

Re: What Linux can learn from Windows...

Source: <http://linux.derkeiler.com/Newsgroups/alt.os.linux/2003-09/1453.html>

From: Walter Mautner (*new.10.eatallspam_at_spamgourmet.com*)

Date: 09/17/03

Date: Wed, 17 Sep 2003 07:56:14 +0200

Tommy Li wrote:

...

> *There is a lack of a unifed place to store program settings and
> information. Currently, settings are scattered about the system in
> text files, XML, or in other cryptic formats. This makes configuring
> a system somewhat of a pain. You must memorize the names and
> locations of files, and what they configure. On top of that, there
> are many different formats used for configuration, from XML, to the
> ALSA asoundrc format, to god knows what.*

>

All configurations that are intended to be editable by users, still
are so – provided you are familiar with vi or emacs :).

> *Windows (back in 95, I think) used to have something like this,
> where information was stored in INI files. I would have to say that
> the Windows registry was great (mostly, since some of the
> applications still didn't follow the registry standards). All the
> configuration everywhere could be edited with one editor, and it was
> easy to find where your programs kept their settings (assuming they
> followed the standards).*

>

Did you ever really look inside the windoze registry, searching for
how to modify a distinct setting, and then were you able to deal with
that successfully without first having to google around for half an
hour?

– there are no comments in the registry, as are common in linux
configuration files

– inside the registry, information regarding one program, is still
strayed across a lot of hkey entries

– old and obsolete entries remain after updates and even uninstalls,
bloating the registry and slowing down the computer without a real
chance to clean up by hand in a reasonable amount of time. Registry
cleaners (even mickisofts regclean) often also remove/alter vital
entries

– a single corrupted entry might break the whole OS

– vital device assignments are stored therein, making it a tough job
to alter physical drive locations

... and many more big suckers.

> *Perhaps we could adopt something like this. People could write their*

First I had to look at the headers, if you weren't a wintroll. Forgive

...

> *own programs to edit the registry, and it would be much easier to*
> *edit configurations like now. Instead of "/etc/fstab", people would*
> *just use their favorite registry editor, and navigate to "Local*
> *System -> Mount Points" or whatever. Backwards compatibility could*
> *probably be maintained for the hardcore old-timers: programs should*
> *be able still configured from virtual files editable by text*
> *editors.*

>

Well, we have the editable (even from a rescue CD) files. Now, have you tried to regedit from a windoze2K recovery console started from CD?

> *The other thing that bothers me is the fact that there isn't really*
> *a set place to install programs (as we discussed thoroughly in that*
> *last old topic). Windows partially solved that problem with the*
> *Program Files folder. In Linux, there are many different standards*
> *for this with many programs not following any of them. As a result,*
> *chaos ensues.*

>

http://www.felixgers.de/teaching/unix/linux_tree.html

You have to follow the standards. Of course, some distros and their rpms handle things differently. but for self-compiled sources you can always add a --prefix to put it where you think it should be.

However, the different hierarchy folders (/bin, /sbin, /usr/bin, /usr/sbin, /usr/local ...) make sense, as well as the libs paths.

Now compared to windoze, at least until the most recent versions, you already had the choice between your own (program-supplied) libraries and the common system libraries. With windoze, you ended in dll hell often enough, with equal dll names.

Also, it's up to the windoze installers where to put the programs.

Some still refuse to use the programs folder. Some make a "Program Files" folder in a german "Programme" windoze and you end up with 2 programs folders. Some ancient program installers still insist in "C:" and crash if your windows is elsewhere.

> *I think we should do something about this with Linux. From a*
> *conceptual*
> *viewpoint, there are to approaches to organization of applications:*
> *grouping by application, or grouping by application component type.*
> *Windows does the first (generally), and keeps everything in it's own*
> *folder under "Program Files". Linux does the latter, and scatters*
> *the program files in "/usr/local/bin/", "/usr/local/lib/",*
> *"/usr/local/doc/", etc.*

>

Well, windoze also groups by type dlls go into the system directories, together with and often overwriting the builtins. Now, since 2K, we have that "dll cache" folder ... used to limit damage and waste a lot of disk space, after all.

> *Both approaches have their pros and cons. On the one hand, you can
> access everything related to a program from a single directory. On*

Not true. You can not uninstall a windoze program by deleting that directory. Well, it won't work anymore, but cause troubles instead. Leftover hooks and libs (dlls), autostart entries (registry and elsewhere), links you have to clean by hand, stale registry entries and filetype links ...

You still have to use the (hopefully supplied/available) uninstaller which will be called from control panel/uninstall software.

> *the other hand, all the documentation, libraries, and binaries will
> be in one place. I believe, with clever use of symbolic links, we
> can get the advantages of both.*

>

Now, to continue with my last (uninstaller) issue, in Linux you also have the "make uninstall" for source packages, or the rpm -e commands. Inside the os, you will find a lot of symlinks already in place. But then, when you delete a directory containing a symlink, it just deletes the link entry, not (fortunately in most cases) the target.

> *Makefiles should be able to be set with a flag to install to /opt
> and create symbolic links from, "/usr/doc/appname/" to
> "/opt/appname/doc".*

>

You can use the --prefix and many other flags from the .configure scripts, and later on put the symlinks into /usr/local/bin if necessary. Or just create a link on your gui desktop.

> *Another option is to have Makefiles install in the normal linux
> convention, create a directory of "/opt/appname/", and make symbolic
> links within "/opt/appname/" to "/usr/local/bin/appname/",
> "/usr/local/doc/appname/", and other files related to the
> application. This way, all the application files could be accessed
> (and deleted) from one place.*

>

You misunderstand the meaning of links.

<http://www.felixgers.de/teaching/unix/inode.html>

Still, there are "make uninstall" scripts as long as you keep the sources (but that's the same with windows programs which require the install media to uninstall).

> *What do you think, guys?*

>

alt.os.linux: Re: What Linux can learn from Windows...

Well, there always are changes and different views, as well as different interpretations of conventions and standards, but since unix (hey, watch hpux, solaris, aix and all the different brands) has been on the carpet, linux (again all flavors) appears to have the most consistent one. Thanks to the distributors and the rpm system, which makes a good reason for various distributors to standardize the file locations in order to make common usable rpm packages.