

Managing your system with YaST

SUPER TOOL

One of the highlights of openSUSE is YaST, the Swiss Army knife of configuration tools for Linux. YaST will help you with everything from managing hardware to configuring users.

BY JOE "ZONKER" BROCKMEIER

When you first launch YaST, it might seem a bit overwhelming because this comprehensive tool includes dozens of modules for system administration and management, including hardware, network devices, network services, Novell AppArmor, security and users, software, and system virtualization.

Some system management tools only make it possible to administer a system with a GUI or command-line tool. YaST lets you use the same tool no matter what environment you choose. Whether you prefer Gnome or KDE, or if you're shelled into a server over SSH, you can perform the same management tasks with YaST and its modules.

YaST has three separate front ends: Gtk for Gnome, Qt for KDE, and an ncurses (a menu toolkit for consoles) front end that can be used at the console. If you're logged into Gnome or KDE and try to run YaST from a terminal on your desktop, it will default to the correct front end for that desktop. To launch the ncurses front end, run

```
yast --ncurses
```

and you'll get the ncurses interface.

ncurses YaST Interface

The Gtk/Qt interfaces for YaST are pretty self-explanatory: Apply mouse clicks liberally to interface as needed, rinse, repeat, and so on. You can also navigate YaST's GUIs from the keyboard to some extent, but I'm assuming that it won't be hard to get around YaST's GUI interface.

Getting around the ncurses interface isn't hard, either, but it might not be in-

YaST – Yet another Setup Tool or Your awesome Setup Tool – has a long history with SUSE. SUSE, which was called S.u.S.E. for “Software- und System-Entwicklung” (Software and System Development), was a Slackware-based distro with an early version of YaST. Originally, YaST wasn't the buffed-up, GUI-enabled tool you now know and love – the first iterations of YaST were text mode-only, and nowhere near as full-featured as YaST is today.

In 1999, YaST2 made its first appearance in SuSE Linux 6.3, as the installer. Over time, YaST has continued to evolve and mature. After Novell acquired SUSE in 2004, YaST was released under the GPL. In this article, I'm really talking about YaST2, but to simplify things, I'll just call it YaST.

```
File Edit View Terminal Tabs Help
Linux-jqr6:/home/jzb/Desktop # yast2 timezone help

YaST Configuration Module timezone
-----

Time zone configuration

Basic Syntax:
yast2 timezone interactive
yast2 timezone <command> [verbose] [options]
yast2 timezone help
yast2 timezone longhelp
yast2 timezone xmlhelp
yast2 timezone <command> help

Commands:
list      List all available time zones
set       Set new values for time zone configuration
summary   Time zone configuration summary

Run 'yast2 timezone <command> help' for a list of available options.

Linux-jqr6:/home/jzb/Desktop # █
```

Figure 1: Help in the timezone module.

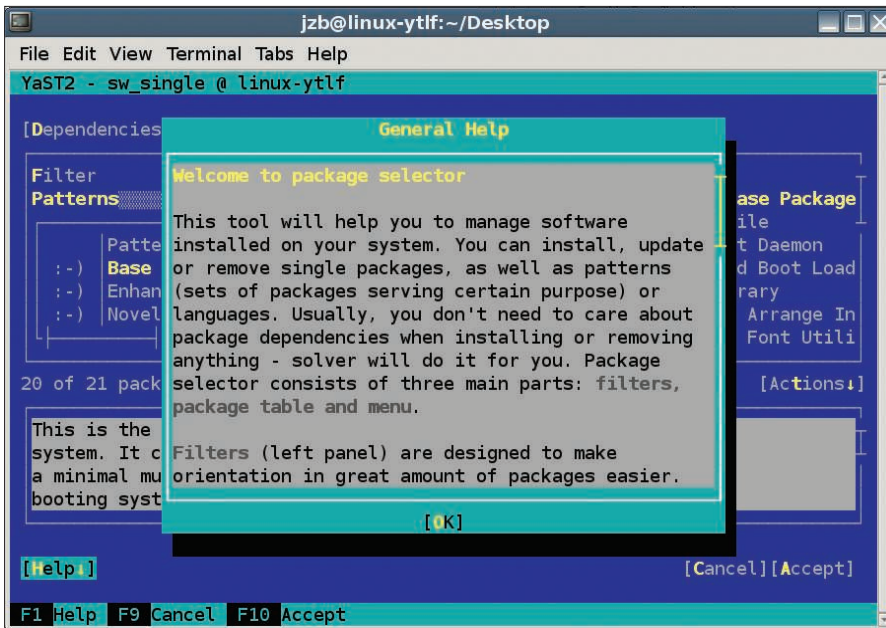


Figure 2: Alt plus the highlighted letter activates the function.

