

# The Unix Koans of Master Foo

 Home

## Master Foo and the Ten Thousand Lines

Master Foo once said to a visiting programmer: “There is more Unix-nature in one line of shell script than there is in ten thousand lines of C.”

The programmer, who was very proud of his mastery of C, said: “How can this be? C is the language in which the very kernel of Unix is implemented!”

Master Foo replied: “That is so. Nevertheless, there is more Unix-nature in one line of shell script than there is in ten thousand lines of C.”

The programmer grew distressed. “But through the C language we experience the enlightenment of the Patriarch Ritchie! We become as one with the operating system and the machine, reaping matchless performance!”

Master Foo replied: “All that you say is true. But there is still more Unix-nature in one line of shell script than there is in ten thousand lines of C.”

The programmer scoffed at Master Foo and rose to depart. But Master Foo nodded to his student Nubi, who wrote a line of shell script on a nearby whiteboard, and said: “Master programmer, consider this pipeline. Implemented in pure C, would it not span ten thousand lines?”

The programmer muttered through his beard, contemplating what Nubi had written. Finally he agreed that it was so.

“And how many hours would you require to implement and debug that C program?” asked Nubi.

“Many,” admitted the visiting programmer. “But only a fool would spend the time to do that when so many more worthy tasks await him.”

“And who better understands the Unix-nature?” Master Foo asked. “Is it he who writes the ten thousand lines, or he who, perceiving the emptiness of the task, gains merit by not coding?”

Upon hearing this, the programmer was enlightened.

## Master Foo and the Script Kiddie

A stranger from the land of Woot came to Master Foo as he was eating the morning meal with his students.

“I hear y00 are very l33t,” he said. “Pl33z teach m3 all y00 know.”

Master Foo's students looked at each other, confused by the stranger's barbarous language. Master Foo just smiled and replied: “You wish to learn the Way of Unix?”

“I want to b3 a wizard hax0r,” the stranger replied, “and 0wn ever3one's b0xen.”

“I do not teach that Way,” replied Master Foo.

The stranger grew agitated. “D00d, y00 r nothing but a p0ser,” he said. “If y00 n00 anything, y00 wud t33ch m3.”

“There is a path,” said Master Foo, “that might bring you to wisdom.” The master scribbled an IP address on a piece of paper. “Cracking this box should pose you little difficulty, as its guardians are incompetent. Return and tell me what you find.”

The stranger bowed and left. Master Foo finished his meal.

Days passed, then months. The stranger was forgotten.

Years later, the stranger from the land of Woot returned.

“Damn you!” he said, “I cracked that box, and it was easy like you said. But I got busted by the FBI and thrown in jail.”

“Good,” said Master Foo. “You are ready for the next lesson.” He scribbled an IP address on another piece of paper and handed it to the stranger.

“Are you *crazy*?” the stranger yelled. “After what I’ve been through, I’m never going to break into a computer again!”

Master Foo smiled. “Here,” he said, “is the beginning of wisdom.”

On hearing this, the stranger was enlightened.

## **Master Foo Discourses on the Two Paths**

Master Foo instructed his students:

“There is a line of dharma teaching, exemplified by the Patriarch McIlroy’s mantra ‘Do one thing well’, which emphasizes that software partakes of the Unix way when it has simple and consistent behavior, with properties that can be readily modeled by the mind of the user and used by other programs.”

“But there is another line of dharma teaching, exemplified by the Patriarch Thompson’s great mantra ‘When in doubt, use brute force’, and various sutras on the value of getting 90% of cases right *now*, rather than 100% *later*, which emphasizes robustness and simplicity of implementation.”

“Now tell me: which programs have the Unix nature?”

After a silence, Nubi observed:

“Master, these teachings may conflict.”

“A simple implementation is likely to lack logic for edge cases, such as resource exhaustion, or failure to close a race window, or a timeout during an uncompleted transaction.”

“When such edge cases occur, the behavior of the software will become irregular and difficult. Surely this is not the Way of Unix?”

Master Foo nodded in agreement.

“On the other hand, it is well known that fancy algorithms are brittle. Further, each attempt to cover an edge case tends to interact with both the program’s central algorithms and the code covering other edge cases.”

“Thus, attempts to cover all edge cases in advance, guaranteeing ‘simplicity of description’, may in fact produce code that is overcomplicated and brittle or which, plagued by bugs, never ships at all. Surely this is not the Way of Unix?”

Master Foo nodded in agreement.

“What, then, is the proper dharma path?” asked Nubi.

