
Basic tasks for new Linux users

Help! I got my Linux installed. Now what?

Skill Level: Introductory

[Ian Shields](#)

Senior Programmer
IBM

07 Apr 2011

Updated 19 Nov 2011

If you're a Windows user, you may find your new Linux system to be a bit different from what you're used to. One of your first tasks will probably be installing additional software, including some of IBM's extensive software for Linux. In many cases, installing software on a Linux system requires you to log in, become the super user (or root) for some tasks, open a terminal or shell window, and mount a CD-ROM, DVD, or USB drive. This article guides you through these basic first-time user tasks and more.

Section 1. Introduction

Basics of Linux system administration: Working at the console

<http://www.ibm.com/developerworks/training/kp/l-kp-command/index.html> Basics of Linux system administration: Setting up your system and software

<http://www.ibm.com/developerworks/training/kp/l-kp-setup/index.html>

If you're a Windows user who wants to try Linux, and you are new to Linux, this tutorial is for you. We'll cover a few basic tasks that may be different from what you are used to. We'll cover these tasks:

Login

Logging in and understanding the Linux desktop

Navigation and settings

Tweaking settings to suit your personal needs

Command line access

When you really need to work at the command line

Becoming superuser (or root)

When ordinary user authority isn't enough

Using a GUI application as another user

Getting comfortable with multi-user system capabilities

Removable devices

Using removable media and making sure you don't lose data when you detach removable devices

Logout

When you're done for a while or need a panel icon for logout or other applications

Adding users and groups

Adding more users or user groups to your system

These instructions and examples focus on three popular distributions and their default graphical desktops:

- Fedora 14 using a GNOME desktop
- OpenSUSE 11.3 using a KDE 4 desktop
- Ubuntu 10.10 using a GNOME desktop

Other distributions, desktops, and window managers are available, and we encourage you to explore further on your own. Even if some of the information here is specific to a particular distribution and even a particular version of a distribution, what you learn here should help you navigate a strange distribution. Because this tutorial simply aims to help you get started, you won't find information on more advanced tasks like recompiling your kernel or installing software. We recommend our no-charge [Linux \(LPI\) certification self-study guides](#) for deeper information.

At the time of writing, GNOME 3 is likely to be available soon. Expect changes in that user interface from what is described here.

Before using these instructions, install a Linux system and create at least one *non-root* user as part of your installation process.

Section 2. Login

Once you have installed your Linux distribution and booted your system, either you will be automatically logged in or you will see a login screen. The next section shows how to switch between these two startup methods and discusses the security implications of automatic login. For this section, we'll assume you are logging in. The three systems we consider in this tutorial implement login a little differently from each other, but all will prompt with the name of one or more users. Once a user is selected, you need to enter a password.

The login screen

The appearance of the screen varies according to your Linux distribution, and it will probably come as no surprise that you can customize it further, although we won't cover that in this short tutorial. A typical login screen for an Ubuntu 10.10 system is shown in Figure 1, and a typical one for Fedora 14 is shown in Figure 2. These both illustrate the default GNOME desktop used on these systems.

Figure 1. Ubuntu 10.10 login screen

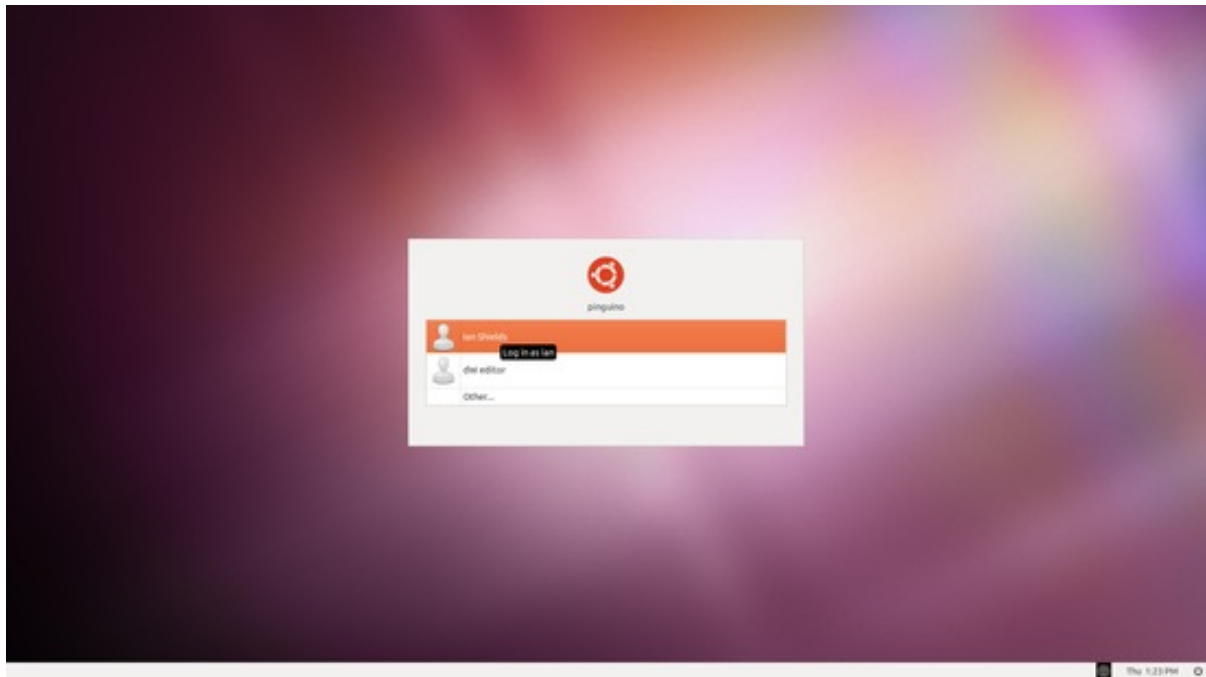
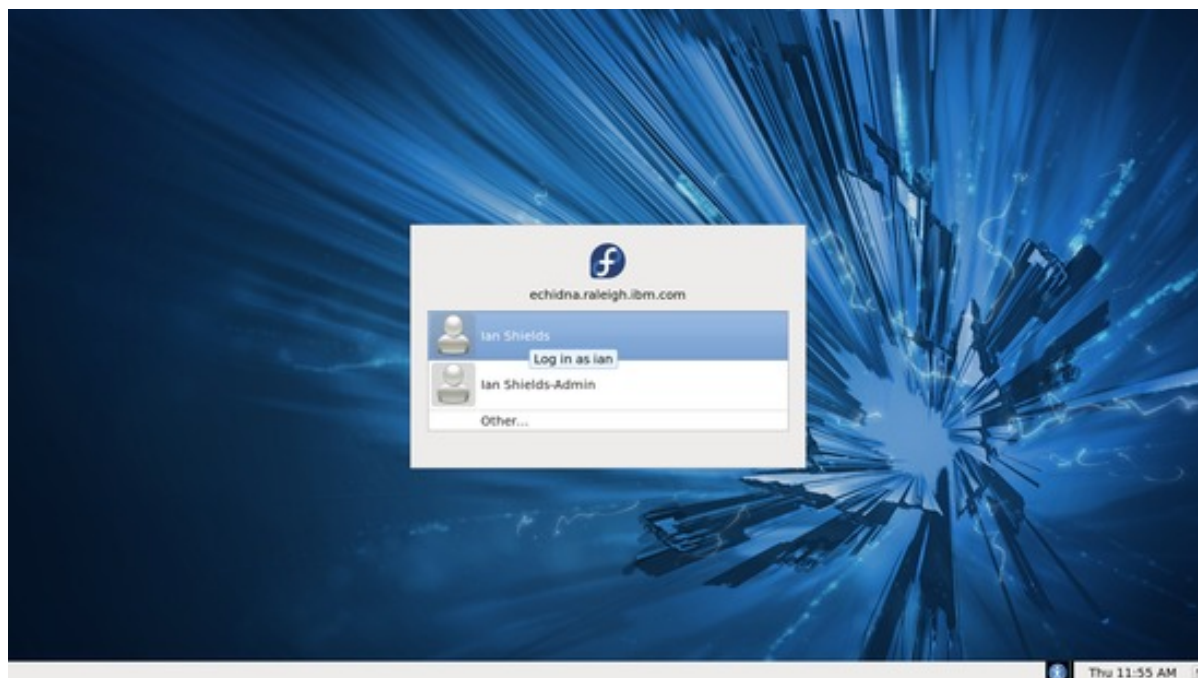
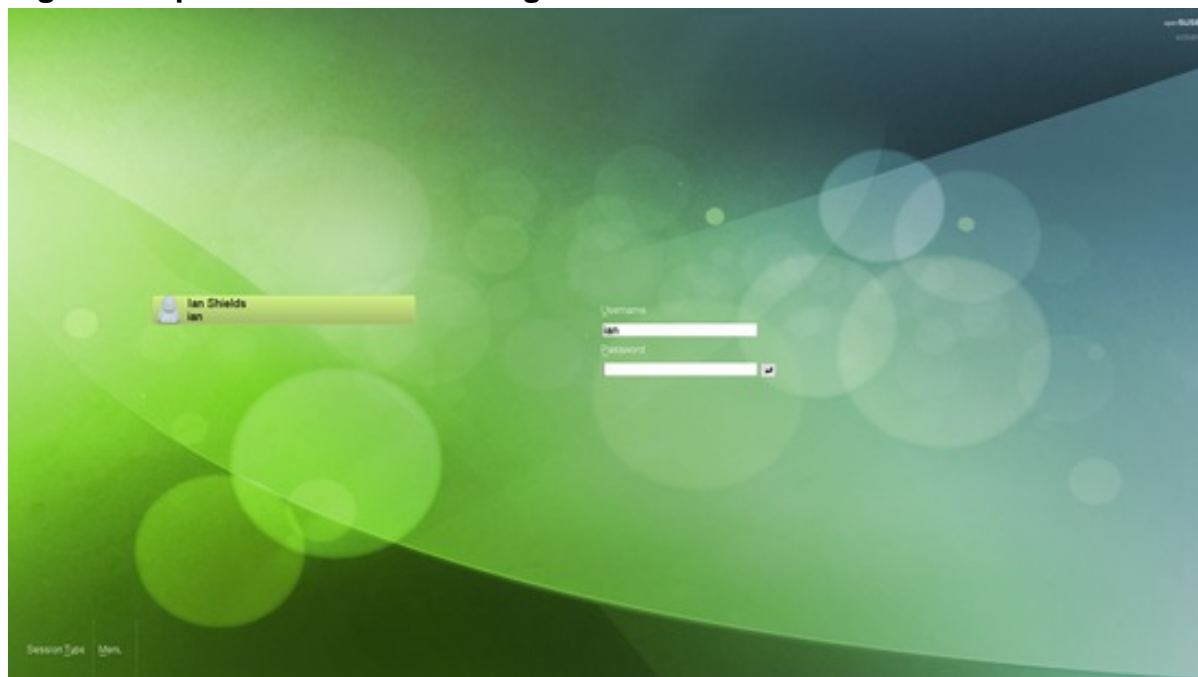


Figure 2. Fedora 14 login screen



A typical login screen for an OpenSUSE 11.3 system with the default KDE 4 desktop is shown in Figure 3 or

Figure 3. OpenSUSE Linux 11.3 login screen



When you enter or select an id and press **Enter**, you will be prompted for your password. If a password field is showing on the login screen, as in the OpenSUSE example here, you can tab to the password field and enter your password. If there is no entry field for an id showing and you need to enter an id that is not in the list of

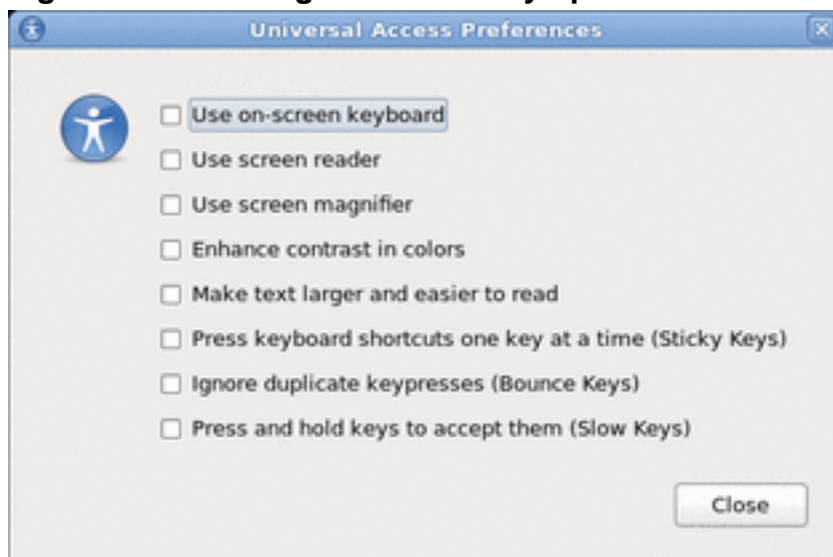
available ids, you will usually find an entry for something like "other..." as shown in Figure 2. Select that and you should see an entry field for the id. We'll talk more about this in the section [Becoming superuser \(or root\)](#).

