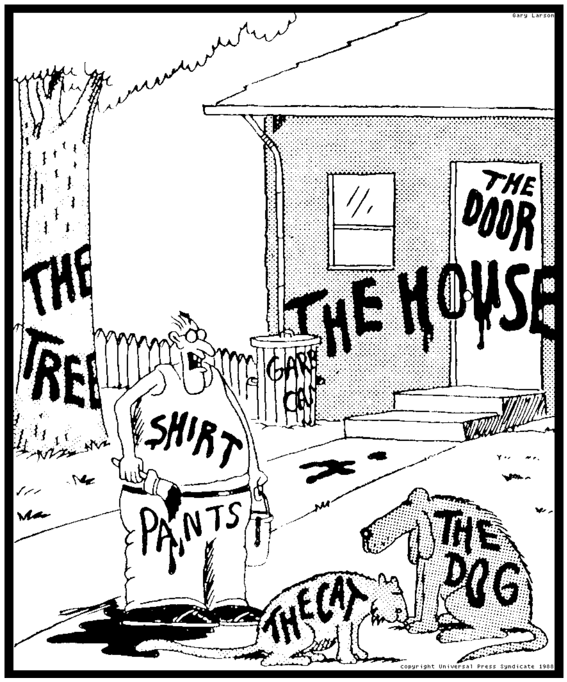
Choosing a Name for your Computer

Don Libes

Integrated Systems Group
National Institute of Standards and Technology
Gaithersburg, MD 20899



“Now! *That* should clear up a few things around here!”

ABSTRACT

In order to easily distinguish between multiple computers, we give them names. Experience has taught us that it is as easy to choose bad names as it is to choose good ones. This essay presents guidelines for deciding what makes a name good or bad.

Keywords: domain name system, naming conventions, computer administration, computer network management

Reprint of *Communications of the ACM*, Vol. 37, No. 11, pp. 1289-1291, November 1989. Also publishised as *RFC 1178*, August 1989. Text is contribution of the National Institute of Standards and Technology, and not subject to copyright. Gary Larson’s cartoon (above) copyright owned by United Tribute Syndicate.

June 30, 1989

Introduction

As soon as you deal with more than one computer, you need to distinguish between them. For example, to tell your system administrator that your computer is busted, you might say, "*Hey Ken. Goon is down!*"

Computers also have to be able to distinguish between themselves. Thus, when sending mail to a colleague at another computer, you might use the command "*mail libes@goon*".

In both cases, "*goon*" refers to a particular computer. How the name is actually dereferenced by a human or computer need not concern us here. This essay is only concerned with choosing a *good* name. (It is assumed that the reader has a basic understanding of the domain name system as described by [Mockapetris].)

By picking a good name for your computer, you can avoid a number of problems that people stumble over again and again.

Here are some guidelines on what *not* to do.

Don't overload other terms already in common use.

Using a word that has strong semantic implications in the current context will cause confusion. This is especially true in conversation where punctuation is not obvious and grammar is often incorrect.

For example, a distributed database had been built on top of several computers. Each one had a different name. One machine was named "*up*", as it was the only one that accepted updates. Conversations would sound like this: "*Is up down?*" and "*Boot the machine up.*" followed by "*Which machine?*"

While it didn't take long to catch on and get used to this zaniness, it was annoying when occasionally your mind would stumble, and you would have to stop and think about each word in a sentence. It is as if, all of a sudden, English has become a foreign language.

Don't choose a name after a project unique to that machine.

A manufacturing project had named a machine "*shop*" since it was going to be used to control a number of machines on a shop floor. A while later, a new machine was acquired to help with some of the processing. Needless to say, it couldn't be called "*shop*" as well. Indeed, both machines ended up performing more specific tasks, allowing more precision in naming. A year later, five new machines were installed and the original one was moved to an unrelated project. It is simply impossible to choose generic names that remain appropriate for very long.

Of course, they could have called the second one "*shop2*" and so on. But then one is really only distinguishing machines by their number. You might as well just call them "*1*", "2", and "*3*". The only time this kind of naming scheme is appropriate is when you have a lot of machines and there are no reasons for any human to distinguish between them. For example, a master computer might be controlling an array of one hundred computers. In this case, it makes sense to refer to them with the array indices.