tuitive. Plus, it's possible that your desktop environment or terminal emulator might grab some of the keystrokes (such as F1, if you're using Gnome) before they reach YaST. To get around in YaST, use the Tab key to move between "panes" in the YaST window. To move within the various panes, use your arrow keys. To select something, just hit Enter.

If you have a button with a highlighted letter, as shown in Figure 2, you can activate its function with Alt + the highlighted letter. For instance, to dismiss the help dialog, you'd hit Alt+O; however, if you're using a terminal emulator or window manager that grabs Alt+O first, you can use Esc+O instead.

If Tab (to move forward) and Shift + Tab (to move backward) are bound for something else, you can also move forward with Ctrl+F and backward with Ctrl+B. To manage systems remotely, you'll want to be comfortable with YaST's ncurses interface.

If you use a module frequently, it might be more convenient to launch just that module rather than sorting through all of YaST's modules. For example, you might have already noticed that when you run the *Install Software* command from the menu, it's actually running:

```
sudo yast2 --install
```

To get a full list of modules, run:

```
yast2 -l
```

You should see a full list of available modules. So, for example, if you want to manage a list of users who can run sudo on your system, instead of launching YaST and then looking for the sudo module, run the following as root:

```
yast2 sudo
```

If you prefer, you can preface it with *sudo*. You can also get help for a specific module using:

```
yast2 module help
```

and

```
yast2 module longhelp
```

For instance, help on the timezone module looks like Figure 1. Depending on the module maintainer, the help ranges from comprehensive to a bit sparse.

As you can see here, you can launch YaST to manage time zones with

```
yast2 timezone interactive
```

or you can get additional help for the timezone module using

```
yast2 timezone set help
```

to find out about the *set* command, or get all timezones with

```
yast2 timezone list
```

and so forth.

Some Handy YaST Modules

Even after using openSUSE for my daily work for quite some time, I haven't had to use every YaST module available – and I probably never will. There is so much functionality within YaST that it's unlikely that any user is going to interact with every module. Therefore, I will focus on an "introductory" set of modules that will help you get started with system management.

Because software installation with the use of YaST's tools are covered elsewhere in this magazine, I'll start with User management.

Managing */etc/sysconfig*

If you want to configure some of the settings in one fell swoop, launch the YaST *sysconfig* module, which will let you modify settings in */etc/sysconfig*. (Note that it's not a good idea to modify some of the configuration files by hand because YaST can and will overwrite them.) You can either launch this under the System group, or you can run the command:

```
yast2 sysconfig
```

Either way, you'll see a window that looks like that shown in Figure 3.

Using this module, you can easily configure any setting found in */etc/sysconfig* – from the number of days files will be stored under */tmp* before being deleted by a cron job, to the default bootloader.

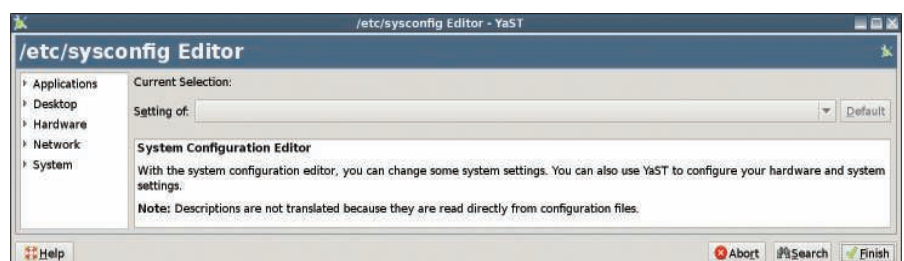


Figure 3: The YaST */etc/sysconfig* editor.



Figure 4: Changing the default bootloader with YaST.

On the left-hand side, you'll find a list of settings that can be changed and you'll see a description of the setting along with a drop-down menu of the options on the right-hand side. For instance, Figure 4 displays the bootloader option within the `/etc/sysconfig` Editor. As you can see, YaST is a much nicer tool for making configuration changes here, in that it gives some description of the options and it rules out invalid options. For example, you can't enter "true" for a configuration directive that only accepts "yes" and "no" options.