Login screens may have other items on them, including a clock, perhaps the name of the system, and icons or named menus that allow you to shut down or restart the system.

Accessibility

On the GNOME desktops is a small icon that looks something like a figure of a person inside a circle. Click this and you will see a dialog for accessibility options such as an on-screen keyboard or larger font. An example from our Fedora system is shown in Figure 4. We used the option to make text larger and easier to read for the login screen shown in [Figure 2](#).

Figure 4. Fedora login accessibility options

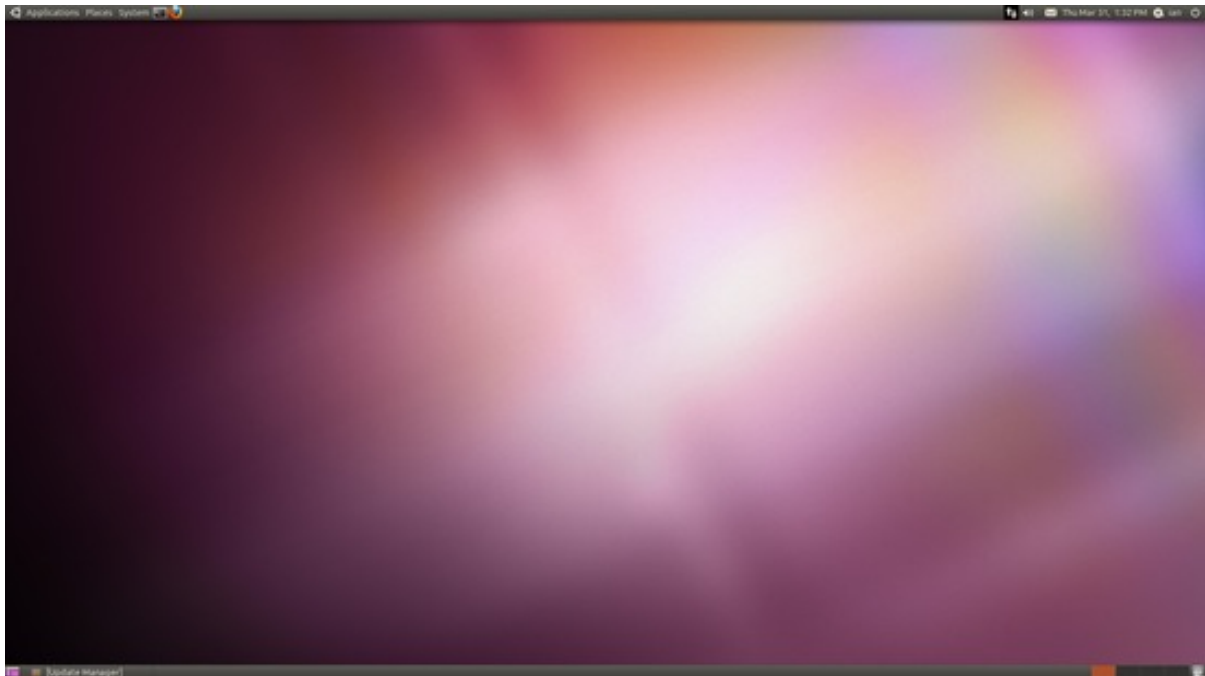


The desktop screen

After you type in your password and press **Enter** again, you should be logged in and see your desktop. Figure 5 shows what you might see as user `ian` on an Ubuntu 10.10 system, with a *panel* along the top and another along the bottom. To explore the desktop, move your mouse over the icons or click on them.

Note: The next three images are intended to give you an impression of the way your desktop will look. Don't worry if you can't read the tiny print on them.

Figure 5. Sample initial window for Ubuntu 10 and GNOME desktop



Fedora 14 also uses a GNOME desktop. In Figure 6, we clicked the **System** icon on the left part of the top panel and then selected **Preferences**. As we hover over the **Desktop Effects** choice in the subsidiary menu, the hover help shows **Select desktop effects**. This is where you select desktop effects such as having your window edges wobble as you move them around the desktop. Note that this requires 3-D graphics capabilities and possibly a graphics driver that is not open source.

Figure 6. Sample initial window for Fedora 14 and GNOME desktop

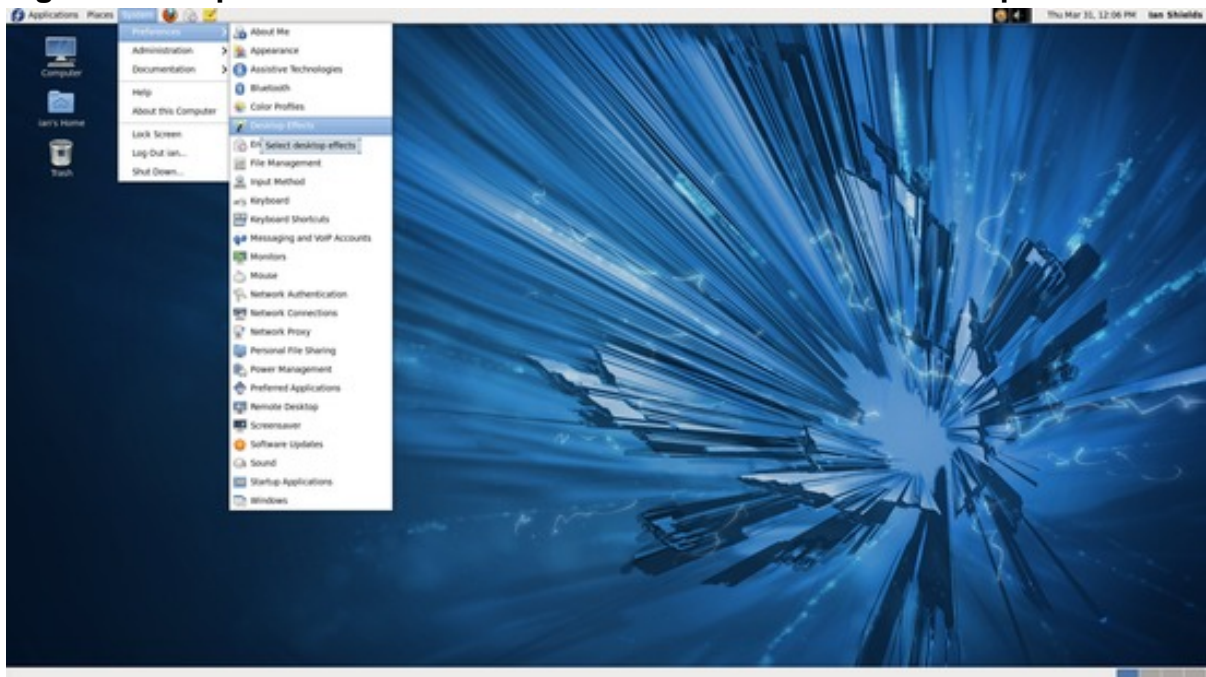


Figure 7 shows what you might see with OpenSUSE and a KDE 4 desktop.

Figure 7. Sample initial window for SUSE Linux and KDE desktop



The relatively new KDE 4 desktop uses a different navigation metaphor, which we'll discuss more in the section [Navigation and settings](#). Note that the window menu for Desktop Folder slides out to the left or right of the window, rather than being fixed at the top.

Section 3. Navigation and settings

Let's spend a moment exploring the Linux panels on the desktop and then look at navigation and an example of how you can customize your desktop by switching to left-handed mouse usage.

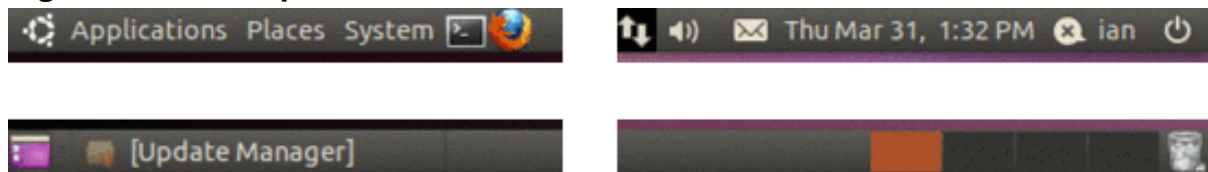
Desktop panels

Panels give you a starting place for interacting with your desktop and provide information about your system. You will usually find one or two panels on your desktop. Typically these will default to being on the top, the bottom, or both the top and bottom edges of the screen. You can move them to the left or right edges if you wish, typically by right clicking and changing the properties.

Different distributions and different desktops often differ in panel layout, so expect

differences here. Our Ubuntu GNOME panels are shown in Figure 8. We've shown the ends of the panels for clarity.

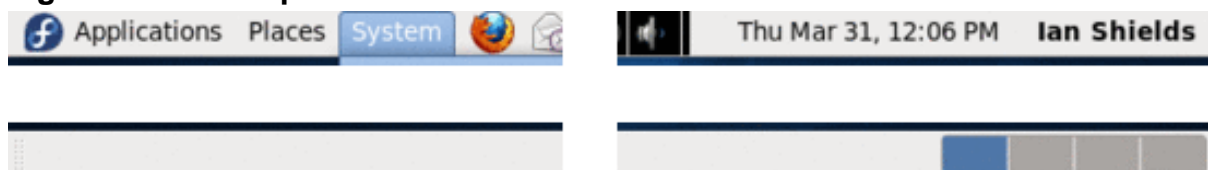
Figure 8. GNOME panel features on Ubuntu



- The left part of the top panel provides a launching place for accessing programs, folders (directories), or system settings and information.
- The right part of the top panel provides information such as time and date, along with several quick-access buttons for functions like setting volume control, opening chat windows, and logging out.
- The left part of the bottom panel has a button to hide all windows and show the desktop, along with buttons for active windows.
- The right part of the bottom panel contains a trash bin and four rectangles that allow you to navigate between your virtual desktops. Most Linux systems set up multiple distinct desktops, so you can keep your email and web browsing on one desktop, while doing program development on another and testing on yet another, for example. You switch between them by clicking the appropriate smaller square or by using a key combination. For GNOME, **Ctrl-Alt-left arrow** or **Ctrl-Alt-right arrow** usually cycles through them in the same way that **Alt-tab** will cycle through application windows on a given desktop. For OpenSUSE, **ctrl-F1** through **ctrl-F4** will directly select desktops 1 through 4. When exploring, make sure you log in as a non-root user to avoid accidents; such mistakes may be more serious when you have unlimited authority.

The panels for our Fedora system are shown in Figure 9. The System item in the upper left is selected as we used the image from Figure 6 to create this image. Notice that we do not have quick access to the logout function in the upper right, the hide all windows function in the lower left, or the trash icon that we saw in the lower right of the Ubuntu panels. Otherwise, the panel layout is reasonably similar.

Figure 9. GNOME panel features on Fedora



The OpenSUSE system has a panel across the bottom only, as shown in Figure 10. Access to programs as well as folders and system functions starts with the large button at the far left, which we will refer to as the **Start** button. Also on the left are

quick-access buttons to a browser and desktop navigator, followed by buttons for each of the five virtual desktops. On the right end of the panel, you find a clock and several convenience buttons similar to those at the right end of the upper Ubuntu GNOME panel.

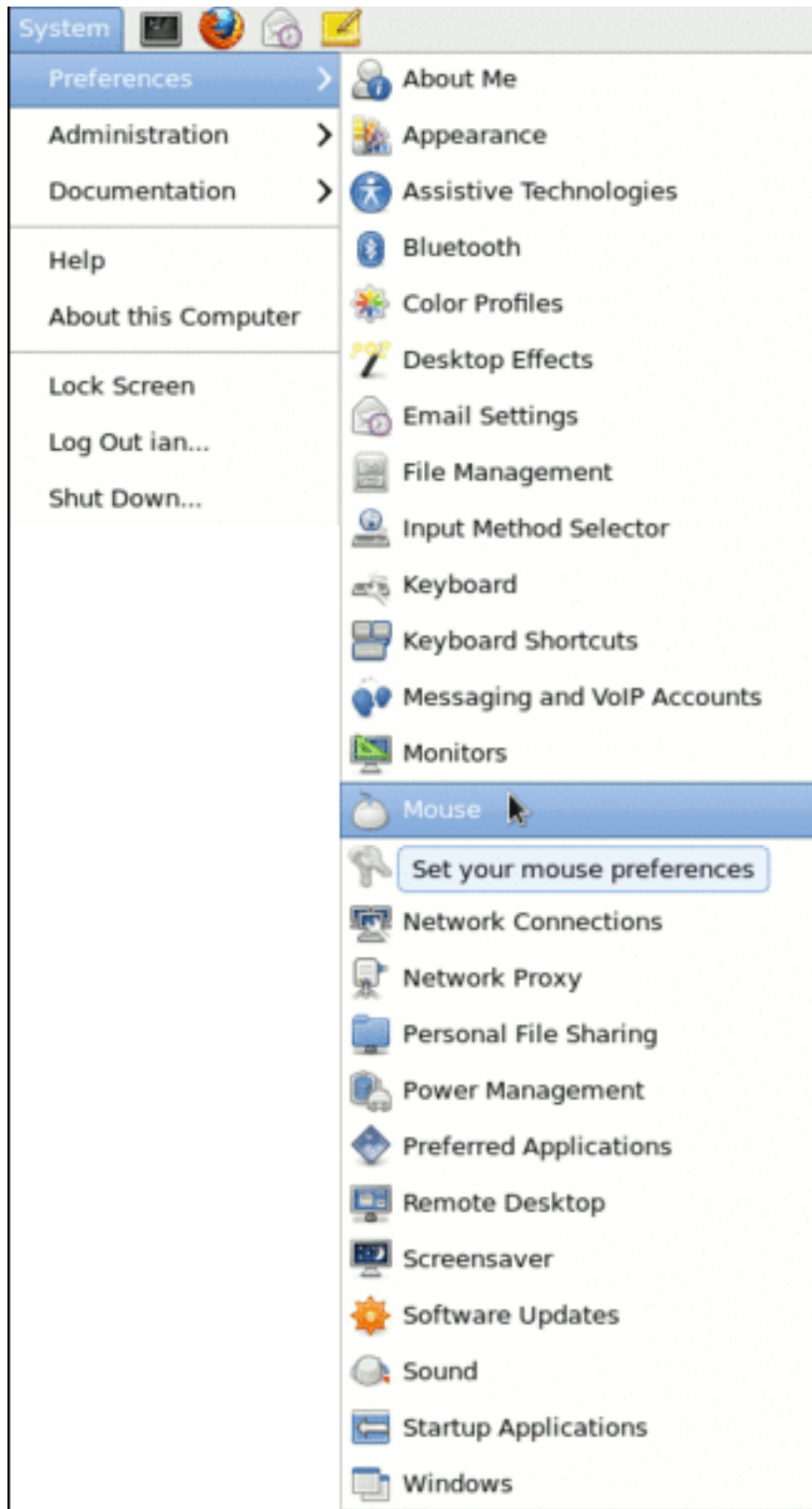
Figure 10. KDE 4 panel features on OpenSUSE