The master spoke:

“When the eagle flies, does it forget that its feet have touched the ground? When the tiger lands upon its prey, does it forget its moment in the air? Three pounds of VAX!”

On hearing this, Nubi was enlightened.

## **Master Foo and the Methodologist**

When Master Foo and his student Nubi journeyed among the sacred sites, it was the Master's custom in the evenings to offer public instruction to Unix neophytes of the towns and villages in which they stopped for the night.

On one such occasion, a methodologist was among those who gathered to listen.

“If you do not repeatedly profile your code for hot spots while tuning, you will be like a fisherman who casts his net in an empty lake,” said Master Foo.

“Is it not, then, also true,” said the methodology consultant, “that if you do not continually measure your productivity while managing resources, you will be like a fisherman who casts his net in an empty lake?”

“I once came upon a fisherman who just at that moment let his net fall in the lake on which his boat was floating,” said Master Foo. “He scabbled around in the bottom of his boat for quite a while looking for it.”

“But,” said the methodologist, “if he had dropped his net in the lake, why was he looking in the boat?”

“Because he could not swim,” replied Master Foo.

Upon hearing this, the methodologist was enlightened.

## **Master Foo Discourses on the Graphical User Interface**

One evening, Master Foo and Nubi attended a gathering of programmers who had met to learn from each other. One of the programmers asked Nubi to what school he and his master belonged. Upon being told they were followers of the Great Way of Unix, the programmer grew scornful.

“The command-line tools of Unix are crude and backward,” he scoffed. “Modern, properly designed operating systems do everything through a graphical user interface.”

Master Foo said nothing, but pointed at the moon. A nearby dog began to bark at the master's hand.

“I don't understand you!” said the programmer.

Master Foo remained silent, and pointed at an image of the Buddha. Then he pointed at a window.

“What are you trying to tell me?” asked the programmer.

Master Foo pointed at the programmer's head. Then he pointed at a rock.

“Why can't you make yourself clear?” demanded the programmer.

Master Foo frowned thoughtfully, tapped the programmer twice on the nose, and dropped him in a nearby trashcan.

As the programmer was attempting to extricate himself from the garbage, the dog wandered over and piddled on him.

At that moment, the programmer achieved enlightenment.

## **Master Foo and the Old Hand**

An experienced Unix programmer, hearing of Master Foo's wisdom, came to him for guidance. Approaching the Master, he bowed three times and said:

“Master Foo, I am gravely troubled. In my youth, those who followed the Great Way of Unix used software that was simple and unaffected, like `ed` and `mailx`. Today, they use `vim` and `mutt`. Tomorrow I fear they will use `KMail` and `Evolution`, and Unix will have become like Windows — bloated and covered over with GUIs.”

Master Foo said: “But what software do you use when you want to draw a poster?”

The programmer replied: “I...have never done that. But I am sure that I could use `LaTeX` or `pic` to accomplish it without GUIs, in the proper Unix way.”

Master Foo then said: “Which one will reach the other side of the river: The one who dreams of a raft, or the one that hitchhikes to the next bridge?”

Upon hearing this, the programmer was enlightened.

## **Master Foo and the Shell Tools**

A Unix novice came to Master Foo and said: “I am confused. Is it not the Unix way that every program should concentrate on one thing and do it well?”

Master Foo nodded.

The novice continued: “Isn't it also the Unix way that the wheel should not be reinvented?”

Master Foo nodded again.

“Why, then, are there several tools with similar capabilities in text processing: `sed`, `awk` and `Perl`? With which one can I best practice the Unix way?”

Master Foo asked the novice: “If you have a text file, what tool would you use to produce a copy with a few words in it replaced by strings of your choosing?”

The novice frowned and said: “`Perl`'s regexps would be excessive for so simple a task. I do not know `awk`, and I have been writing `sed` scripts in the last few weeks. As I have some experience with `sed`, at the moment I would prefer it. But if the job only needed to be done once rather than repeatedly, a text editor would suffice.”

Master Foo nodded and replied: “When you are hungry, eat; when you are thirsty, drink; when you are tired, sleep.”

Upon hearing this, the novice was enlightened.

## **Master Foo and the Nervous Novice**

There was a novice who learned much at the Master's feet, but felt something to be missing. After meditating on his doubts for some time, he found the courage to approach Master Foo about his problem.

“Master Foo,” he asked “why do Unix users not employ antivirus programs? And defragmentors? And malware cleaners?”

Master Foo smiled, and said “When your house is well constructed, there is no need to add pillars to keep the roof in place.”

