation of computers bases on integrated circuits were becoming prevailing in the computing industry.

Very early in the development of Forth, a state of closure was reached. Chuck was able to generate new Forth systems from an existing Forth system through meta-compilation. He did not need other programming tools to build new Forth systems, and Forth started to evolve independent of the existing operating systems and programming languages. This state of closure was very interesting, like the ying-yang cycle. One could not find an entry point once the cycle was closed. In that Chuck had the monopoly on Forth, because very few people possessed the understanding to cut in the cycle in order to build new Forth systems. He felt quite secure in giving users the complete source listings, fairly sure of that nobody could reversed engineered the Forth technology, even though the source listings were complete and truthful.

Indeed, the source code was very difficult to read, because a Forth system was generated by the meta-compiler, and the meta-compiler was written in Forth. To understand Forth, one had to understand the meta-compiler. To understand the meta-compiler, one had to understand Forth completely. Where do you start?

Forth thus became a legend. The astronomers loved it so much that they made it the standard language for observatory automation. It was fairly easy to use but very difficult to understand. The source code traveled to the far corners of the world with the telescopes, but the knowledge and understanding of Forth was only passed from mouth to mouth and heart to heart. Hence Forth became an oral tradition these days. Forth code tended to be concise and often packed tightly in blocks. In-line documentation and comments were deemed too expensive, and most code was poorly commented. Forth thus acquired the reputation of a write-only language.

Several manuals were circulated among the observatories, documenting a few of the most popular Forth implementations. These manuals mostly contained a short section introducing Forth and discussing how to use that particular Forth system, and a long dictionary documenting what each word did. These manuals told the users what Forth was, but provided very little help as to how Forth worked.

Acceptance of Zen

Huineng, the Sixth Patriarch, was a genius. He couldn't read because he was borne poor and gathered wood for a living, but he could explain the Sutras when people read them to him. He went to learn from the Fifth Patriarch Hungjen, and Hungjensent him to labor in the kitchen. As Hungjen got old and wanted to pass on the garment and the bowl, he asked his students to write poems to show him their understanding of the enlightenment. His best student Shenhsiu wrote the following poem:

> *Our body is the bodhi tree,*
> *And our mind a mirror bright.*

Carefully we wipe them hour by hour,
And let no dust alight.

Hearing this poem, Huineng asked a scholar to write down his own poem, because he couldn't write himself:

There is no bodhi tree,
Nor stand of a mirror bright.
Since all is void,
Where can the dust alight?

When Hungjen saw this poem, he passed the garment and the bowl to Huineng and told Huineng: "You are the one Buddhidharma prophesied. Zen will flourish in China through you. Take the garment and the bowl to be the Sixth Patriarch, but do not pass them on any more. '" Hungjen was in such a hurry to pass things to Huineng that he didn't even shave Huineng's hair (to admit him to Buddhist order), as Huineng was still a layman.

About 20 years later, when Huineng was well established as the Master of Zen, he was asked by a Provincial Officer to give lectures on Zen. The Officer had Huineng's eldest student Fahai recorded his lectures and had the lectures printed as the 'Platform Sutra, Lectures by the Six Patriarch'. When Huineng was about to die, Fahai asked him: "What are you going to do with the garment and the bowl? Who's going the inherit them?" Huineng said: "As commanded by Buddhidharma, the garment and the bowl will not be passed on. But now you have the Platform Sutra. Go forth and teach others according to this Sutra. Everything I learned I put down in it. When you read it, it is as if you am talking to me."

20 years after Huineng died, the Northern School was favored by the royal court and dominated the Buddhist landscape. The Southern School was scattered and mute in Southern China. One of Huineng's student, Shenhui (686-760 AD), went to the capital and challenged the doctrine and the Zen inheritance of the Northern School in a series of lectures and public debates. He convinced the court and the public of the historical significance of Huineng and established the Platform Sutra as the orthodox doctrine of Zen Buddhism.