Navigating

The GNOME 2.3x desktop uses the cascading menus that have now become familiar. Figure 11 illustrates how to access the mouse settings from the Fedora System icon. Different distributions may arrange these menus differently. For example, you will find the mouse settings on Ubuntu in the same location, but if you are looking for the Desktop Effects preferences, which you find under **System->Preferences->Desktop Effects** on our Fedora system, you find it as the **Visual Effects** tab under **System->Preferences->Appearance**. Exploring graphical applications is often like turning over different rocks to see what is hiding underneath them.

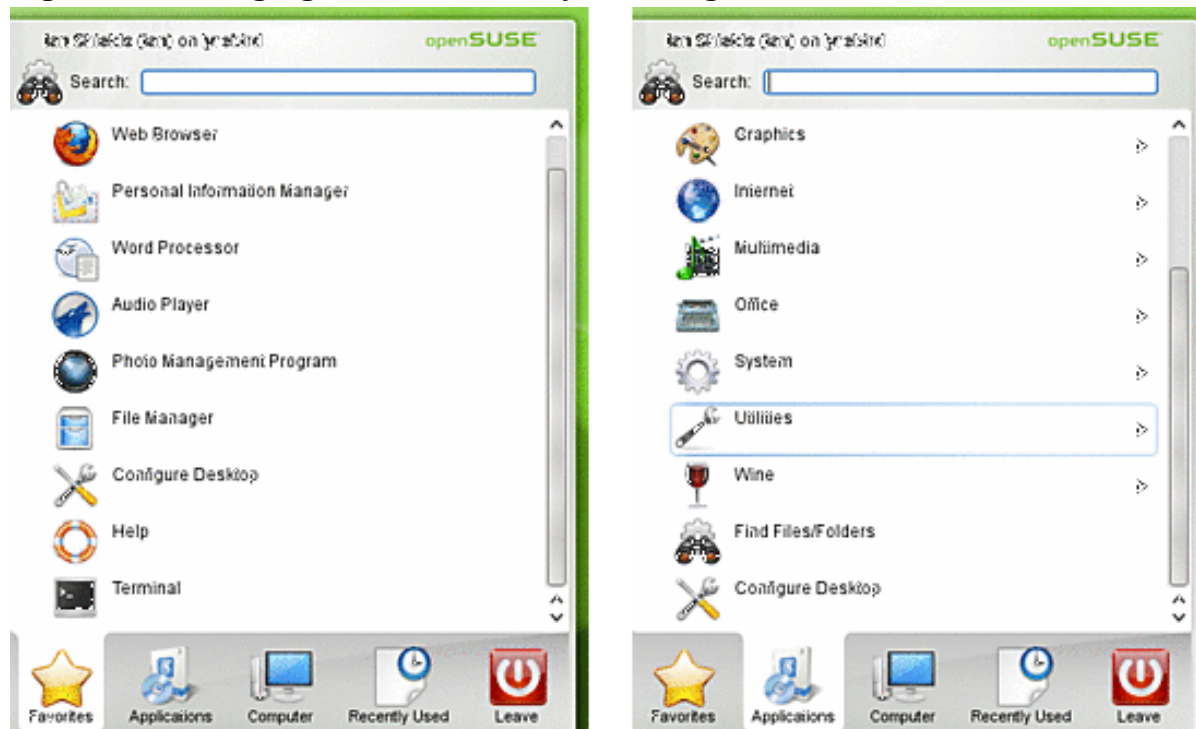
Figure 11. Accessing mouse settings in Fedora



In contrast, the KDE 4 desktop uses a different metaphor for the **Start** menu. Menu

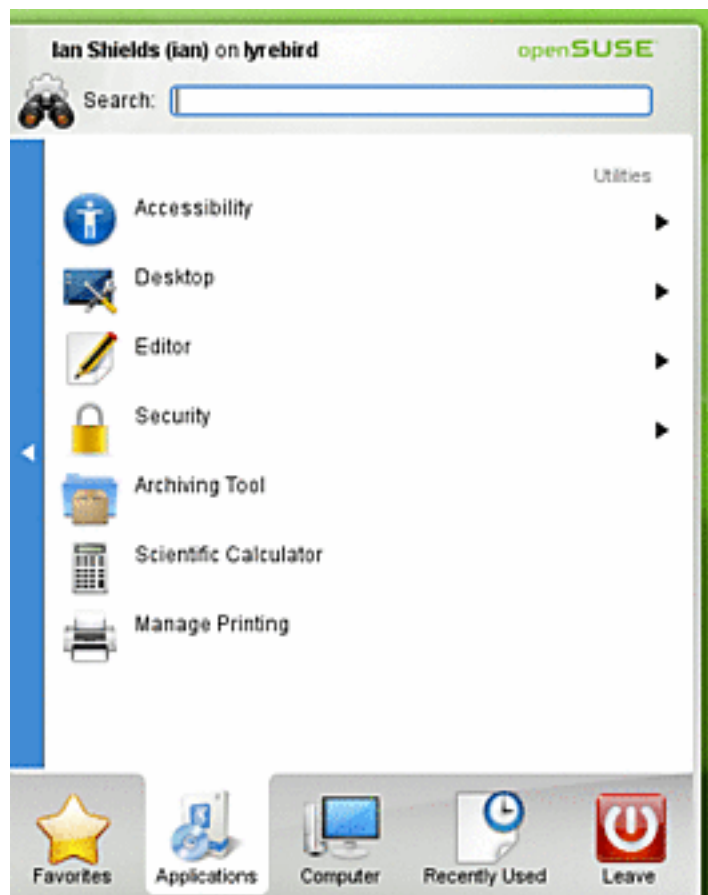
panels replace each other, and you navigate by clicking on items in the menu or by mousing over the icons at the bottom of the menu. Figure 12 illustrates the **Favorites** and **Applications** menus.

Figure 12. Changing KDE 4 menus by mousing over icons



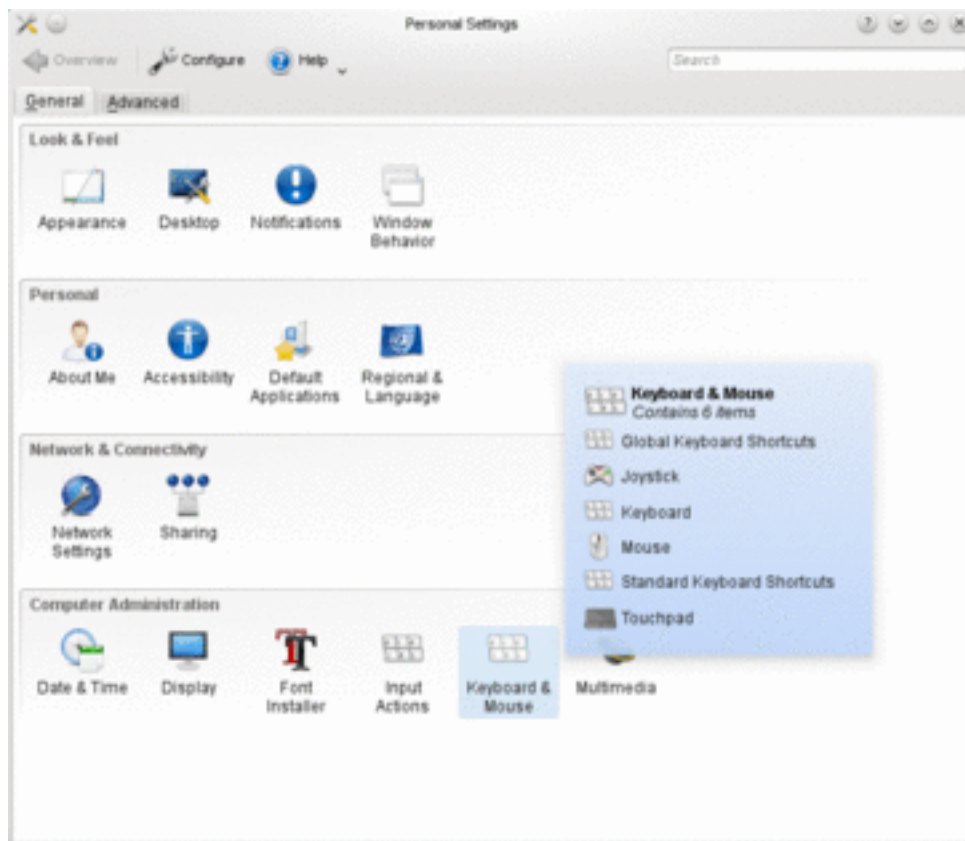
When submenus are selected, such as from **Start->Applications->Utilities**, a back button opens along the left of the menu so you can return to the previous menu level. Figure 13 illustrates this.

Figure 13. KDE 4 menu back button



Some **Start** menu items on OpenSUSE open a dialog box, possibly containing further selections. An example is the **Start->Applications->Configure Desktop** menu, which opens a window like that in Figure 14, where we show the hover help for the **Keyboard & Mouse** settings menu item.

Figure 14. KDE 4 Configure Desktop menu

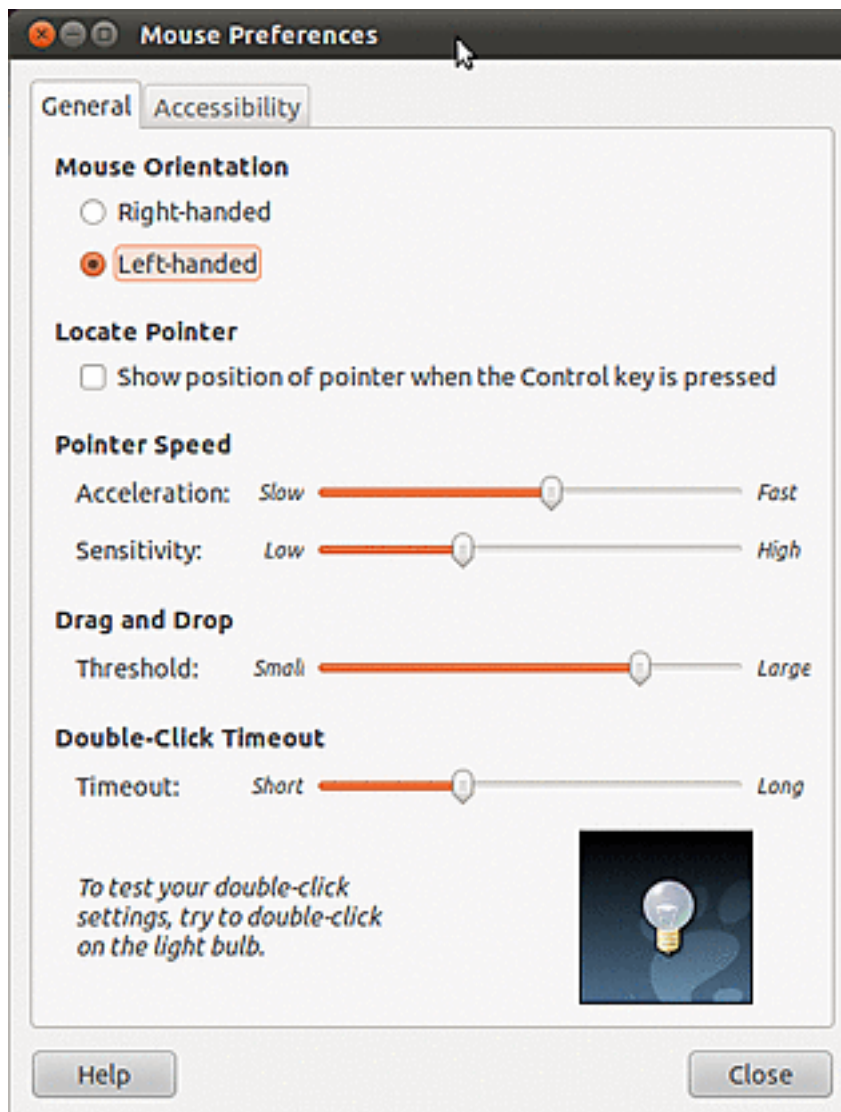


Switching to left-handed mouse usage

A right-handed user is generally assumed, but you can change your mouse configuration for left-handed use, along with many other desktop settings. Refer back to Figure 11 or Figure 14 to navigate to the mouse settings dialog.