While computers aren't quite analogous to people, their names are. Nobody expects to learn much about a person by their name. Just because a person is named "*Don*" doesn't mean he is the ruler of the world (despite what the "*Choosing a Name for your Baby*" books say). In reality, names are just arbitrary tags. You cannot tell what a person does for a living, what their hobbies are, and so on.

Don't use your own name.

Even if a computer is sitting on your desktop, it is a mistake to name it after yourself. This is another case of overloading, in which statements become ambiguous. Does "*give the disk drive to don*" refer to a person or computer?

Even using your initials (or some other moniker) is unsatisfactory. What happens if I get a different machine after a year? Someone else gets stuck with "*don*" and I end up living with "*jim*". The machines can be renamed, but that is excess work and besides, a program that used a special peripheral or database on "*don*" would start failing when it wasn't found on the *new* "*don*".

It is especially tempting to name your first computer after yourself, but think about it. Do you name any of your other possessions after yourself? No. Your dog has its own name, as do your children. If you are one of those who feel so inclined to name your car and other objects, you certainly don't reuse your own name. Otherwise you would have a great deal of trouble distinguishing between them in speech.

For the same reason, it follows that naming your computer the same thing as your car or another possession is a mistake.

Don't use long names.

This is hard to quantify, but experience has shown that names longer than eight characters simply annoy people.

Most systems will allow prespecified abbreviations, but why not choose a name that you don't have to abbreviate to begin with? This removes any chance of confusion.

Avoid alternate spellings.

Once we called a machine "*czek*". In discussion, people continually thought we were talking about a machine called "*check*". Indeed, "*czek*" isn't even a word (although "*Czech*" is).

Purposely incorrect (but cute) spellings also tend to annoy a large subset of people. Also, people who have learned English as a second language often question their own knowledge upon seeing a word that they know but spelled differently. ("*I guess I've always been spelling "funxion" incorrectly. How embarrassing!*")

By now you may be saying to yourself, "*This is all very silly...people who have to know how to spell a name will learn it and that's that.*" While it is true that some people will learn the spelling, it will eventually cause problems somewhere.

For example, one day a machine named "*pythagoris*" (sic) went awry and began sending a tremendous number of messages to the site administrator's computer. The administrator, who wasn't a very good speller to begin with, had never seen this machine before (someone else had set it up and named it), but he had to deal with it since it was clogging up the network as well as bogging down his own machine which was logging all the errors. Needless to say, he had to look it up every time he needed to spell "*pythagoris*". (He suspected there was an abbreviation, but he would have had to log into yet another computer (the local nameserver) to find out and the network was too jammed to waste time doing that.)

Avoid domain names.

For technical reasons, domain names should be avoided. In particular, name resolution of non-absolute hostnames is problematic. Resolvers will check names against domains before checking them against hostnames. But we have seen instances of mailers that refuse to treat single-token names as domains. For example, assume that you mail to "*libes@rutgers*" from *yale.edu*. Depending upon the implementation, the mail may go to *rutgers.edu* or *rutgers.yale.edu* (assuming both exist).

Avoid domain-like names.

Domain names are either organizational (e.g., *cia.gov*) or geographical (e.g., *dallas.tx.us*). Using anything like these tends to imply some connection. For example, the name "*tahiti*" sounds like it means you are located there. This is confusing if it is really somewhere else (e.g., "*tahiti.cia.gov is located in Langley, Virginia? I thought it was the CIA's Tahiti office!*"). If it really is located there, the name implies that it is the only computer there. If this isn't wrong now, it inevitably will be.

There are some organizational and geographical names that work fine. These are exactly the ones that do not function well as domain names. For example, amorphous names such as rivers, mythological places and other impossibilities are very suitable. ("*earth*" is not yet a domain name.)

Don't use antagonistic or otherwise embarrassing names.

Words like "*moron*" or "*twit*" are good names if no one else is going to see them. But if you ever give someone a demo on your machine, you may find that they are distracted by seeing a nasty word on your screen. (Maybe their spouse called them that this morning.) Why bother taking the chance that they will be turned off by something completely irrelevant to your demo.

Don’t use digits at the beginning of the name.

Many programs accept a numerical internet address as well as a name. Unfortunately, some programs do not correctly distinguish between the two and may be fooled, for example, by a string beginning with a decimal digit.

Names consisting entirely of hexadecimal digits, such as "*beef*", are also problematic, since they can be interpreted entirely as hexadecimal numbers as well as alphabetic strings.