SaX2

One of YaST's more popular tools is the SUSE advanced X11 configuration module: SaX2. With SaX2, you can configure your video card and display, mouse, keyboard, and supported tablets and touch screens. You can launch SaX2 in several ways. If you're in YaST's Control Center, go to Hardware and select *Graphics Card and ...* (yes, it has ellipses in the title).

From the command line, you can run:

```
sax2
```

or

```
yast2 x11
```

After SaX2 launches, you'll see a window with five options on the left-hand side: Monitor, Mouse, Keyboard, Tablet, and Touchscreen. Usually, your monitor will be detected and configured properly during installation. However, if you upgrade your video card or monitor, or if it wasn't properly configured in the beginning, you can make changes using SaX.

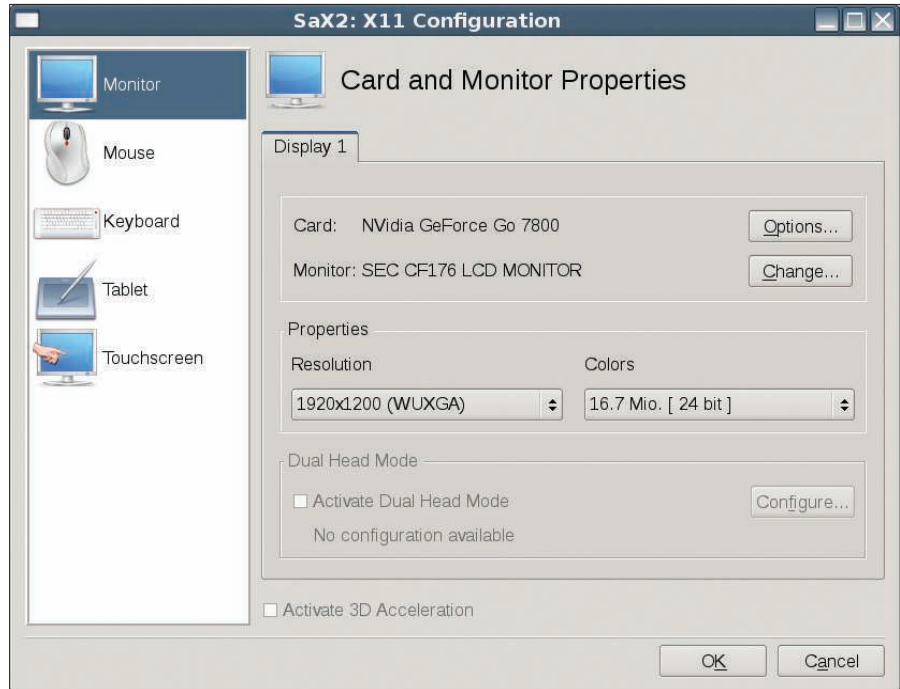


Figure 5: SaX2 X11 configuration.

To change your video settings, select *Monitor* on the left-hand side. (*Monitor* is a bit misleading, because you can also change your video card settings here.) To change your video card settings, click *Options* where it says *Card*, followed by the name of the card that was detected during setup.

Also, note that you can start SaX without X running, so that is can help you configure X11 (see Figure 5). This is par-

ticularly useful if you don't have a working X configuration.

Also, you can set your monitor's resolution and frequency ranges in the Card and Monitor Properties dialog. Clicking *Change* next to the Monitor line lets you configure the display size and sync frequencies, if they haven't already been detected properly. Note that the sync frequencies are usually not terribly important for LCD monitors. However, it's