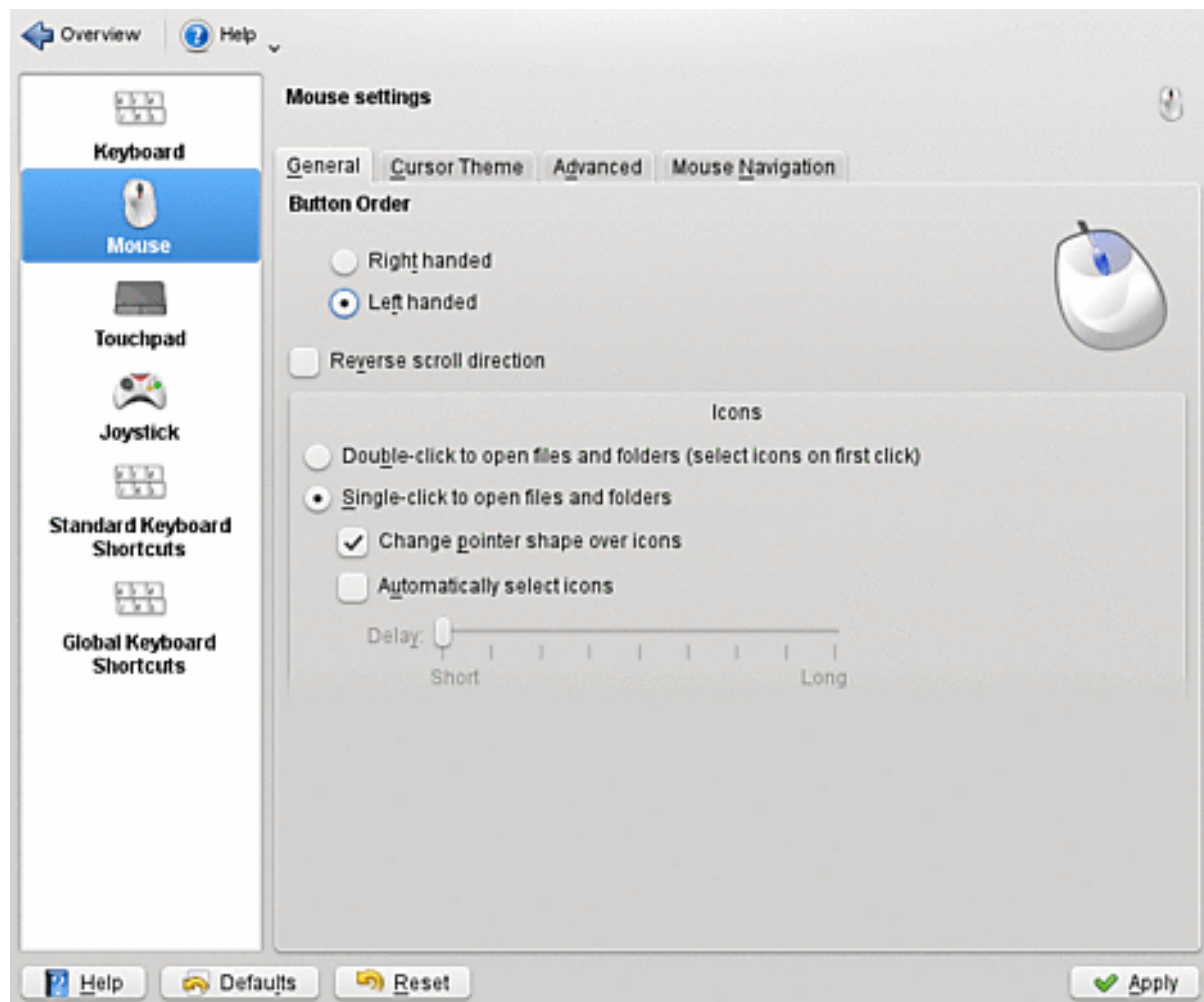
On an Ubuntu system, you should see a window similar to Figure 15 where you can change your mouse settings. In addition to basic left-handed or right-handed use, there are several other settings you can change and a tab of additional settings for accessibility. Settings take effect immediately, so once you click the left-handed choice, your mouse is set for left-handed use and you'll have to use the right mouse button as button 1 to close the dialog. The dialog on a Fedora system is similar.

Figure 15. GNOME dialog to change mouse settings



On an OpenSUSE system with KDE, you should see a window similar to Figure 16 where you can change your mouse settings. Note the mouse image in the dialog. If you switch to left-handed use, the right button in the mouse illustration will be highlighted to indicate that it is your primary selection button. Switch back to right-handed use, and the left button will be highlighted. As with the Ubuntu dialog, there are several other options that you can set. Unlike the Ubuntu dialog, the changes only take effect when you click the **Apply** button.

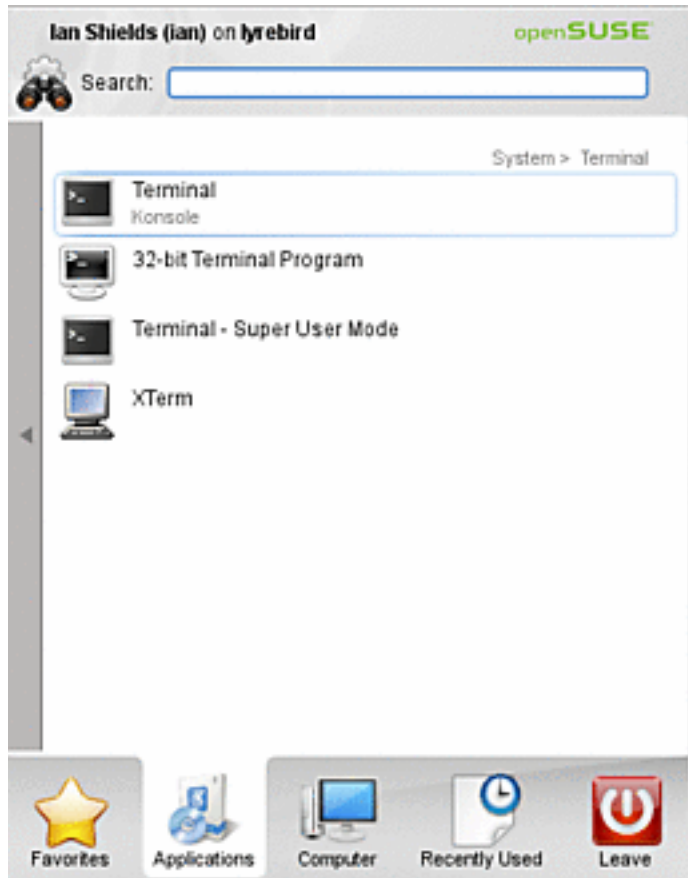
Figure 16. KDE dialog to change mouse settings



Section 4. Command line access

Sometimes on a Linux system you need to enter Linux commands rather than using a GUI. A Linux *Terminal* window is similar to a DOS command prompt under Windows. On our KDE desktop, you can reach a list of available terminal programs using **Start->Applications->System->Terminal** as shown in Figure 17.

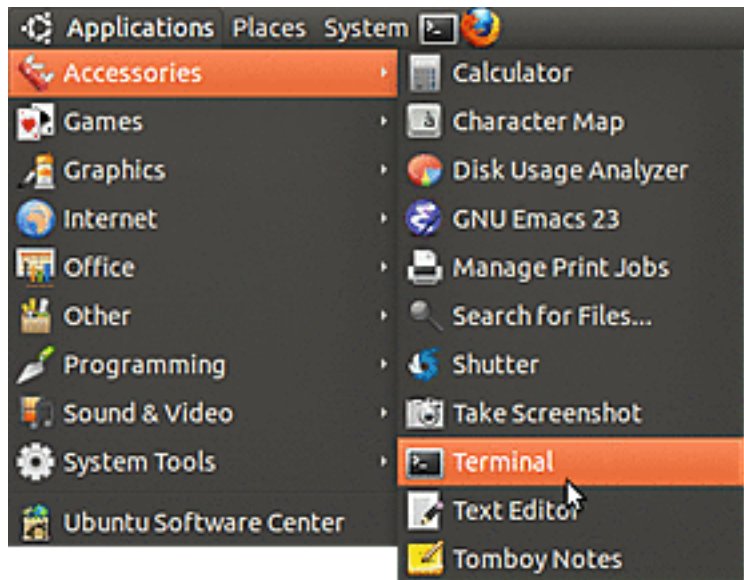
Figure 17. Opening a terminal on OpenSUSE with KDE 4



On the Ubuntu system, you can find a terminal window either under **Applications->Accessories->Terminal** as shown in Figure 18 or under **Applications->System->Terminal**. On Fedora systems, look under **Applications->System Tools->Terminal**.

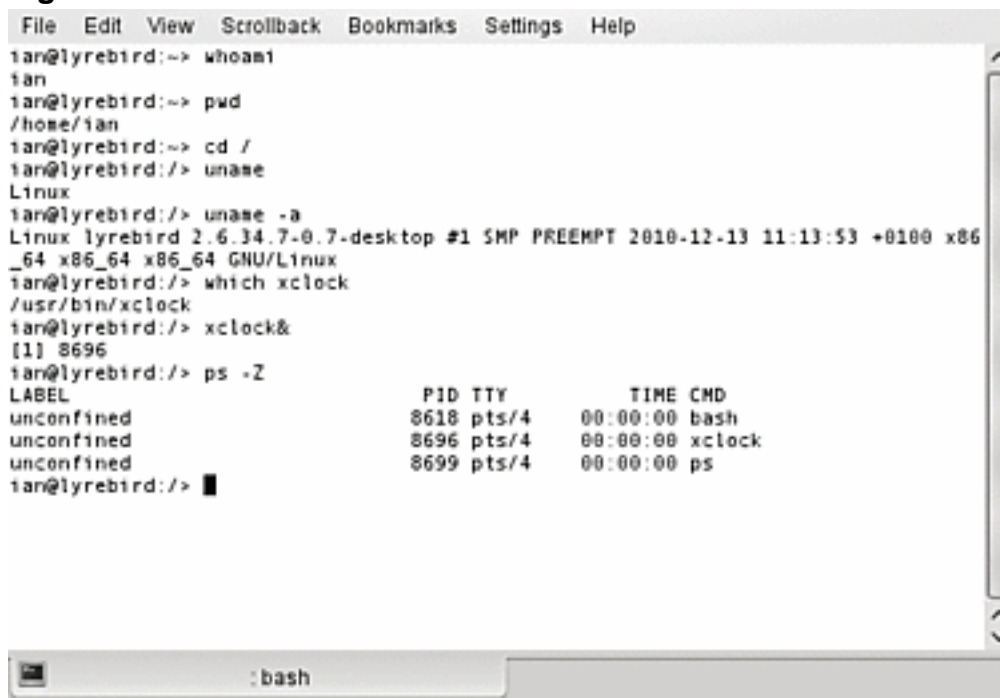
Note: If you right-click (or use the appropriate button if you reconfigured your mouse) the terminal menu choice, you will usually see an option to add this icon to your panel or desktop. Adding it to your panel gives you quick access to a terminal without going through the menus.