The novice replied “Would it not be better to use these things anyway, just to be certain?”

Master Foo reached for a nearby ball of string, and began wrapping it around the novice's feet.

“What are you doing?” the novice asked in surprise.

Master Foo replied simply: “Tying your shoes.”

Upon hearing this, the novice was enlightened.

## **Master Foo and the Recruiter**

A technical recruiter, having discovered that that the ways of Unix hackers were strange to him, sought an audience with Master Foo to learn more about the Way. Master Foo met the recruiter in the HR offices of a large firm.

The recruiter said, “I have observed that Unix hackers scowl or become annoyed when I ask them how many years of experience they have in a new programming language. Why is this so?”

Master Foo stood, and began to pace across the office floor. The recruiter was puzzled, and asked “What are you doing?”

“I am learning to walk,” replied Master Foo.

“I saw you walk through that door” the recruiter exclaimed, “and you are not stumbling over your own feet. Obviously you already know how to walk.”

“Yes, but this floor is new to me.” replied Master Foo.

Upon hearing this, the recruiter was enlightened.

## **Master Foo Discourses on Returning to Windows**

A student said: “We have learned that Unix is not just an operating system, but also a style of approaching problems.”

Master Foo nodded in agreement.

The student continued: “Then, the Great Way of Unix can be applied on other operating systems?”

Master Foo sat silent for a moment, then said: “In every operating system there is a path to the Great Way, if only we can find it.”

The student continued: “What, then, of Windows? It is preinstalled on most computers, and though its tools are mostly far inferior, they are easy to use for beginners. Surely, Windows users could benefit from the Unix philosophy.”

Master Foo nodded again.

The student said: “How, then, are those enlightened in the Unix Way to return to the Windows world?”

Master Foo said: “To return to Windows, you have but to boot it up.”

The student said, growing agitated: “Master Foo, if it is so easy, why are there so many monolithic and broken software packages for Windows? Elegant software should also be possible with a GUI and fancy colors, but there is little evidence that this occurs. What becomes of an enlightened one who returns to Windows?”

Master Foo: “A broken mirror never reflects again; fallen flowers never go back to the old branches.”

Upon hearing this, all present were enlightened.

## **Master Foo and the Unix Zealot**

A Unix zealot, having heard that Master Foo was wise in the Great Way, came to him for instruction. Master Foo said to him:

“When the Patriarch Thompson invented Unix, he did not understand it. Then he gained in understanding, and no longer invented it.”

“When the Patriarch McIlroy invented the pipe, he knew that it would transform software, but did not know that it would transform mind.”

“When the Patriarch Ritchie invented C, he condemned programmers to a thousand hells of buffer overruns, heap corruption, and stale-pointer bugs.”

“Truly, the Patriarchs were blind and foolish!”

The zealot was greatly angered by the Master's words.

“These enlightened ones,” he protested, “gave us the Great Way of Unix. Surely, if we mock them we will lose merit and be reborn as beasts or MCSEs.”

“Is your code ever completely without stain and flaw?” demanded Master Foo.

“No,” admitted the zealot, “no man's is.”

“The wisdom of the Patriarchs” said Master Foo, “was that they *knew* they were fools.”

Upon hearing this, the zealot was enlightened.

## **Master Foo Discourses on the Unix-Nature**

A student said to Master Foo: “We are told that the firm called Novell holds true dominion over Unix.”

Master Foo nodded.

The student continued, “Yet we are also told that the firm called OpenGroup also holds true dominion over Unix.”

Master Foo nodded.

“How can this be?” asked the student.

Master Foo replied:

“Novell indeed has dominion over the code of Unix, but the code of Unix is not Unix. OpenGroup indeed has dominion over the name of Unix, but the name of Unix is not Unix.”

“What, then, is the Unix-nature?” asked the student.

Master Foo replied:

“Not code. Not name. Not mind. Not things. Always changing, yet never changing.”

“The Unix-nature is simple and empty. Because it is simple and empty, it is more powerful than a typhoon.”

“Moving in accordance with the law of nature, it unfolds inexorably in the minds of programmers, assimilating designs to its own nature. All software that would compete with it must become like to it; empty, empty, profoundly empty, perfectly void, hail!”

Upon hearing this, the student was enlightened.

## **Master Foo and the MCSE**

Once, a famous Windows system administrator came to Master Foo and asked him for instruction: “I have heard that you are a powerful Unix wizard. Let us trade secrets, that we may both gain thereby.”

Master Foo said: “It is good that you seek wisdom. But in the Way of Unix, there are no secrets.”