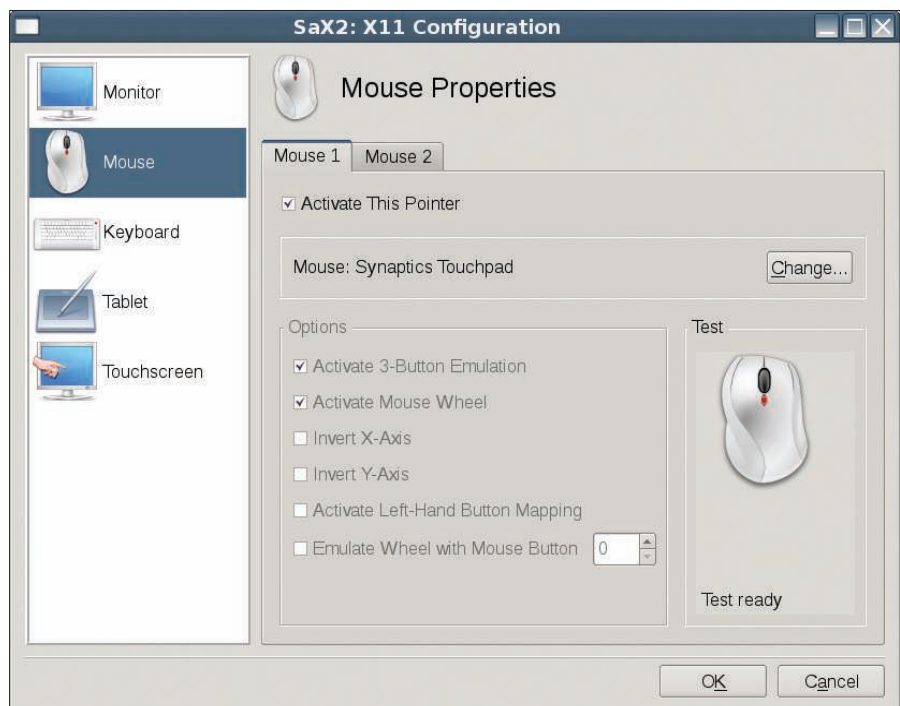


Figure 6: Mouse Properties in SaX2.

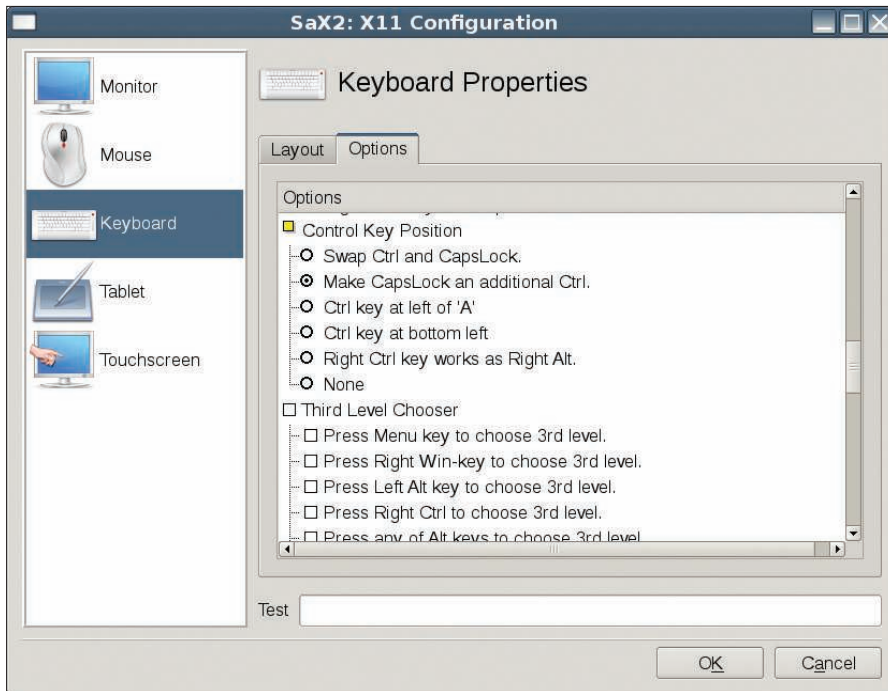


Figure 7: Keyboard Properties.

vitaly important that you don't enter frequencies that are out of range for CRT monitors because it can damage some older monitors. (Most monitors will simply display "out of range" if they get input that they can't handle, but it's best not to take chances.) You can also tweak the default resolution and colors here and enable 3D acceleration if it's available with your video driver. If you have multiple monitors, this is also where you'd enable dual-head mode.

If you want to manage your mouse or keyboard, you can do so within SaX. If you want to make changes to your mouse, click *Mouse* (Figure 6). If you're left handed, you can swap the mouse keys to correspond to left-handed use.