Figure 18. KDE shell icon



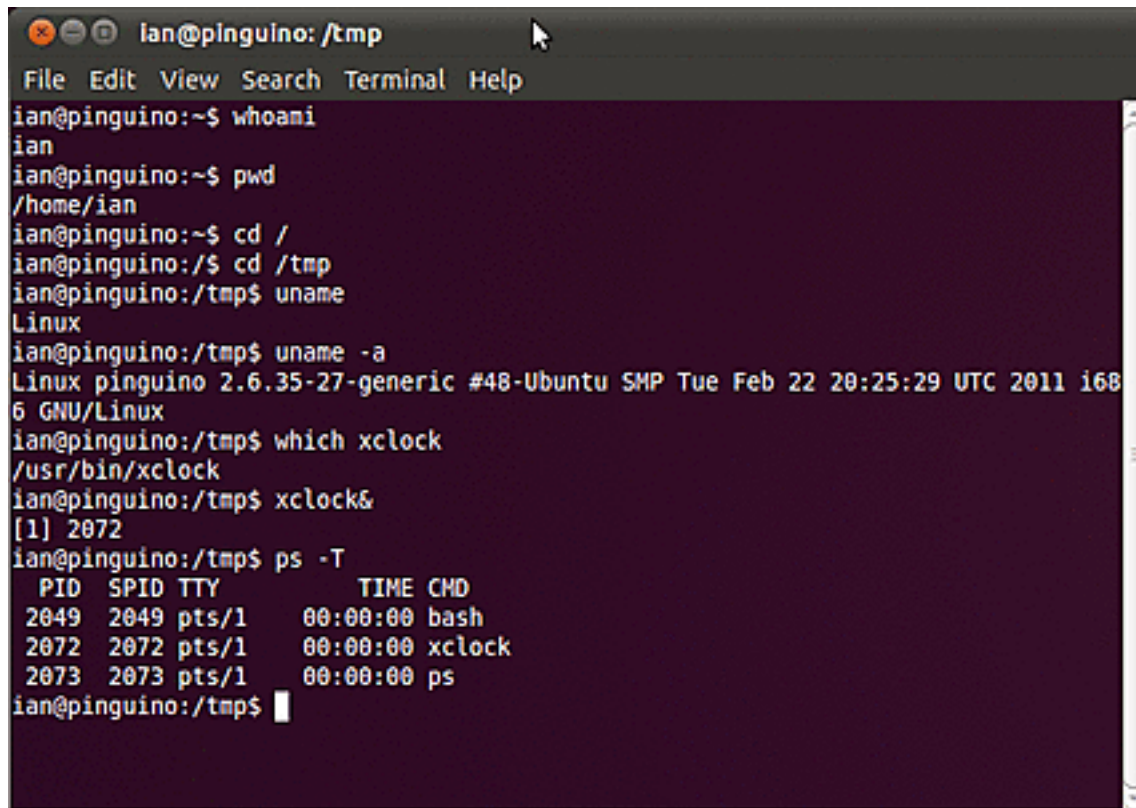
On the OpenSUSE system, select the Terminal (konsole) choice and you will get a terminal window something like Figure 19. In both Figure 19 and Figure 20, we've included some commands and their output that we'll discuss below.

Figure 19. KDE shell icon



The Ubuntu terminal window will look something like Figure 20.

Figure 20. KDE shell icon



```
ian@pinguino: /tmp
File Edit View Search Terminal Help
ian@pinguino:~$ whoami
ian
ian@pinguino:~$ pwd
/home/ian
ian@pinguino:~$ cd /
ian@pinguino:/$ cd /tmp
ian@pinguino:/tmp$ uname
Linux
ian@pinguino:/tmp$ uname -a
Linux pinguino 2.6.35-27-generic #48-Ubuntu SMP Tue Feb 22 20:25:29 UTC 2011 i686 GNU/Linux
ian@pinguino:/tmp$ which xclock
/usr/bin/xclock
ian@pinguino:/tmp$ xclock &
[1] 2072
ian@pinguino:/tmp$ ps -T
  PID  SPID  TTY      TIME CMD
  2049   2049 pts/1    00:00:00 bash
  2072   2072 pts/1    00:00:00 xclock
  2073   2073 pts/1    00:00:00 ps
ian@pinguino:/tmp$
```

The default appearance of your shell window depends on your distribution and your choice of desktop. To resize it, you can use the left mouse button to drag the corners or sides of the window. To scroll back through the most recent history, you can use the scroll bar. The command prompt typically ends with a \$ character for users other than `root`. Usually the command prompt will end with a # character indicating that the user of this shell is user `root` or has root authority. You can use the up arrow to recall previous commands and modify them if necessary. You will usually find a Settings or Terminal menu where you change things like window colors and fonts.

Figure 19 and Figure 20 show a few commands and their output:

whoami

Shows who is using this terminal window, `ian` in this case.

pwd

Prints the full name of the current working directory, which is `/home/ian` in this case. Note that the tilde (~) before the \$ in the command prompt shows that the user is currently in his or her home directory.

cd

Changes the current or working directory. We illustrate changing to the `/` (or `root`) directory and then to the `/tmp` directory, which is usually used for storing temporary files. Note that `/` is the root of the whole file system, and `/root` is the

home directory of the `root` user. Using `cd` without any directory name returns the user to the home directory, and using `cd -` returns you to the last directory you were in before the current one. Users other than `root` will normally have a home directory under `/home`. For example, `/home/ian` is my home directory on a system where my id is `ian`. Remember that tilde (`~`) is shorthand for the home directory of the current user. Add the name of a specific user to reference that user's home directory. For example, the home directory for user `ian` can also be referenced as `~ian`.

uname

Without parameters, shows the name of the operating system: `Linux`. With the `-a` parameter, displays additional information about your system.

which

Scans the directories in your `PATH` environment variable, and shows the full path to an executable program that would be executed if you typed the command at the shell prompt. In this case, we see that the `xclock` program would be run from `/usr/bin/xclock`. Note: This application is not always installed in a default Linux installation. It is usually part of a package with a name like `xorg-x11-apps`, so you may have to find and install the appropriate package to use it.

xclock

Launches a new window on your desktop with a clock. Note the trailing `&` on the command, which indicates that the command processor should return control to the terminal window rather than wait for the command to finish. Note also that this is the first such process spawned by this terminal window, and it has a process id (`PID`) of 1774.

ps

With the `-T` option, displays all processes started by this terminal. On some systems, the default display from the `ps` command includes a process status. See the man pages for details on all possible options and output columns. If the status were displayed in this example, you would see the `bash` shell program waiting for input (status `S` for sleeping) as would be the `xclock` process. The `ps` command is running and would have status `R` for runnable.

The output from these commands is shown in the two terminal windows above. A text form from the Ubuntu system is shown in Listing 1.

Listing 1. Ubuntu output from some basic commands

```
ian@pinguino:~$ whoami
ian
ian@pinguino:~$ pwd
/home/ian
ian@pinguino:~$ cd /
```

```
ian@pinguino:/$ cd /tmp
ian@pinguino:/tmp$ uname
Linux
ian@pinguino:/tmp$ uname -a
Linux pinguino 2.6.35-27-generic #48-Ubuntu SMP Tue Feb 22 20:25:29 UTC 2011
i686 GNU/Linux
ian@pinguino:/tmp$ which xclock
/usr/bin/xclock
ian@pinguino:/tmp$ xclock&
[1] 2072
ian@pinguino:/tmp$ ps -T
  PID  SPID  TTY      TIME  CMD
  2049   2049 pts/1    00:00:00 bash
  2072   2072 pts/1    00:00:00 xclock
  2073   2073 pts/1    00:00:00 ps
ian@pinguino:/tmp$
```

Some other commands that you might find useful include:

info *cmd_name*

Displays information about the command named *cmd_name*. Try `info info` to find out about the info documentation system.

man *cmd_name*

Is an interface to the online manual (man) pages about the command named *cmd_name*. Some information is in info format, while some is available only in man page format. Try `man man` to find out more about manual pages.

Section 5. Becoming superuser (or root)

For many tasks on Linux, you need *root* or *superuser* authority. The *root* user, sometimes called the *superuser*, is the user that is normally used for administrative tasks like configuring the system or installing software. **Use root only when you need to do administrative tasks; avoid using root for your normal work.** The root user can do anything, including accidentally destroying your system, which is usually not a good thing. Normal users have fewer privileges, and the system is much more protected from being inadvertently damaged by normal users.

Most administrative applications that have a graphical interface now ask for the root password before allowing non-root users access to the function. When you need to run commands from a terminal window as root, this doesn't help.

Your first thought may be to switch to another userid by logging out of the current userid and logging in as the new userid. But what if you only need to run a couple of quick commands as another user? Linux has a solution for you: The `su` (substitute user) and `sudo` commands allow you to temporarily run one or more commands as another user. This is often used for tasks that require root access. Indeed, if you

connect in remotely to a system using a terminal program such as `ssh` (or the very insecure `telnet`), then many Linux distributions will prevent you from signing in as root. This is a good security practice, and we encourage you to not try to circumvent it. Rather, you should sign in as a non-privileged user and then use the `su` or `sudo` command to do the work you need to do with root authority.

To summarize, there are two main ways to run an arbitrary command with root authority.

1. Use the `su` command, usually with the `-` option to become root.
2. Use the `sudo` command to execute a single command with root authority.

On systems such as Fedora or OpenSUSE, both methods are available, although using `su` is perhaps more common. On Debian-based systems such as Ubuntu, the security model prevents root login, so you can neither log in as root nor use `su` to become root, so you must use `sudo`.

Using su

Suppose you are logged in and looking at a terminal window, and you are not the root user but need to run a command, such as `fdisk`, which requires root authority. You switch to root using the `su` command alone, or, more commonly, add the `-` option.

The `su` command without the `-` option simply switches you to become root, but does not change your environment variables, including your path. The `-` option, which may also be typed as `-l` or `-login` if you really like typing extra letters, allows the login startup files for the substitute user to be read, thus setting things such as the path, environment, and prompt to those of the target user. Listing 2 shows examples of these two forms on our Fedora system. We've used the `pwd` (print working directory) command to show the current working directory in each case. Note how the prompts differ. If you'd like to understand more about how to customize your own prompts or what makes these prompts appear as they do, check out the ["Prompt magic" tip on developerWorks](#).

Listing 2. Switching to the root user

```
[ian@echidna ~]$ su
Password:
[root@echidna ian]# pwd
/home/ian
[root@echidna ian]# su -
[root@echidna ~]# pwd
/root
```

You will notice, not surprisingly, that you had to provide a password to switch to root. Once you have root authority, you can use `su` or `su -` to switch to another user or to switch to root with the login option. If you want to switch to a non-root user, just add the id. As before, you can use the `-` option or not, according to your needs. For example:

```
su - db2inst1
```

To return to the previous id, press `Ctrl-d` or type `exit` and press **Enter** if you are using the bash shell, which is the default on most Linux systems.

Now that we've learned how to use `su`, let's put it into practice with the `fdisk` command.

Listing 3. Running the `fdisk` command with `su`

```
[ian@echidna ~]$ fdisk /dev/sda
Unable to open /dev/sda
[ian@echidna ~]$ su -
Password:
[root@echidna ~]# fdisk /dev/sda

Command (m for help): m
Command action
  a   toggle a bootable flag
  b   edit bsd disklabel
  c   toggle the dos compatibility flag
  d   delete a partition
  l   list known partition types
  m   print this menu
  n   add a new partition
  o   create a new empty DOS partition table
  p   print the partition table
  q   quit without saving changes
  s   create a new empty Sun disklabel
  t   change a partition's system id
  u   change display/entry units
  v   verify the partition table
  w   write table to disk and exit
  x   extra functionality (experts only)

Command (m for help): q
[root@echidna ~]# exit
logout
[ian@echidna ~]$
```

Using `sudo`

Like the `su` command, the `sudo` command allows you to run commands with the authority of another user. The commands that a given user or class of users may execute are listed in the `/etc/sudoers` file. In contrast to the `su` command, you do **not** need to know the password of the root, or other user, although you will need to provide your own password. The `/etc/sudoers` file is maintained by root and can be

edited using the `visudo` command.

Usually, if you are executing multiple `sudo` commands in rapid succession, you will not need to reenter your password for each one. An alternative is to run `sudo` with the `-s` option, which runs a shell for you, from which you can run many commands as the target user until you close the shell. Listing 4 illustrates both of these methods.

Listing 4. Using the `sudo` command on Ubuntu

```
ian@pinguino:~$ fdisk /dev/sda

Unable to open /dev/sda
ian@pinguino:~$ sudo fdisk /dev/sda
[sudo] password for ian:

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 120.0 GB, 120034123776 bytes
255 heads, 63 sectors/track, 14593 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x54085408

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1          2611     20972826   7   HPFS/NTFS
/dev/sda2             2612          2624       104422+  83   Linux
/dev/sda3             2625          14593     96140962   5   Extended
/dev/sda5             2625          2689        522081  82   Linux swap / Solaris
/dev/sda6             2690          5180     20008926  83   Linux
/dev/sda7             5181          9341     33423201  83   Linux
/dev/sda8             9342          14593     42186658+  83   Linux

Command (m for help): q

ian@pinguino:~$ sudo -s
root@pinguino:~# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

Command (m for help): q

root@pinguino:~# exit
ian@pinguino:~$
```

If you are not authorized in the `sudoers` file, you will receive an error message similar to that in Listing 5.

Listing 5. Attempting to use `sudo` without authority

```
[ian@echidna ~]$ sudo fdisk /dev/sda
[sudo] password for ian:
ian is not in the sudoers file. This incident will be reported.
```

Section 6. Using a GUI application as another user

You may have noticed in the discussion of the `su` command in the [previous section](#) that we ran only commands that displayed output in the terminal window. Usually you will be able to run GUI commands too. For example, some installation programs require you to have root authority to install a program and have a GUI installer. If you find you cannot start GUI applications as another user, then read on, as you may have to take additional steps on some distributions in order to run GUI applications as another user.

