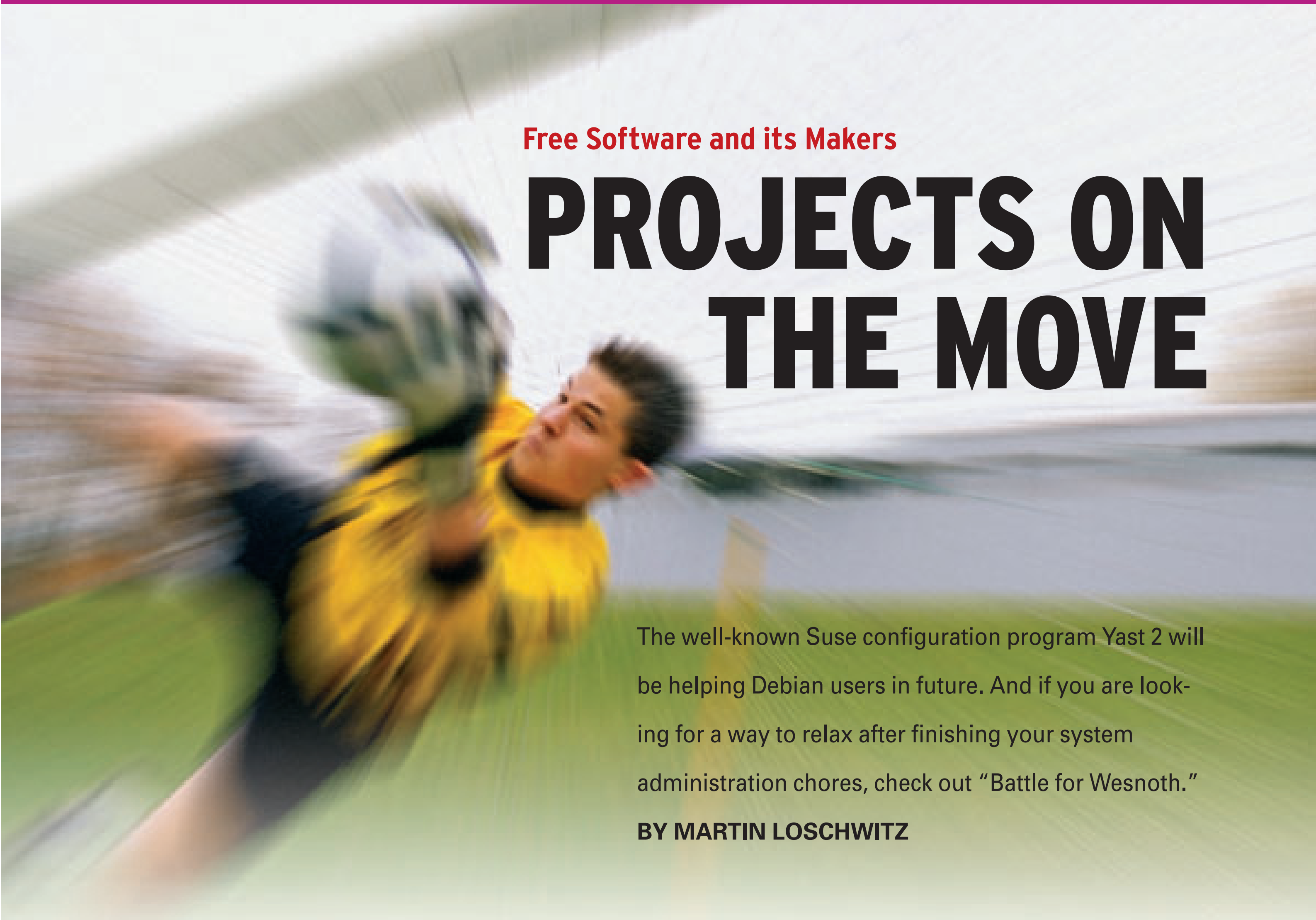


Free Software and its Makers

PROJECTS ON THE MOVE

The well-known Suse configuration program Yast 2 will be helping Debian users in future. And if you are looking for a way to relax after finishing your system administration chores, check out "Battle for Wesnoth."