If you have more than one pointing device on the system, you'll see multiple mouse tabs. For instance, I usually attach a USB mouse to my laptop, so it displays a pointer for the touchpad and the USB mouse. Also, you can disable a pointing device, such as the touchpad. This might be useful if you accidentally hit the touchpad while typing and don't care for random pointer input.

If the mouse wasn't detected correctly, it's also possible to change the type of mouse. Occasionally, a mouse will be assigned a "generic" mouse profile when a more specific profile might be better, such as in the case of some Logitech mice.

If you're like me, you probably don't care for the Caps Lock key – it's not terribly useful and can be a bit of a pain if you accidentally hit it while typing. If you'd like to make the Caps Lock key a bit more useful, you can use the Keyboard Properties module to turn it into a Ctrl key instead. To do so, go to *Keyboard | Options*, then scroll down to *Control Key Position*, click on *Make*

CapsLock an additional Ctrl, and select *OK* (Figure 7). If you prefer, you can also swap the Ctrl and Esc keys or otherwise modify it with some of the options under *CapsLock Key Behavior*.

Network Configuration

For "standard" configurations, your network will usually be detected and configured correctly during the installation process. My experience using openSUSE 11.0 on laptops and desktop systems has been that I don't need to do anything additional to configure my network settings. However, my network configuration fits in a pretty standard-use case – I use desktop machines hooked up to Ethernet with DHCP and laptops set up either to configure to wireless access points, or Ethernet with DHCP, or with an EV-DO card. In all three cases, I can configure my network settings through NetworkManager or KNetworkManager (depending on whether I'm in Gnome or KDE, respectively).

If you need to configure a static setup, multiple Ethernet cards, or a modem to connect to dial-up service, you might need to do a little tweaking.

Printers

If you're connecting your system to a printer over a USB connection, odds are that you'll be prompted to set it up the

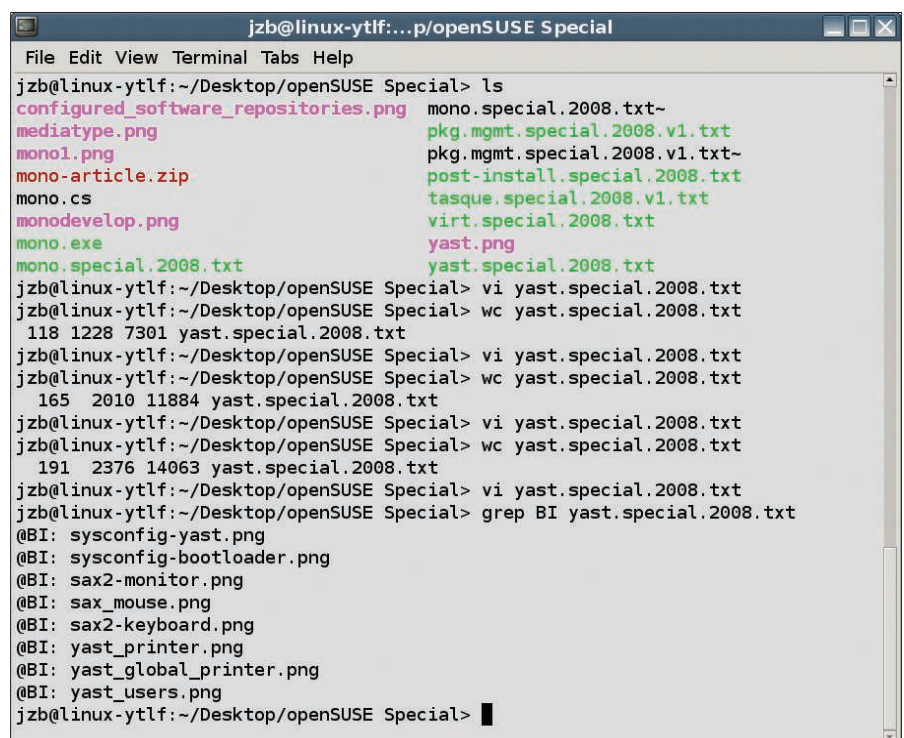


Figure 8: YaST Printers and Queues dialog.