Note: Recent distributions often let you have multiple desktops open at once and switch between them using a key sequence such as Ctrl-Alt-F7 or Ctrl-Alt-F8. Depending on what you need to do, this may be another alternative. See the section on [Logout](#) for details on this menu option.

GUI applications on Linux use the X Window System, a client-server system designed to allow multiple users to access a computer across a network using windowed applications. An X display is known by a name of the form *hostname:displaynumber.screennumber*. For Linux running on a workstation such as a PC, there is typically only one display with a single screen. In this case, the *displayname* may be, and usually is, omitted so the display is known as :0.0, or sometimes just :0.

The X Window System server needs to know the display and also whether you are authorized to connect to the server. Authorization is most commonly done using the MIT-MAGIC-COOKIE-1, which is a long random string that is regenerated whenever the server is reset. So that applications can pass this information to the X server, you will have `DISPLAY` and `XAUTHORITY` variables set in your environment. The `XAUTHORITY` variable will point to a file that is usually only able to be read or written by the owning user as a security precaution. We assume you are using a graphical login if you are reading this, so your startup should have already set these for you. The example in Listing 6 is from our Ubuntu system and shows the values of the `DISPLAY` and `XAUTHORITY` variables as well as the ownership for the file pointed to by the `XAUTHORITY` variable.

Listing 6. DISPLAY and XAUTHORITY

```
ian@pinguino:~$ echo $DISPLAY
:0.0
ian@pinguino:~$ echo $XAUTHORITY
/var/run/gdm/auth-for-ian-WoeKHn/database
ian@pinguino:~$ ls -l $XAUTHORITY
-rw----- 1 ian ian 53 2011-04-01 16:24 /var/run/gdm/auth-for-ian-WoeKHn/database
```

Now we attempt to use `sudo` to run the `xclock` command as user `editor`, again on our Ubuntu system. As Listing 7 shows, the values for `DISPLAY` and `XAUTHORITY` variables are the same as for user `ian`, but the `xclock` command fails.

Listing 7. `DISPLAY` and `XAUTHORITY` with `sudo`

```
ian@pinguino:~$ sudo -u editor echo $DISPLAY
[sudo] password for ian:
:0.0
ian@pinguino:~$ sudo -u editor echo $XAUTHORITY
/var/run/gdm/auth-for-ian-WoeKHn/database
ian@pinguino:~$ sudo -u editor xclock
No protocol specified
Error: Can't open display: :0.0
```

In this case, user `editor` has the `XAUTHORITY` variable set to `/var/run/gdm/auth-for-ian-WoeKHn/database`, but we already saw that the permissions on that file only allow user `ian` to read or write it. The variable may as well not be set if user `editor` cannot read the file it points to. Before we look at how to address this issue, let's look at what happens if we unset either the `DISPLAY` or `XAUTHORITY` variables for user `ian`. We'll do this by running the `xclock` command with an environment variable modified using the `env` command's `-u` option to unset an environment variable before running `xclock`. Our results are in Listing 8.

Listing 8. Unsetting `DISPLAY` and `XAUTHORITY`

```
ian@pinguino:~$ env -u DISPLAY xclock
Error: Can't open display:
ian@pinguino:~$ env -u XAUTHORITY xclock
```

You may be surprised to see that the `xclock` command runs, even though we unset the `XAUTHORITY` environment variable.

So far, we have mentioned the MIT-MAGIC-COOKIE-1 security method. If the token is not provided, the X server will also check a list of authorized hosts. Use the `xhost` command to display or update the list. Use the `+` option to add entries and the `-` option to remove entries. Use the special *family* entry `local:` (note the `:`) to allow access to the display by any local user on the system. Since you are a single user system, this means that you can `su` to an arbitrary non-root user and can now launch `xclock` or other X applications. We illustrate the use of the `xhost` command in Listing 9.

Listing 9. Using `xhost`

```

ian@pinguino:~$ xhost
access control enabled, only authorized clients can connect
SI:localuser:ian
ian@pinguino:~$ xhost +local:
non-network local connections being added to access control list
ian@pinguino:~$ xhost
access control enabled, only authorized clients can connect
LOCAL:
SI:localuser:ian
ian@pinguino:~$ sudo -u editor xclock
ian@pinguino:~$ # Close the xclock window to return here
ian@pinguino:~$ xhost -local:
non-network local connections being removed from access control list
ian@pinguino:~$ xhost
access control enabled, only authorized clients can connect
SI:localuser:ian

```

On a single user system, enabling the display for use by all local users is usually a reasonable and simple solution. If you need to be more restrictive, use the `xauth` to extract the cookie from your `XAUTHORITY` file and give it to the user who needs access to the display. In Listing 10 we perform the following tasks:

1. Use `xauth` as user `ian` to display the cookie in a format that can be emailed or otherwise sent to the desired other user.
2. Use `sudo -s` and switch to user `editor` to run several commands.
3. Use `xauth` to create a new authorization file. Note that we used `echo` to pipe the data to `stdin` while splitting the command over several lines by using a trailing backslash (`\`).
4. We export a new value for the `XAUTHORITY` variable so it now points to our newly created authorization file.
5. Finally, we run the `xclock` command using a trailing ampersand (`&`) to run it in the background and retain control of our terminal window.

Listing 10. Using `xauth`

```

ian@pinguino:~$ xauth -f $XAUTHORITY nextract - :0
0100 0008 70696e6775696e6f 0001 30 0012 4d49542d4d414749432d434f4f4b49452d31 0010 3c4bc87
c2ce4ce5e97f8199c213b4ec9
ian@pinguino:~$ sudo -s -u editor
editor@pinguino:~$ echo "0100 0008 70696e6775696e6f 0001 30 0012"\
> " 4d49542d4d414749432d434f4f4b49452d31"\
> " 0010 3c4bc87c2ce4ce5e97f8199c213b4ec9" |
> xauth -f ~editor/temp-xauth nmerge -
xauth: creating new authority file /home/editor/temp-xauth
editor@pinguino:~$ export XAUTHORITY=~editor/temp-xauth
editor@pinguino:~$ xclock&
[1] 4827

```

This brief introduction will probably get you started with running X applications as

another user. Although we have used Ubuntu as an example, the basic principles we have demonstrated here apply to all distributions. For more details on using the `xauth` and `xhost` commands, you can use any of these commands as appropriate to view the online manual pages:

- `info xauth`
 - `man xauth`
 - `info xhost`
 - `man xhost`
-

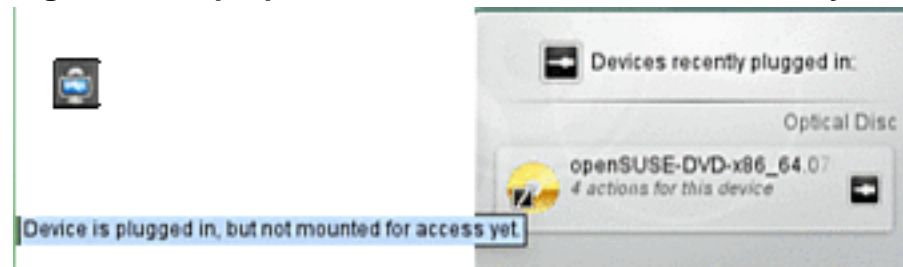
Section 7. Removable devices

On Linux and UNIX® systems, all files are accessed as part of a single large tree that is rooted at `/`. To access the files on a CD-ROM, you need to mount the CD-ROM device at some mount point in the file tree. On current distributions, this is usually automated for you; you need only insert the disc and it will be recognized and mounted. Once mounted, it is important to properly unmount the device to avoid data loss.

Mounting removable devices

When you insert a CD or DVD disc into a SUSE 11.3 system, or attach a USB drive, you will see a pop-up window similar to Figure 21. If you miss the pop-up before it closes, you can use the panel icon that we've shown in the top left of the figure to open it again. Hovering over the icon shows the tooltip indicating that the device is not yet mounted.

Figure 21. Pop-up when CD or DVD loaded on SUSE system



Hovering over the plug icon to the right of the image reveals a tooltip saying "Click to access this device from other applications". Clicking over the text "4 actions for this device" expands the image to look like Figure 22. The available actions may differ on

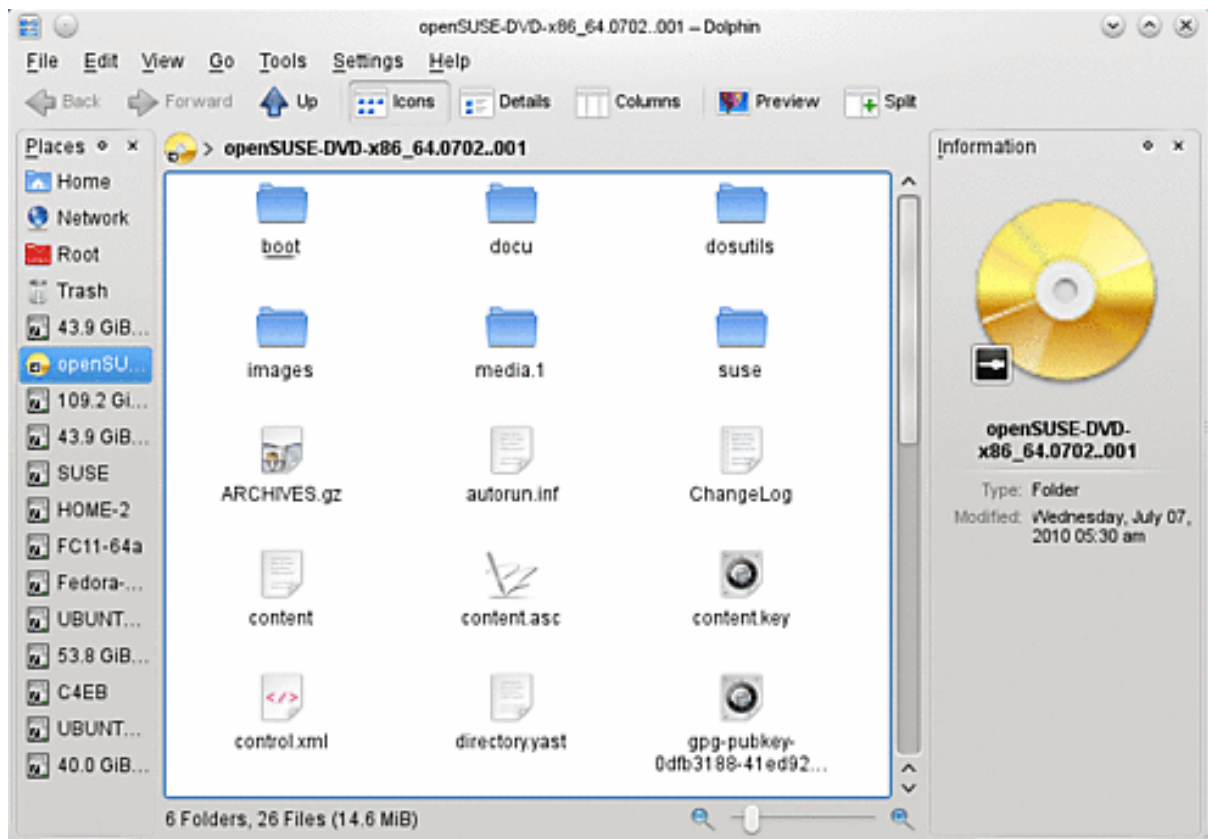
your system if you have installed different software packages.

Figure 22. Available actions for CD or DVD



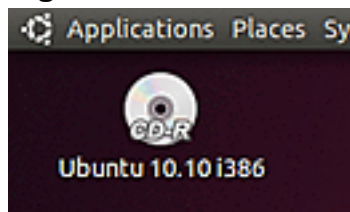
If you click the "Open with File Manager" selection, you will see a window something like Figure 23. From this window you can navigate around the DVD, open files, or run applications. Hover over an item to see a description in the right part of the window.

Figure 23. SUSE File Manager



On Ubuntu and Fedora systems, the default action on inserting a disc is slightly different. Usually, an icon similar to that shown in Figure 24 will appear on your desktop. The file manager (Nautilus) may also open automatically. If it does not, you can double-click the icon to open the file manager. Depending on your system, you may also be prompted as to whether to run the `autorun` file in the root of the disc or not.

Figure 24. Inserted CD on Ubuntu



If you clicked to access the device from other applications (OpenSUSE), or opened it with File Manager or another application on any distribution, then the device will have been mounted for you. It will usually be mounted in the `/media` directory and will probably use the disc label as a mount point. You can verify by opening a terminal window and running the `mount` command as shown for our OpenSUSE system in Listing 11. Other systems are very similar, although the mount options may differ slightly.

Listing 11. Using the mount command

```
ian@lyrebird:~> mount
/dev/sdb12 on / type ext4 (rw,acl,user_xattr)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
...
/dev/sr0 on /media/openSUSE-DVD-x86_64.0702..001 type iso9660 (ro,nosuid,nodev,uid=1000,utf8)
```

In this case our disc is mounted at `/media/openSUSE-DVD-x86_64.0702..001` and you can explore it or change to a directory on the disc from the terminal window. Note that the CD/DVD device is `/dev/sr0`. Devices on a Linux system also appear in the filesystem tree. To put this another way, the filesystem that is on the media in the device `/dev/sr0` is accessible to applications starting at the mount point `/media/openSUSE-DVD-x86_64.0702..001`. If you'd like to learn more about how devices are mounted, see our article [Learn Linux, 101: Control mounting and unmounting of filesystems](#).

