Subject: Re: Final release
Posted by mdelfede on Sat, 26 Jul 2008 19:07:16 GMT
View Forum Message <> Reply to Message

unodgs wrote on Thu, 24 July 2008 23:30
I never loved the linux because of that. Why the system can't look like this:
/kernel
/xorg
/gnome
/kde
/users -> /me /wife...
/apps
All user applications are stored in apps directory. Every app has it's own directory and it keeps there all the files it needs there. That all /opt /usr/bin /usr/lib /bin /share /etc are crap very messy and unintuitive.
With such structure there could be one app installer for every distro.

With an 'uniform' file structure, I agree 100%. The only real linux problem is incompatibility with distros about file structure and used libs.
The former would be easy solveable, but I lost hope on it since years. Ubuntu is making a standard, and for me is ok.
The latter problem is more difficult, but can be solved installing needed libs as app dependency.

Your 'proposed' file structure... well, that's a bit more problematic. You say 'every app has it's own directory'.
Just think at 'path' problem and you'll see that's not manageable, OR you loose the ability to run apps from command line. Second problem... where do you put config files ? in app folder... impossible, unix doesn't have a *reliable* way to locate executable location. So it must be in a 'fixed' location, which could be /apps/myapp/ or, as it is by now, /etc/myapp. Just a naming difference.....
But the real, big problem are share libraries, as usual. Just look at windows, they name it "dll hell" because of something.
Do you want a truly 'shared' dll ? so it must be located on a fixed location, which is by now /usr/lib. Being so, if your app needs an upgraded shared lib, you must install it along the one already present --> dll hell problem.
You want to call it 'shared' but be sure that each app calls 'right' lib ? so put it in a by-app location, you solve dll hell problem but you can have 20 identical shared libs loaded in memory by 20 different apps.... I guess that's done, more or less, with .net apps.

IMHO, one best way to manage a big part of all problems is in .deb packages. RPM's had (IMHO.... please no flames here !)much more problems. They state *clearly* which libs an app needs, the location where download it, if some conflicts with others. And no need to ship a package burded with tons of dlls inside, which it's a not so small advantage.

Conclusion :

-windows way : easy, installers bundled with tons of dlls, accessory files and apps, don't detect easy dlls conflicts, they usually leave tons of files/registry entries when uninstalled.

IMHO one of the worst way to package stuffs.
You have sometimes problems when switching to a new os version.

-rpms way : more complicated installer, no dlls, but you must (IIRC...) manually get dependent rpms. Or use advanced tools like URPMI or so. With advanced tools, not so bad.

-debs way : small installers, they take care of dependency solving, they signals conflicts, they usually can detect if a package is not anymore needed and purge it. IMHO, one of the best ways to deploy stuffs.

with windows way, if you upgrade os you MUST reinstall all apps. No other reliable way... people that tells they just upgraded windows keeping old setups just have 2-3 apps installed, and usually also those have problems.

with ubuntu way, just click a button and wait some 3 ours to have your brand new system. Yep, sometimes with small problems, but easy solved (up to now, I went from 7.04 to 7.10 to 8.04 with no problem at all).

Ciao

Max