Acceptance of Forth

The major breakthrough in the Forth arena was due to the Forth Interest Group, which was founded by Bill Ragsdale in 1978 in the Silicon Valley. The most important contribution FIG made was to reverse-engineer a Forth system from ground zero, thus breaking the infinite ying-yang cycle. FIG organized a Forth Implementation Team which built and released 6 Forth implementations for the 6 then most popular microprocessors based on the figForth model. These implementations were written in assembly code of the native microprocessors. People who were familiar with the assembly code could then easily

implement figForth on their own microcomputers. figForth thus trained a new generation of Forth programmers outside the Forth oral tradition.

There were a host of Forth literature appearing in the early 1980's, which further helped the popularization of Forth among the personal computer users. Among them was Leo Brodie's 'Starting Forth' , 'Thinking Forth', and the 1979 Special Forth Language Issue in the Byte magazine. 'Forth Dimensions' from FIG and 'Journal of Forth Applications and Research' from the Forth Institute were the two major publications on Forth. These literature showed that Forth penetrated into many different scientific communities and technical industries.

My most important contribution to Forth was the publication of 'Systems Guide to figForth', first released in 1979. Instead of telling people what Forth did, it systematically explored how Forth did things and why things were done the ways they were. It put to rest the myth that Forth was a write-only language by showing that Forth could be understood by the average user with some casual study. It showed how the inner interpreter and the outer interpreter worked, and why words and dictionary in Forth wereconstructed the ways they were by necessity. It proved that the understanding of Forth could be transmitted through the paper medium without personal interaction. The impact of this work, I would like to believe myself, was similar to what Huineng caused with his Platform Sutra on Zen.

Simplicity in Zen.

Buddhism is very complicated because it is not a monolithic system of thoughts and philosophy. It accumulated many centuries of cultural and philosophical development. The Sutras were all attributed to Buddha but were most likely written by people remotely associated with Buddha. Lots of the mystic Hinduism found their ways into Buddhism, which was inevitable because Buddhism was developed in the Hindu environment, like the 33 layers of heaven, 18 layer in hell, the reincarnation of all animals, etc.

There were many different theories about how life, death, and Nirvana. There were many sectors and schools about how one could attain Nirvana to avoid the reincarnation into a lower animal form. Thing got complicated and confusion reigned supreme. In essence, everybody just picked what he believed and convinced others that his was the best and most logical way to deal with life and all its ramifications.

The general consensus was that vegetarianism was good, giving to the temple was good, kindness to people and animal was good, reciting Sutra was good, meditation was good, worshipping Buddha and other Buddhist deities was good, dedicating to priesthood was good, etc. Could one attain Nirvana by doing all these? Maybe. Maybe not.

Zen was a great simplification of all these. Huineng maintained that Buddha hood and enlightenment could not be achieved through generally accepted Buddhist practices, like reciting Sutras, making offerings, meditation in special sitting positions. As everybody

already had the Buddha nature in him, all he had to do is looking inward to find the true Buddha. Our senses and our thoughts tended to veil us from the Buddha nature and they should not be trusted. The process of Zen (Dhyana, Ch'an, meditation), was to reject the influences of senses and thoughts, and to arrive at a state of ideallessness, nonobjectivity, and nonattachment. In this state, nothing external of ourselves and within our own minds can influence us and dragged us back to the earthy existence.

Simplicity in Forth

The poem by Shenhsiu and the poem by Huineng provide the best contrast for us to compare the conventional wisdom in the current computer industry against the Forth philosophy. Let me paraphrase the poems to show my point. From the mainstream of the computer science, one will advice our youngster:

Hardware is complicated,
Software even more so.
Study hard day and night,
Maybe you'll find a way to go.

From the point of view of Forth, we might say:

Hardware is the reality,
Software but an illusion.
Learn your Forth well,
And beat both into submission.