The administrator looked puzzled at this. “But it is said that you are a great Unix guru who knows all the innermost mysteries. As do I in Windows; I am an MCSE, and I have many other certifications of knowledge not common in the world. I know even the most obscure registry entries by heart. I can tell you everything about the Windows API, yes, even secrets those of Redmond have half-forgotten. What is the arcane lore that gives you your power?”

Master Foo said: “I have none. Nothing is hidden, nothing is revealed.”

Growing angry, the administrator said “Very well, if you hold no secrets, then tell me: what do I have to know to become as powerful in the Unix way as you?”

Master Foo said: “A man who mistakes secrets for knowledge is like a man who, seeking light, hugs a candle so closely that he smothers it and burns his hand.”

Upon hearing this, the administrator was enlightened.

## **Master Foo and the End User**

On another occasion when Master Foo gave public instruction, an end user, having heard tales of the Master's wisdom, came to him for guidance.

He bowed three times to Master Foo. “I wish to learn the Great Way of Unix,” he said “but the command line confuses me.”

Some of the onlooking neophytes began to mock the end user, calling him “clueless” and saying that the Way of Unix is only for those of discipline and intelligence.

The Master held up a hand for silence, and called the most obstreperous of the neophytes who had mocked forward, to where he and the end user sat.

“Tell me,” he asked the neophyte, “of the code you have written and the works of design you have uttered.”

The neophyte began to stammer out a reply, but fell silent.

Master Foo turned to the end-user. “Tell me,” he inquired, “why do you seek the Way?”

“I am discontent with the software I see around me,” the end user replied. “It neither performs reliably nor pleases the eye and hand. Having heard that the Unix way, though difficult, is superior, I seek to cast aside all snares and delusions.”

“And what do you do in the world,” asked Master Foo, “that you must strive with software?”

“I am a builder,” the end user replied, “Many of the houses of this town were made under my chop.”

Master Foo turned back to the neophyte. “The housecat may mock the tiger,” said the master, “but doing so will not make his purr into a roar.”

Upon hearing this, the neophyte was enlightened.

## **Master Foo and the Programming Prodigy**

There was a time when rumors began to reach Master Foo and his students of a prodigiously gifted programmer, a young man who wandered the length and breadth of the land performing mighty feats of coding and humiliating all who dared set their skill against his.

Eventually this prodigy came to visit Master Foo, who received him politely and offered him tea. The Prodigy accepted with equal politeness and explained the motive for his visit.

“I have come to you,” he said “seeking a code and design review of my latest project. For it is of surpassing complexity, and I do not have peers capable of understanding it. Only an acknowledged master such as yourself (and here the Prodigy bowed deeply) can have the discernment required.”

Master Foo bowed politely in return and began examining the Prodigy's code. After some time he raised his eyes from the screen. “This code is at first sight very impressive,” he said. “It is elegant in design, utilizing original algorithms of great ingenuity, and appears to be implemented in a craftsmanlike way which minimizes the possibility of errors.”

The Prodigy looked very pleased at this praise, but Master Foo continued: “However, I detect one significant flaw.”

“Flaw?” the Prodigy said. “What flaw?”

“This code is difficult to read,” said Master Foo. “It is only thinly commented, its invariants are not specified, and I see no narrative description of its architecture or internal data structures anywhere. These problems will seriously impede your cooperation with other programmers.”

The Prodigy drew himself up haughtily. “I do not seek the cooperation of other programmers,” he said. “Every time I thought I had found one who might match me in skill I have been disappointed. Thus, I work alone.”

“But even the hacker who works alone,” said Master Foo, “collaborates with others, and must constantly communicate clearly to them, lest his work become confused and lost.”

“Of what others do you speak?” the Prodigy demanded.

Master Foo said: “All your future selves.”

Upon hearing this, the Prodigy was enlightened.

## **Master Foo and the Hardware Designer**



On one occasion, as Master Foo was traveling to a conference with a few of his senior disciples, he was accosted by a hardware designer.

The hardware designer said: “It is rumored that you are a great programmer. How many lines of code do you write per year?”

Master Foo replied with a question: “How many square inches of silicon do you lay out per year?”

“Why...we hardware designers never measure our work in that way,” the man said.

“And why not?” Master Foo inquired.

“If we did so,” the hardware designer replied, “we would be tempted to design chips so large that they cannot be fabricated - and, if they were fabricated, their overwhelming complexity would make it be impossible to generate proper test vectors for them.”

Master Foo smiled, and bowed to the hardware designer.

In that moment, the hardware designer achieved enlightenment.