Don’t use non-alphanumeric characters in a name.

Your own computer may handle punctuation or control characters in a name, but most others do not. If you ever expect to connect your computer to a heterogeneous network, you can count on a variety of interpretations of non-alphanumeric characters in names. Network conventions on this are surprisingly nonstandard.

Don’t expect case to be preserved.

Upper and lowercase characters look the same to a great deal of internet software, often under the assumption that it is doing you a favor. It may seem appropriate to capitalize a name the same way you might do it in English, but convention dictates that computer names appear all lowercase. (And it saves holding down the shift key.)

Now that we’ve heard what not to do, here are some suggestions on names that work well.

Use words/names that are rarely used.

While a word like "*typical*" or "*up*" (see above) isn’t computer jargon, it is just too likely to arise in discussion and throw off one’s concentration while determining the correct referent. Instead, use words like "*lurch*" or "*squire*" which are unlikely to cause any confusion.

You might feel it is safe to use the name "*jose*" just because no one is named that in your group, but you will have a problem if you should happen to hire Jose. A name like "*sphinx*" will be less likely to conflict with new hires.

Use theme names.

Naming groups of machines in a common way is very popular, and enhances communality while displaying depth of knowledge as well as imagination. A simple example is to use colors, such as "*red*" and "*blue*". Personality can be injected by choices such as "*aqua*" and "*crimson*".

Certain sets are finite, such as the seven dwarfs. When you order your first seven computers, keep in mind that you will probably get more next year. Colors will never run out.

Some more suggestions are: mythical places (e.g. *Midgard*, *Styx*, *Paradise*), mythical people (e.g. *Procne*, *Tereus*, *Zeus*), killers (e.g., *Cain*, *Burr*, *Boleyn*), babies (e.g. *colt*, *puppy*, *tadpole*, *elver*), collectives (e.g. *passel*, *plague*, *bevy*, *covey*), elements (e.g. *helium*, *argon*, *zinc*), flowers (e.g. *tulip, peony*, *lilac*, *arbutus*). Get the idea?

Use real words.

Random strings are inappropriate for the same reason that they are so useful for passwords. They are hard to remember. Use real words.

Don't worry about reusing someone else's hostname.

Extremely well-known hostnames such as "*sri-nic*" and "*uunet*" should be avoided since they are understood in conversation as absolute addresses even without a domain. In all other cases, the local domain is assumed to qualify single-part hostnames. This is similar to the way phone numbers are qualified by an area code when dialed from another area.

In other words, if you have choosen a reasonable name, you do not have to worry that it has already been used in another domain. The number of hosts in a bottom-level domain is small, so it shouldn't be hard to pick a name unique only to that domain.

There is always room for an exception.

I don't think any explanation is needed here. However, let me add that if you later decide to change a name (to something sensible like you should have chosen in the first place), you are going to be amazed at the amount of pain awaiting you. No matter how easy the manuals suggest it is to change a name, you will find that lots of obscure software has rapidly accumulated which refers to that computer using that now-ugly name. It all has to be found and changed. People mailing to you from other sites have to be told. And you will have to remember that names on old backup media labels correspond to different names.

I could go on but it would be easier just to forget this guideline exists.

Conclusion

Most people don't have the opportunity to name more than one or two computers, while site administrators name large numbers of them. By choosing a name wisely, both user and administrator will have an easier time of remembering, discussing and typing the names of their computers.

I have tried to formalize useful guidelines for naming computers, along with plenty of examples to make my points obvious. Having been both a user and site administrator, many of these anecdotes come from real experiences which I have no desire to relive. Hopefully, you will avoid all of the pitfalls I have discussed by choosing your computer's name wisely.

Credits

Thanks to the following people for suggesting some of these guidelinesand participating in numerous discussions on computer naming: Ed Barkmeyer, Peter Brown, Chuck Hedrick, Ken Manheimer, and Scott Paisley.

References

Mockapetris, P., "*Domain Names - Concepts and Facilities*", RFC 1034, Information Sciences Institute, CA, November 1987.

Author Bio

Don Libes is a computer scientist at the National Institute of Standards and Technology. He is currently working on distributed data system research for automated manufacturing. In this heterogeneous environment, he has been exposed to computers named in all sorts of inexplicable ways, accompanied by the usual grief that follows. Don has also recently completed his first book, "Life With UNIX", published by Prentice Hall.