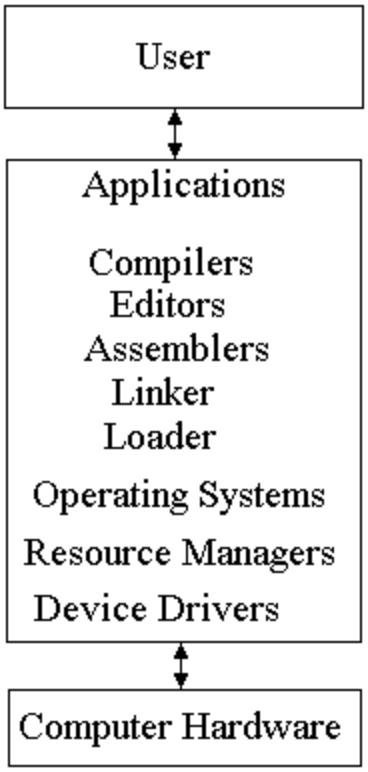
Computer hardware is difficult. Computer software is even more difficult. We have volumes and volumes of literature to prove them. Hardware and software are difficult, only because people are not given the right tools to deal with the complexity in these systems. If we insist on asking whether the complexity is necessary, we can convince ourselves that they should not be complicated. The computer hardware evolved trying to solved the perceived software problems. The software evolved trying to solve the perceived hardware problems and the problems in the human interface. If these preconceived problems do not exist at all, the hardware can be simple and powerful. The software can be simple and powerful as well.

In a typical computer system, there are layers upon layers of software between the user and the computer hardware. The operating system and its utilities, the compilers, assemblers, editors, linkers and loaders are all very complicated and mostly proprietary programs. They helped the user to start his journey into this computer juggle. After a while, they tend to hinder the users progress, because they insulate the user from the hardware, and deliberate efforts were expended to prevent the user to fully make use of the capabilities built in the hardware.

This issue of simplicity can be illustrated in the following diagram. The operating systems

and the applications separate the user and the computer hardware. The software protects the hardware, because the user is not to be trusted. Leaving to the user, he will abuse the hardware and causes the system to crash. In the days of mainframe computers, the greatest sin was to crash the computer, because the livelihood of the computer priesthood depended upon the continuing operation of the computer. The hardware and the operating system had to be protected at all costs.

In this age of personal computers, the priority is turned by 180 degrees. The user owns the computer. He is responsible for its operation. Is it necessary to protect the computer against its owner?



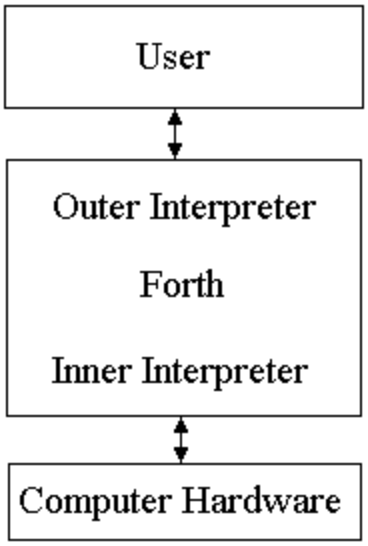
Conventional Computer System

Forth provides a much simplified interface between the user and his computer as show in the following diagram. Forth is a simple and integrated interface between the user and his computer.

Through Forth, the use can directly control the computer hardware, because every part of the computer system is freely accessible to the user. Therefore, the user can explore the best way to use the computer system to suit his applications.

With the freedom to access computer hardware comes the responsibility to use the

hardware properly. The user may crash the system frequently. It will cause no harm as long as the computer can recover quickly from the crashes with minimal damages to the data stored in the computer system. After all, the user owns the computer.



A Simple Forth System

There are other examples which shows that complexity does not necessarily mean performance. Simple systems are generally faster and more resilient.

Registers in the CPU are designed to hold temporary data so that the CPU does not have to go to the slow memory to fetch and store data. However, large number of registers become a burden when you call subroutines, and the registers must be saved before the call and restored after the return. In order to speed up a computer, you tend to have more and more registers and make less and less subroutine calls. If we recognize that in high level languages and in structure programming the subroutine is the most important mechanism, we should instead optimize the memory access in the subroutine call-return and put all the intermediate data on the data stack.

Another example is the prefix arithmetical notation prevailing in conventional programming languages. The prefix notation is unnatural and was forced upon all young minds in their algebra lessons. It is much more natural for both computers and humans to think in linear lists, sequentially executed. Thinking algebraically, you need two-pass compilers to break the equations down into pieces and reassemble them for the computer to execute. Thinking in the postfix terms, a simple one-pass interpreter can be drafted to perform all the functions required of a complicated compiler.

