

SFS Technology V1.1.0.2

v1.1.0.2

Now sfslauncher can export Gnome/KDE menu items (.desktop files) in a specified directory.

Inserted SFSsettings, the settings program of the SFS technology.

Solved the daemon's thread distruction error.

v 1.1.0.1

Improved the multithreading handle.

The new sfs technology solves software dependencies and installation problems. It consists of a unique file that can be mounted as a normal file-system with all libraries and files needed by the specific program. When mounted the sfslauncher will start the application stored into it. The procedure is completely transparent to the end user; Just double click on the "program.sfs" and sfslauncher will do the magic. Starting with version 1.0 we improved the integration: the operating system now knows the "sfs" file type as a native executable file.

How does it work?

Well, the file "program.sfs" is a squashfs compressed file that can be used as a virtual filesystem. The .sfs file can be mounted by a daemon in an hidden directory (/mounted/"softwarename"). Once mounted the file, sfslauncher program will read the executable's name and all the parameters stored into the "parameters" file (a simple text file). Then the sfslauncher will read the description.id file and will execute the executable file. When the user will close the executable(the software into the package) sfslauncher call the daemon and send to him a command called "flush": the daemon will release the thread, unmount the .sfs file and delete the hidden directory.

What is NOT contained in the sfs file ?

The Desktop environment's libraries and other operating system's native libraries as for example KDE libraries, gtk+2, glib, libc6 and many more.

What will contain the .sfs file ?

The sfs file contains just the most necessary libraries to perform a correct software execution.

These are what we think are the highlight of the SFS Technology:

1--Ease of use;

2--Ease of software sharing;

3--The base configuration of the operating system remains unaltered;

- 4--It allows dynamic library linking;
- 5--It allows to use different version of a software;
- 6--The sfs technology preserve the environment: it work in USER MODE;
- 7--Operates also in 64 Bit environment;
- 8--Released under the GNU/GPL license;
- 9--A version for Linux Mobile (ARM processors) is under develop;

SFS Technology Version 1.0 beta

The version 1.0 beta of SFS Technology provide a Daemon that mount the sfs packages and a software sfslauncher (that works in user mode) that launch and handle the application contained into the package. All this is done in a completely transparent way! The sfstechnology is designed for be compiled on x86,AMD64/EM64T(64Bit like platforms),ARM architectures.

Details

Into the package you will find:

- The sfslauncher sources (Free Pascal code).
- The sfslauncher i386 binary file.
- The daemon sources (Free Pascal code).
- The daemon i386 binary file.

How to use this beta version

To use it you must:

- 1) **copy** the file called **MAKEloop** into the *"/usr/sbin/sfs"* directory,
- 2)then you must run this command:

sudo daemon

- 3)from the user mode console (don't use sudo or switch user) type:

sfslauncher "put here a file sfs"

You can use the SFS version of Google Earth5 from here:

<http://www.unixteam.net/index.php>
[mod=none_Fdplus&fdaction=download&url=sections/Download/Sfs_Applications_Directory_Download/Internet/GoogleEarth5.sfs](http://www.unixteam.net/index.php?mod=Download/Sfs_Applications_Directory_Download/Internet/GoogleEarth5.sfs)

Or you can download a sfs file from here:

http://www.unixteam.net/index.php?mod=Download/Sfs_Applications_Directory_Download

Make sure to have a kernel that support the squashfs module! Otherwise it doesn't work

Use Lazarus with FPC version 2.2.0 for x86 x64 arm processors.

The important sources contained in this package are:

--sfslauncher.lpr
--daemon/daemon.lpr
--SFSError/sfserror.lpi
--SFSinit/sfs.lpr

Contacts :

see

<http://code.google.com/p/sfs-technology/>

vincenzo@unixteam.net

or

vincenzodentamaro@hotmail.com