BY MARTIN LOSCHWITZ

It has been quite a while since "Projects on the Move" talked about a game. The reason for this hiatus is the lack of new and interesting games for Linux. To atone for this lapse, we will focus this month on a role-playing game called *The Battle for Wesnoth* (Figure 1). *The Battle for Wesnoth* is lean enough for older hardware, and it provides an interesting design that more than makes up for the lack of sophisticated 3D graphics.

I'll also talk about Debian's efforts to incorporate Yast 2, and I'll describe the SSHFS file system. I'll close with an update on Debian Sarge.

Gaming Strategy

The queen of Wesnoth is plotting against her husband with the aim of putting a crony on the throne. She has therefore sealed secret pacts with the members of the royal court. The royal magician, who is loyal to the king, discovers the plot and turns against the queen. If you recall playing "Warlords" back in the days of the Amiga, you will feel at home in Wesnoth. *Battle for Wesnoth* also has many elements of the classic "Warlords" game.

Strategy games typically have two modes. Realtime mode means the game carries on regardless of the players; round-based mode means the game will wait for the players to make moves. Wesnoth uses the round-based approach. In contrast to many other strategy games, the game starts off with a full collection of buildings, so players are not expected to get involved in building projects.

A Sight for Sore Eyes

The map shows the kingdom of Wesnoth. The developers have obviously put a lot of TLC into the sound and graphics. Wesnoth thus escapes the fate of many games where an excellent game plan suffers from badly drawn graphics.

A variety of characters and groups, including archers and cavalry, provide for interesting battles. The soldiers have different levels of life-energy, maintenance costs, and even intelligence. Intelligence is expressed by the speed at which the troops learn from their experience in battle. Shamans take care of the wounded.

The game differentiates between close and long distance combat. You can also choose the type of weapons for your attack. Defense is automatic, assuming the unit under attack has the necessary resources. A unit that does not have a close-combat weapon may be doomed. The main objective is to take as many buildings as possible and beat the enemy!

A tutorial helps players enter the nightmarish scenario of medieval combat. Wesnoth is frugal in its hardware and software requirements; it simply requires the SDL library. And if you do not like the maps that come with the game, you can fire up the map editor and design your own. Check out the homepage at [1] for the details.

Yast 2 without Suse

Yast 2 is arguably the single configuration program that has split the Linux community more than any other over the last few years (Figure 2). This invention by Suse has been a thorn in the side of the FOSS (Free and Open Source Software) scene, as it is not released under a free license and thus contradicts the basic philosophy of Linux. At the same time, Yast 2 gives users an intuitive GUI,

Figure 1: In the Battle for Wesnoth, players have to protect court and country against an evil queen.

which both experts and newcomers appreciate.

Users who have not used Suse Linux so far, have had to do without Yast 2. Without a free license, there was no way of porting Yast 2 to other distributions. Fortunately, this situation has now changed. Following Novell's takeover of Suse, Novell announced in March 2004 that they would be releasing Yast 2 under a free license.

The explanation Novell gives makes sense: Yast 2 from Novell will provide a uniform configuration interface for all Linux systems. In the months following the removal of licensing restrictions, there was no obvious reaction. But in November 2004, Mario Fux started thinking aloud about porting Yast 2 on the Debian-desktop mailing list. After all, Debian is lacking in configuration software.

Whereas Mandrake and Fedora have a variety of GUI-based tools for critical administrative chores, Debian users have to resort to the console. Debian even lacks text-based tools for some tasks, and this forces Debian users to fire up their editors and modify configuration files directly in some cases.

Mario Fux surprised bystanders with an interesting announcement. He said he would be willing to pay a reward, and to donate hardware, to the developer that ported Yast 2 to Debian. Fux makes his motivation quite clear; he intends to make Debian an operating system for desktop users come what may.

The Debian developers have been discussing the pros and cons of porting Yast ever since. Software management is regarded as a tricky subject, as Suse and thus Yast 2 are based on RPM packages.

A small group of developers has now formed to support Mario Fux, and the group is busy porting Yast. A chat session at the beginning of February revealed the latest news. Planning is at a fairly advanced stage, and the project looks likely to get underway in earnest in the near future. Mario Fux contacted the main Yast 2 developer at Suse, Klaus Kaempf, to help speed things up. The first target is to create a Debian package for Yast 2 to facilitate the development work. A homepage is being set up to coordinate the developers' activities in the future.