Enlightenment in Zen

Zen was the cumulative synthesis of the Buddhist philosophy and the traditional Chinese Confucianism and Taoism. Zen was also a revolution against the Buddhist traditions and establishments. It discredited the Buddhist practices, which emphasized ceremonies and outer appearances, while claimed that the enlightenment exists only in the minds of individuals.

A huge amount of Buddhist literature had been translated from Sanskrit to Chinese. Because the translations were difficult to understand, a priesthood was established for its dissimulation and interpretation. Towards this literal tradition, the Zen masters proclaimed that enlightenment could not be transmitted by written words, but had to be handed done orally from heart to heart.

In traditional Buddhist theories, it was very difficult to attain Nirvana or Buddhahood. It required a long time of studying, and the practice of self-denial. In the end, there was still no assurance that one could attain it. Even if one attained it in this life, there would be the possibility of losing it in the next life. There were external forces which we could not know and we could not avoid.

Zen placed the possibility and the capability to attain enlightenment and Buddhahood squarely in the individual, by declaring that the Buddha nature is part of the human nature and it exists in everybody. The Buddha nature is vile and corrupted by the worldly desires and thoughts. These desires and thoughts can be purged, the individual can thus be enlightened, and his own Buddha nature can reveal itself. The enlightenment is the realization of this self-sufficient Buddha nature.

As to how one became enlightened, there were two major schools of thoughts. Huineng insisted that enlightenment came suddenly and Senghsui maintained that it ought to be the results of diligent study, mediation and searching. These were the Southern Sudden School and the Northern Gradual School. However, as Huineng maintained, how enlightenment is achieved is not important. The important thing is its realization. People are all different, and they are enlightened in different ways. A master can teach, but he can not enlighten. The enlightenment comes from within. The best a master can do is to inspire, to help, to lead the way, and maybe to strike a sharp blow on the head at the right time.

Enlightenment in Forth

What is the enlightenment in Forth? I think it is the complete understanding of a computer in terms of its operations and its interface to the user. This understanding is not as complicated as we have all being lead to believe. It involves two components of Forth: the inner interpreter which Forth imposes on the computer hardware to execute Forth lists, and the outer interpreter which executes words typed in by the user. If one understands both the inner interpreter and the outer interpreter, he has the complete understanding of computer, in the sense that he can go out and build a Forth system on any computer and make that computer do what he wants it to do. He then will realize that operating systems and languages enslave people to do what the system permits them to do, while Forth gives

them freedom to tell the computer to do what they want to do. This is enlightenment.

Operating systems and programming languages are designed to enslave the users, by their sheer sizes and their complexities. They are too complicated to be understood by individual users. Forth shows that an operating system and a high level language do not have to be complicated. In fact, they can be very simple and can be mastered by individuals with some limited efforts. Once the principles are mastered, they can be applied to all computers. Then, the user can become the master, and the computers become obedient but powerful slaves. To be the master of powerful slaves is very enlightening and satisfying. Once you taste the freedom and the satisfaction of being the master, you will not want to be enslaved again by a computer through its operating systems and programming languages.

(note at the time of publication, 1993, Offete also published the Platform Sutra, and had 25 different eForth disks available.)

(the entire text including chapters 2-8 is available in printed form from Offete Enterprise Inc.)

9. Some Final Thoughts.

Congratulations if you reach this point the first time. As you can see, we have traversed a complete Forth system from the beginning to the end, and it is not as difficult as you might have thought before you began. But, think again what we have accomplished. It is a complete operating system with an integrated interpreter and an integrated compiler all together. If you look in the memory, the whole sytem is less than **7 K**bytes. What else can you do with **7 K**bytes these days?

Forth is like Zen. It is simple, it is accessible, and it can be understood in its entirety without devoting your whole life to it.

Is this the end? Not really. There are many topics important in Forth that we chose to ignore in this simple model. They include multitasking, virtual memory, interrupt control, programming style, source code management, and yes, metacompilation. However these topics can be considered advanced applications of Forth. Once the fundamental principles in Forth are understood, these topics can be subject for further investigations at your leisure.

Forth is not an end in itself. It is only a tool, as useful as the user intends it to be. The most important thing is how the user can use it to solve his problems and build useful applications. What eForth gives you is the understanding of this tool. It is up to you to make use of it.

Ultra Technology
2512 10th Street
Berkeley, CA 94710