Safely removing devices

Once a DVD or CD is mounted, you can use the files on the disk as they are now part of your file system. While a CD-ROM is mounted, Linux will lock the CD so that it cannot be ejected with the Eject button. Of course, this doesn't stop you from unplugging an external CD or USB drive, pulling a USB stick out of the USB socket, or ejecting a floppy disk. To avoid potential data loss, you should always remove the device safely by first *unmounting* it.

You can unmount devices from the graphical desktop, or from the command line. The `umount` command (note `umount` without an 'n') unmounts a device, and the `eject` command first unmounts a device and then tries to eject it, for example by opening a CD drawer. Traditionally, mounting and unmounting devices required root authority. Nowadays, it is common to allow *user mounting*, so that a non-root user is able to mount and unmount devices. You may have noticed back in Listing 11 the option `uid=1000`. The uid of 1000 corresponds to user `ian` on this system as shown in Listing 12.

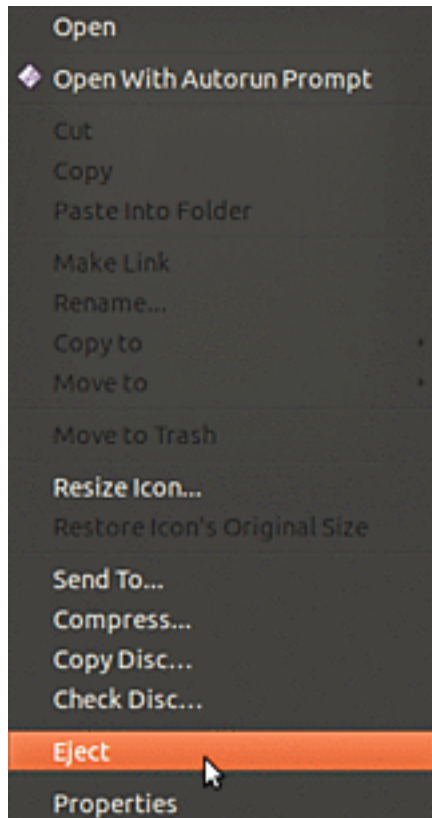
Listing 12. Using the id command for the current user

```
ian@lyrebird:~> id
uid=1000(ian) gid=100(users) groups=33(video),100(users)
```

If you use the graphical desktop tools to unmount the device, you may only need to worry about ids if you have logged out and then back in as a different user. So let's look at the graphical tools, then the command line ones.

On Ubuntu or Fedora, if you right click on the icon for the removable media you will see a context menu similar to that in Figure 25. You will have an option to either unmount or eject the device. Select the eject option for a CD or DVD and the device will be unmounted, the icon will disappear from your desktop, and your drawer will open. For a USB drive, the option may be unmount rather than eject, and you can safely unplug the device after it is unmounted. If you use an external hard drive with multiple partitions, you need to make sure that all partitions are unmounted before removing or powering off the drive.

Figure 25. SUSE File Manager



In Figure 26 we show our OpenSUSE system with an attached USB drive as well as the DVD we had used earlier. The DVD and one partition of the hard drive are mounted. OpenSUSE distinguishes the mounted from the unmounted by changing the plug icon (when a device is not mounted) to an eject button (when it is mounted). You will also note that the small square at the bottom left of the device icon also changes from having a diagonal line to having an asterisk. Note that for writeable media, such as our hard drive, the available space is also displayed.

Figure 26. OpenSUSE File Manager



To unmount or eject a mounted drive, simply click the eject button (shown above for the 2006R1 partition). If you eject a device such as a CD or DVD where a drawer opens or some other similar physical disconnection takes place, then the device will be removed from your list of plugged devices.

Safe removal from the command line

We mentioned the `umount` and `eject` commands earlier. Let's now look at how to use them and the `lsdf` that you may also want to know about. You will find some differences between systems in this area, so be prepared for things not to be exactly as we illustrate here.

To begin, we'll look at using the `umount` command to unmount the CD on our Ubuntu system as shown in Listing 13. We first use `grep` to filter out the entries from the output of `mount` to show only those entries that contain 'media', the usual mount point for removable media. We then use `umount` to unmount the device using its mount point (`/media/Ubuntu 10.10 i386`). Finally, we repeat the `mount + grep` filter to confirm that the device is no longer mounted. Note that you can use either the mount point or the device name (`/dev/sr0` in this case) as the argument to `umount`.

Listing 13. Unmounting a CD on Ubuntu from the command line

```
ian@pinguino:~$ mount | grep media
```



```
/dev/sr0 on /media/Ubuntu 10.10 i386 type iso9660 (ro,nosuid,nodev,uhelper=udisks,
uid=1000,gid=1000,iocharset=utf8,mode=0400,dmode=0500)
ian@pinguino:~$ umount /media/Ubuntu\ 10.10\ i386
ian@pinguino:~$ mount | grep media
```

As you can see, Ubuntu allows you to use the command line to unmount a drive that was automatically mounted. Repeating the above scenario on OpenSUSE 11.3 is likely to result in something like Listing 14.

Listing 14. Unmounting a CD on OpenSUSE from the command line (1)

```
ian@lyrebird:~> mount | grep media
/dev/sr0 on /media/opensUSE-DVD-x86_64.0702..001 type iso9660 (ro,nosuid,nodev,uid=1000,
utf8)
/dev/sdc6 on /media/2006R1 type ext3 (rw,nosuid,nodev)
ian@lyrebird:~> umount /media/opensUSE-DVD-x86_64.0702..001/
umount: /media/opensUSE-DVD-x86_64.0702..001 is not in the fstab (and you are not root)
ian@lyrebird:~> umount /dev/sr0
umount: /dev/sr0 is not in the fstab (and you are not root)
```

Sometimes you will find a disconnect between what you can do graphically and what you might expect to do at a command line. As Linux matures, such disconnects become less frequent, but you will sometimes find them as we have done here. The obvious solution, given what you have already learned in this tutorial, is to use `su` or `sudo` to run the required command with root privileges. So let's try it using `su` as shown in Listing 15.

Listing 15. Unmounting a CD on OpenSUSE from the command line (2)

```
ian@lyrebird:~> su -
Password:
lyrebird:~ # mount | grep media
/dev/sr0 on /media/opensUSE-DVD-x86_64.0702..001 type iso9660 (ro,nosuid,nodev,uid=1000,
utf8)
/dev/sdc6 on /media/2006R1 type ext3 (rw,nosuid,nodev)
lyrebird:~ # umount /dev/sdc6
lyrebird:~ # umount /media/opensUSE-DVD-x86_64.0702..001/
umount: /media/opensUSE-DVD-x86_64.0702..001: device is busy.
(In some cases useful info about processes that use
the device is found by lsof(8) or fuser(1))
```

We were able to unmount `/dev/sdc6` successfully, but were unable to unmount `/media/opensUSE-DVD-x86_64.0702..001` because Linux claimed it was busy. We would get a similar error message if we tried to use the `eject` command. Remember that we said that Linux will lock a CD or DVD closed while it is in use. More generally, you cannot unmount a filesystem if it is in use by some other user. As suggested in the error message above, you can use the `lsof` or `fuser` command to find which user is causing your unmount problem. Typical usage is illustrated in Listing 16.

Listing 16. Using `lsof` and `fuser` (as root)

```

lyrebird:~ # lsdf /media/openSUSE-DVD-x86_64.0702..001/
COMMAND PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
bash    3824  ian   cwd   DIR   11,0      2048 2048 /media/openSUSE-DVD-x86_64.0702..001/
boot
lyrebird:~ # fuser -um /media/openSUSE-DVD-x86_64.0702..001/
/media/openSUSE-DVD-x86_64.0702..001: 3824c(ian)

```

The `lsdf` command shows the actual file or directory (`/media/openSUSE-DVD-x86_64.0702..001/boot` in this case) that is open as well as the process id (3824) and user id (ian) using the file or directory. The `fuser` command shows the user and process id using the `/media/openSUSE-DVD-x86_64.0702..001` filesystem, but not the specific open file or directory.

Normally you would use this information to cleanly close the open process or window using the filesystem. On some Linux systems there is also an `-L` option to `umount` that allows a *lazy* unmount where the filesystem is detached from the filesystem hierarchy immediately, and all references to the filesystem are cleaned up when the filesystem is no longer busy. On OpenSUSE 11.3 this option is not available, so you need to clean up the filesystem references, either by terminating the offending processes or by ensuring they are no longer using affected resources, before unmounting.

We mentioned back in [Listing 14](#) that the OpenSUSE system does not support command line use of `umount` for a device that was mounted from the desktop, so we attacked the problem by using root authority. There is another approach using the `halmount` command to unmount the device. *HAL* is a Hardware Abstraction Layer, which provides a consistent application interface to various hardware devices. If you use `halmount`, you will still be unable to unmount a busy device. Listing 17 shows how to use `halmount` and then switch to `root` to forcibly terminate the process that is keeping the disc busy. Note that killing the process in this way may cause data loss, so we do not recommend it unless you are sure you will not lose data. We then switch back to normal user mode and use `halmount` to unmount the filesystem that is no longer busy. Finally we use `eject` to open the drawer and eject the DVD.

Listing 17. Using `halmount`, `kill`, and `eject`

```

ian@lyrebird:~> halmount -u /dev/sr0
/dev/sr0: org.freedesktop.Hal.Device.Volume.Busy: umount: /media/openSUSE-DVD-x86_64.0702..001: device is busy.
ian@lyrebird:~> lsdf /dev/sr0
COMMAND PID USER  FD  TYPE DEVICE SIZE/OFF NODE NAME
bash    3824  ian   cwd   DIR   11,0      2048 2048 /media/openSUSE-DVD-x86_64.0702..001/
boot
ian@lyrebird:~> su -
Password:
lyrebird:~ # kill -9 3824
lyrebird:~ # ps -ef | grep 3824

```

```
root      5542  5507  0 22:47 pts/2    00:00:00 grep 3824
lyrebird:~ # logout
ian@lyrebird:~> halmount -u /dev/sr0
ian@lyrebird:~> eject /dev/sr0
```

We hope this brief introduction to removable media use on Linux will help you enjoy your removable media files and avoid data loss.

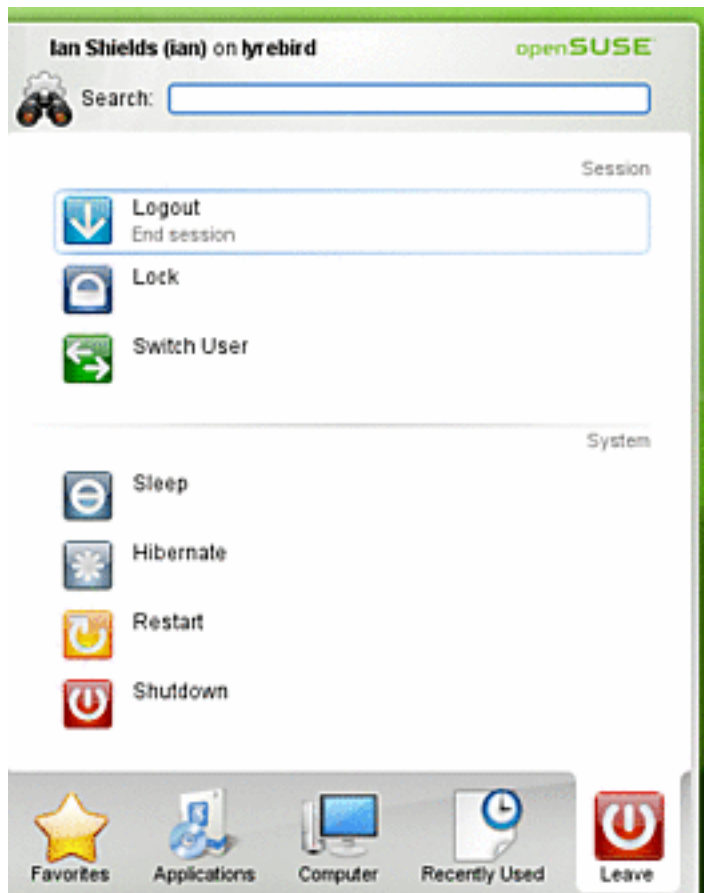
Section 8. Logout

The tasks of logging out, locking your screen, shutting down, and restarting the system are usually related in the sense that they are accessed from the same or similar menus. Frequently your distribution will add a quick launch button to access these tasks. If not, you can [add one yourself](#).