Mario Fux's developer group has discovered one peculiarity; internally, Yast 2 uses a proprietary programming language, Yast Control Language (YCP), for all modules. That makes the problem more difficult. In addition to removing Suse specific elements from Yast 2, the developers need to learn a new programming language that no other project uses.

If the group succeeds in porting the full feature set, Yast 2 could boost the popularity of Debian GNU/Linux as a desktop OS. The only source of information on the porting project at present is the Debian Desktop mailing list [2].

Userspace File Systems

If you have ever had to move a file from one computer to another across a network, you will be familiar with the problems associated with file transfer: FTP is one possibility, but it is insecure. SCP is secure but hardly user-friendly. If you need to delete or rename a file, you need an SSH login.

After logging in you can run familiar and convenient tools such as *cp*, *mv* and *rm*. However, SSH and SCP do not allow users to run these commands remotely. One option would be to add commands for deleting or moving files, like with SCP. However, on the downside, users would need to type the complete command syntax for each operation.

Miklos Szeredi seems to have had enough of this, and developed a new solution. Back in the days of LUFS, he apparently used the SSHFS file system, which simply allowed him to use SSH as a file system driver. After mounting the remote file system using SSHFS, users had access to typical shell commands. Unfortunately, the LUFS maintainer gave up after a short while and a successor was not forthcoming. As Linux 2.6 changed quickly and LUFS needed a Kernel patch, the system is now useless.

Additionally, the LUFS design had a few weaknesses that made it hard to maintain. And to top this off, an alternative was already underway: Fuse ("Filesystem in Userspace") closely resembles LUFS from the user's viewpoint, although it uses a completely different approach. Miklos Szeredi had the idea of re-implementing LUFS-SSHFS based on Fuse and removing the design errors in the process. The result is also called SSHFS.

So how does Fuse work? First of all, it needs a Kernel that supports Fuse. Fuse is not a genuine file system but a framework that can load separate user space drivers. The kernel-based Fuse driver translates between the kernel and the

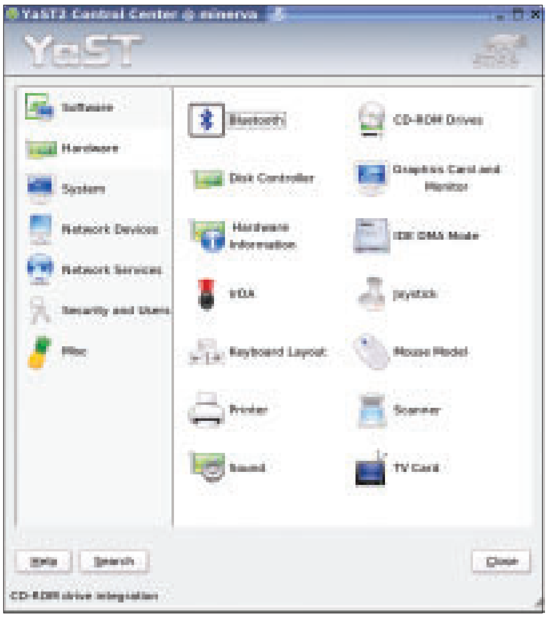
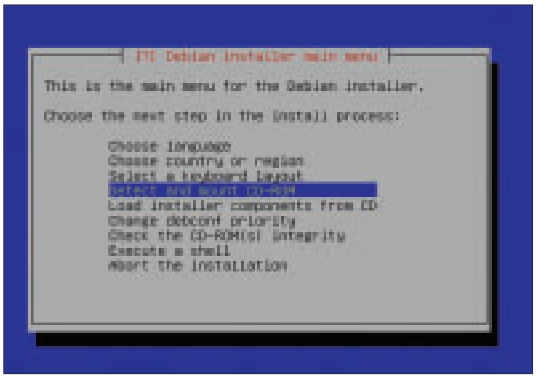


Figure 2: Suse-specific Yast components need to be removed to port Yast to other distributions.



```
[!!] Debian installer main menu

This is the main menu for the Debian installer.

Choose the next step in the install process:

        Choose language
        Choose country or region
        Select a keyboard layout
        Detect and mount CD-ROM
        Load installer components from CD
        Change debconf priority
        Check the CD-ROM(s) integrity
        Execute a shell
        Abort the installation
```

Figure 3: Problems with the Debian Installer: it still uses DevFS, an orphaned project that is due for removal from the kernel in the near future.

user space driver. Superficially, the finished product looks like a normal file system; major operations such as reading, copying, or deleting files can be performed using the normal user space tools. One of Fuse's advantages is that users without *root* privileges can mount the file system. Users need to load the Fuse module into the kernel to do so.

