

I am not clueless  
or  
Myths and misconceptions about the design of  
GoboLinux

Hisham H. Muhammad

13th June 2004

*"Those who do not understand Unix  
are doomed to reinvent it, poorly."  
– Henry Spencer, 1987*

This week we had another release of GoboLinux, and again a number of people, even if indirectly, called me "clueless" for coming up with such a structure for a Linux distribution, for a number of reasons. None of those reasons was new; I heard all of them many times.





on Linux, further isolates the legacy tree as an isolated accessory.

## Do you want to change the standard?

Of course not. For starters, we're not that naive to think that we could. But the actual reason why we don't want to change the standard is because we believe *there should be no standard*. I know this statement may sound even bolder than talking about changing a standard, but the reason I say that is because we believe it is the duty of each application to allow itself to be installed anywhere and to accept that other applications it needs to work with may be installed anywhere (more on this in the next section). Now, if there was a standard stating this, I'd even sign a petition to support it. In fact, there is: the GNU release standards, when they recommend the usage of GNU Autotools, supporting the `--prefix` family of switches, and probing for the location of applications with the `configure` script, do just that. But when a proposed standard like the FHS gives me an arbitrary list of binaries that should be, for unexplained reasons, in a separate directory, I laugh at that.

Different situations imply different needs, and so-called standards that attempt to fit every feet in the same shoe are doomed to failure. Standardize on flexibility instead. That's not we don't propose the GoboLinux tree as a standard to be followed by anyone else. In five, ten or twenty years, we may have completely different needs from the ones we have today. I don't want that the move away from the GoboLinux tree then to be as hard as the move away from the traditional Unix tree is today. Which leads us naturally to our next section...

## An uphill bet to change applications

is not as hard to get the standard of GoboLinux as it is to get the standard of Linux. I have to install every extra app I need in my \$HOME directory. this is a perfectly

community. Let's return to the `$HOME` directory scenario. What if my favorite GNOME com-



- make all programs use a unified configuration file format – Of course, the rewrite effort that this would require is even greater than the one from the previous item. Every idea that starts with “make all programs...” tends to have the same problems: the effort is potentially infinite; you can’t make every project in the world agree to use your that can’t

First of all, there is the issue of the dependency systems. GoboLinux has a very loi6oJJ-167.936-13.54



crease the granularity of the security model, do dilute the power of `root`: ACLs, capabilities, SELinux... It may be argued that some of those add excessive complexity to the model, but I won't dive into this discussion here. The one thing that is clear is that the `root` model