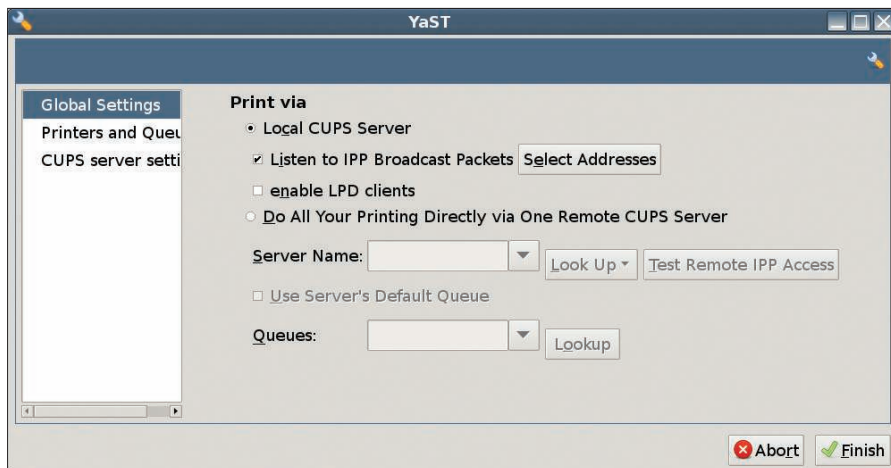


Figure 9: YaST's dialog for global printer settings.

first time you connect the USB cable to your system with the printer powered on. But, if you are connecting to a networked printer or doing something a bit more complex, you might need to set your printer up manually with YaST.

You can either select *Printer* from the *Hardware* group, or run

```
yast2 printer
```

from the command line. Note that the Printer module also refreshes the package list when it's started. If you've recently refreshed the list, you can tell YaST to skip the auto-refresh. Even though YaST's package management has sped up considerably since 10.3, the refresh still takes a few seconds and should be unnecessary unless it's been a few days since your repository information has been updated.

When you fire up the Printer module, you'll first see the Printers and Queue dialog (Figure 8), which lists any printers you have already defined. If you want to add a new printer, click *Add* from the lower left-hand side. YaST will then take you through a set of dialogs to pick the type of printer. Note that you can also configure an additional queue for an existing printer. The many additional options and printer models should be relatively self-explanatory for most printers.

If you're in an office environment or if you've set up a single CUPS server in your home, you can point your system at the remote CUPS server using YaST's printer configuration (Figure 9).

If you have a running CUPS server, click the radio button next to *Do All Your Printing Directly via One Remote CUPS*

Server and then select one of the scan options from the *Look Up* menu. The odds are, YaST will find the remote server, as long as it's configured correctly; otherwise, you can enter the server name or IP address here.

Managing Users

With the user-management features (Figure 10), you can add, manage, and remove users from the system. First, go to *Security and Users* and click *User and Group Management*, or launch it from the command line with:

```
yast2 users
```

To use the ncurses interface, use:

```
yast2 --ncurses users
```

During installation, you're provided with the option of logging in a single user automatically. If you unchecked the option during install, but have decided you'd like to enable automatic login (or want to switch the user that logs in automatically), you can do that here. Click the drop-down menu for *Expert Options* and select *Login Settings*. Click the checkbox next to *Auto Login* and then select for whom you want to allow auto login. If your system is in a secure environment, you can also allow for logins without a

password. Of course, this is not recommended if you are setting up a laptop or a system that will be exposed to multiple users, but it can be much more convenient if you're setting up a home system for a single user. Conversely, you can also disable automatic login here to tighten up your system security a bit.

To add new users, click the *Add* button and fill out the required information in the *New Local User* dialog. Note that you shouldn't need to change anything under *Details*, *Password Settings*, or *Plug-Ins* unless you have some detailed changes you want to make. Generally, the default settings are fine. However, if you want to add a new user to a few additional groups, go to the *Details* tab and select the groups you want them to join.

Usually, I add new users to the "wheel" group if I want to give them sudo access. If your new users prefer something other than GNU Bash, or if you want to set the shell to */bin/false* so they can't actually log in to the system, you can change the login shell. After you're done, click *OK*. To add a lot of users and to use new user defaults other than openSUSE's, you can modify these under *Defaults for New Users*.

Summary

To cover everything about YaST in one place would require a book, but I hope that this article helps you get started on the road to system management with YaST and encourages you to spend some time getting to know YaST. YaST is an excellent utility that makes life much easier for Linux users who want to take control of their systems without needing to spend much time hacking text files. Fire up YaST and have fun! ■



Figure 10: YaST's User and Group Administration module.