Tried and Trusted

SFTP can also handle file management on remote file systems. However, the operations take place within an SFTP session, so transporting data between the systems is more complicated than with SSHFS and the standard *cp*, *mv* and *rm* tools. Thanks to SSHFS, users can easily run scripts and other automatic tools on the remote system.

Another advantage that SSHFS has its support for multi-threading. This allows multiple jobs to run consecutively.

The Fuse framework is mainly responsible for making it easy to integrate an SSH session. But the options are more or less infinite. For example, users could mount Google's G-Mail service via GmailFS, which is also based on Fuse, just like an NFS export. Users could then read mail using their favorite mail clients.

The performance of the Fuse file system is not up to what you would expect of a local or special-purpose network file system, however, assuming you have a quick and stable connection, Fuse can be a useful option.

Debian Installer and DevFS

The next release of Debian GNU/Linux, codename Sarge, will not need boot floppies for the install. Instead, it will have the new Debian Installer (Figure 3) with a completely new and more robust architecture. Joey Hess published the version of the Debian Installer that will be included in the Sarge release in January. But now it appears the apparently infinite release process, which has postponed the appearance of Sarge for months, has started to reek revenge.

The major issue is DevFS: This solution for an organized and uncluttered */dev* directory, which dates back to the days of Linux 2.4, fell out of favor with the developers. Design errors make the code difficult to maintain. Linux 2.6 has a new approach: UDev. Although DevFS is still part of the current kernel, there is a move to remove it for good. For example, Linus Torvalds announced that it would be removed from the kernel mid 2005. This will give distributors an opportunity to replace the old DevFS with UDev in the meantime.

This affects the Debian Installer, which will install Linux 2.6 if required and enable DevFS when doing so. Changing this would mean some fairly low-level work, which Joey Hess has already ruled out. Sarge will therefore be based on obsolete and unsupported software when it is finally released. This may see harmless at first glance – after all, stable Debian versions do not use the latest kernel – remember that kernel developer patches need to be applied in case of security issues.

And this is the biggest danger with DevFS, as there is no official DevFS maintainer. Both the kernel and user space parts of DevFSD have been orphaned for a while now. This means that the Debian security team would actually need to maintain DevFS to remove critical errors or remedy security issues. This is a lot of work, as Sarge will have a lot more packages than the current Woody release; thus the total load on the Security Team will grow considerably.

The Path to Sarge...

Problems of this kind can only be avoided in future by drastically cutting the Debian release intervals. This said, there is still no sign of the Sarge release, and it has been stranded in a more-or-less undefined state for months now. In a posting to the Debian-Devel-Announce mailing list [2] a few weeks ago, Steve Langasek – a member of the Debian Release team – announced that a number of critical issues had been solved in the testing branch thanks to new GCC, KDE 3.3 and Perl packages. He also mentioned that a lot of hard work was being put into security support for Sarge, and that hardware was available for the supported architectures.

... is long and stony

This is like the light at the end of the tunnel, as missing security support has recently been viewed as the major obstacle. To support this positive trend, another Bug Squashing Party took place in the first week of February, and the participants succeeded in removing more issues. If you are interested in reading Steve Langasek's message, in which he also comments on future steps, check out the mailing list archives at [3].

The members of the Security Team are only too familiar with the issue of obsolete software. It has become increasingly difficult to find security patches for Debian Woody. The current release is over two years old – new security patches are often extremely difficult to install. If it takes years for the successor to Sarge to appear, the question is how the Security Team can be expected to support DevFS.

That's all folks...

... for this month at least, but we do have one request before we go: if you can recommend a program that you would like to see featured in *Projects on the Move*, why not mail me with your suggestion [4]? I look forward to your comments! ■

INFO

[1] The Battle for Wesnoth: *http://www.wesnoth.org/*

[2] Debian-Desktop mailing list: *http://lists.debian.org/debian-desktop/*

[3] Steve Langasek on Sarge updates: *http://lists.debian.org/debian-devel-announce/2005/01/msg00011.html*

[4] Tips and suggestions: *projects@linux-magazine.com*