OpenSUSE logout

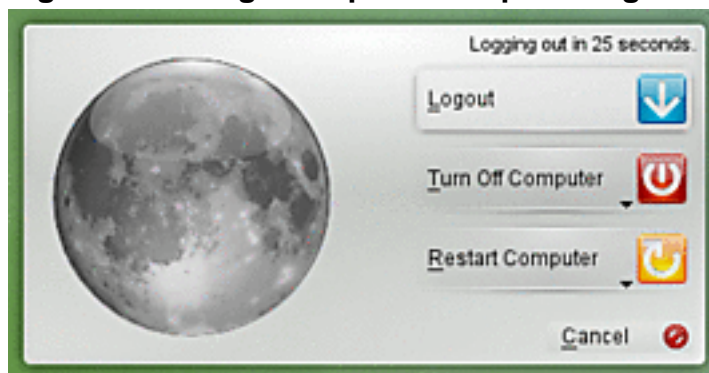
To log out, click **Start** and then **Leave**. You will see a menu like Figure 27 with options for logging out, locking the screen, and shutting down or restarting the system. Click the appropriate selection.

Figure 27. Adding a logout button with Red Hat enterprise Linux 3



You will also see icons for leave and lock at the bottom right of the OpenSUSE panel. Click the **Leave** icon on the right, and you will see a dialog box like Figure 28. Again, make your selection.

Figure 28. Using the OpenSUSE panel logout icon

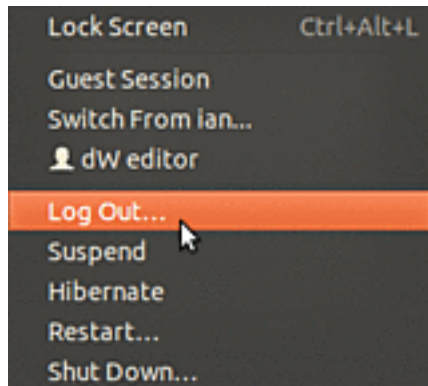


Ubuntu

You access Ubuntu logout and restart functions from a panel icon at the right end of the top Ubuntu panel. This icon is similar to the **Leave** icon on OpenSUSE. Click it to

reveal a dialog similar to Figure 29 where you can make your choice.

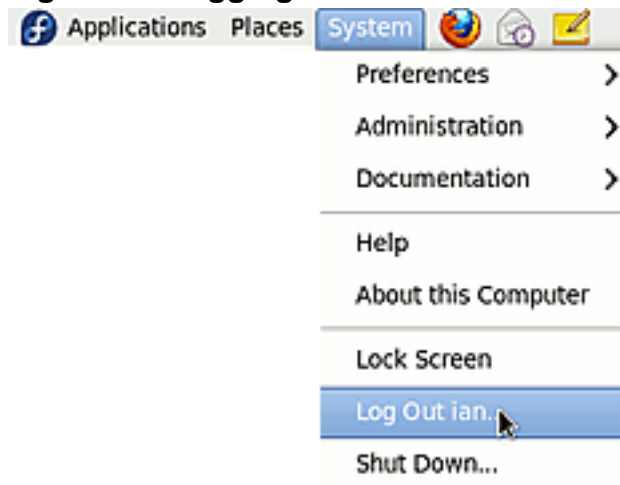
Figure 29. Using the Ubuntu panel logout icon



Fedora logout

Access the Fedora logout and restart functions from the **System** menu on the top panel as shown in Figure 30. Your choices are similar to the OpenSUSE and Ubuntu choices that you have seen above.

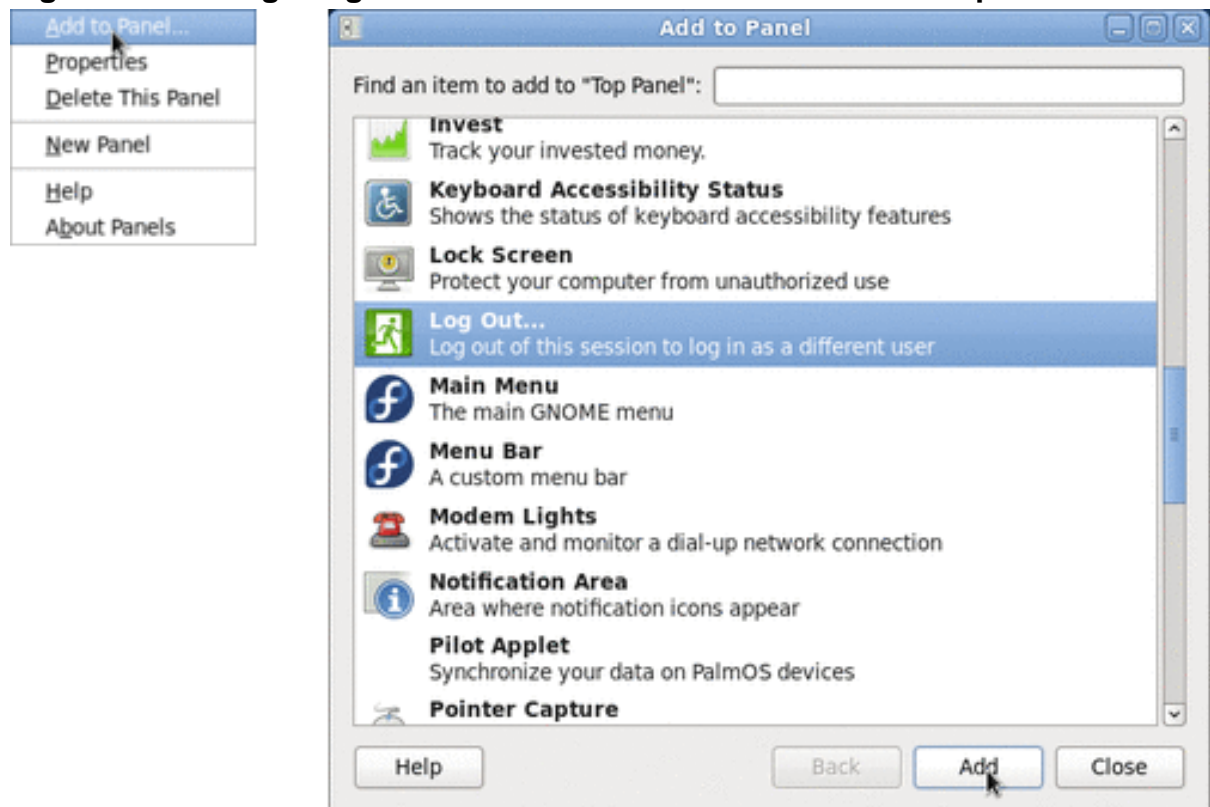
Figure 30. Logging out of the Fedora GNOME desktop



Adding a logout button to your panel

Fedora does not install a panel icon for logout or screen lock similar to those on OpenSUSE or Ubuntu. However, you can add your own quite easily.

Start by right-clicking a blank part of the panel. From the context menu, select **Add to Panel...**, and you will see a selection of items that you may add. Scroll down to the **Log Out** selection and click on it to add it to your panel. You can add the **Lock Screen** icon in the same manner.

Figure 31. Adding a logout button to the Fedora GNOME desktop

A logout icon will be added to your panel.

If an application is not in the list, and you can start it from the **Applications** menu or you can open the menu to the point where you would start the application, then right-click to open a context menu that includes choices to add a launcher to the panel or to your desktop.

Section 9. Adding users and groups

Sometimes you will need to define new users of your system and new groups for those users. For example, you may need to define a user called **mqm** and a group also called **mqm** (as well as another group called **mqbrkrs** when installing the embedded messaging component in WebSphere Application Server), or you may need to create users to administer databases.

If you do not specify a user number, graphical tools will usually assign the next available user number. For Fedora and Red Hat systems, user numbers start at 500, so the user you created when you installed your system is probably user 500. For

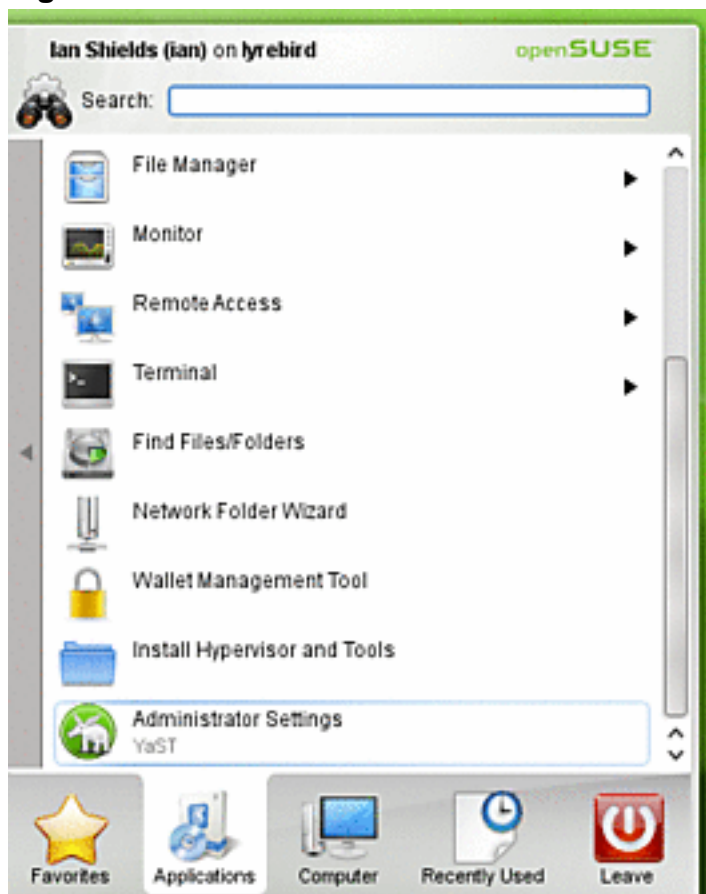
OpenSUSE and Ubuntu, the numbers start at 1000. If you use the same id on several systems as I do, you will probably find it convenient to use the same id and group numbers on each system too.

For the purpose of this section, we will add a user called `testuser` with id 2000 and group 2000. Usually you will define the group first and then define the users who will use the group, so that's what we will do here. You can either use the graphical tools for user administration or enter commands in a terminal window. We'll give an overview of the graphical process here using [OpenSUSE's system administration tools](#). Then we'll tell you where to find the corresponding [tools on a Fedora or Ubuntu system](#). Finally we'll give you the [commands](#) if you really want to do it from the command line.

Adding users and groups to your OpenSUSE system

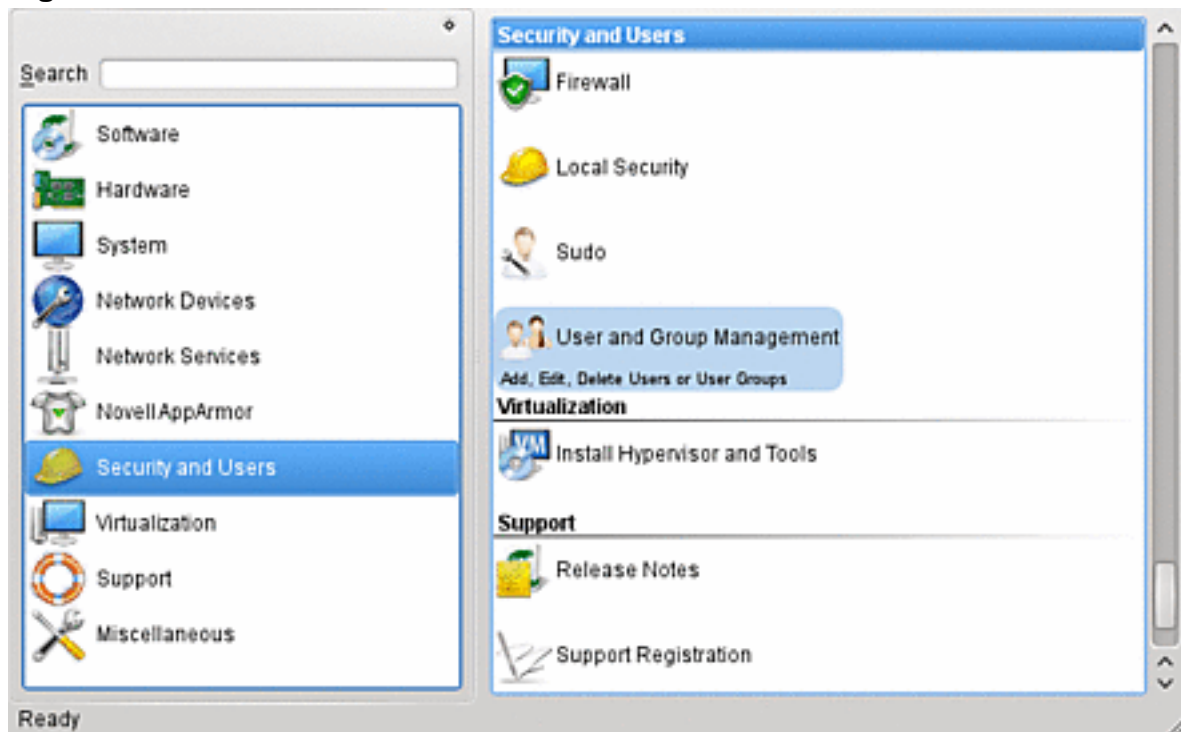
On an OpenSUSE system with KDE, you access the YaST (Yet Another System Tool) control center using **Start->Applications**, then select **System** and scroll down to **Administrator Settings** as shown in Figure 32.

Figure 32. YaST2 Control Center



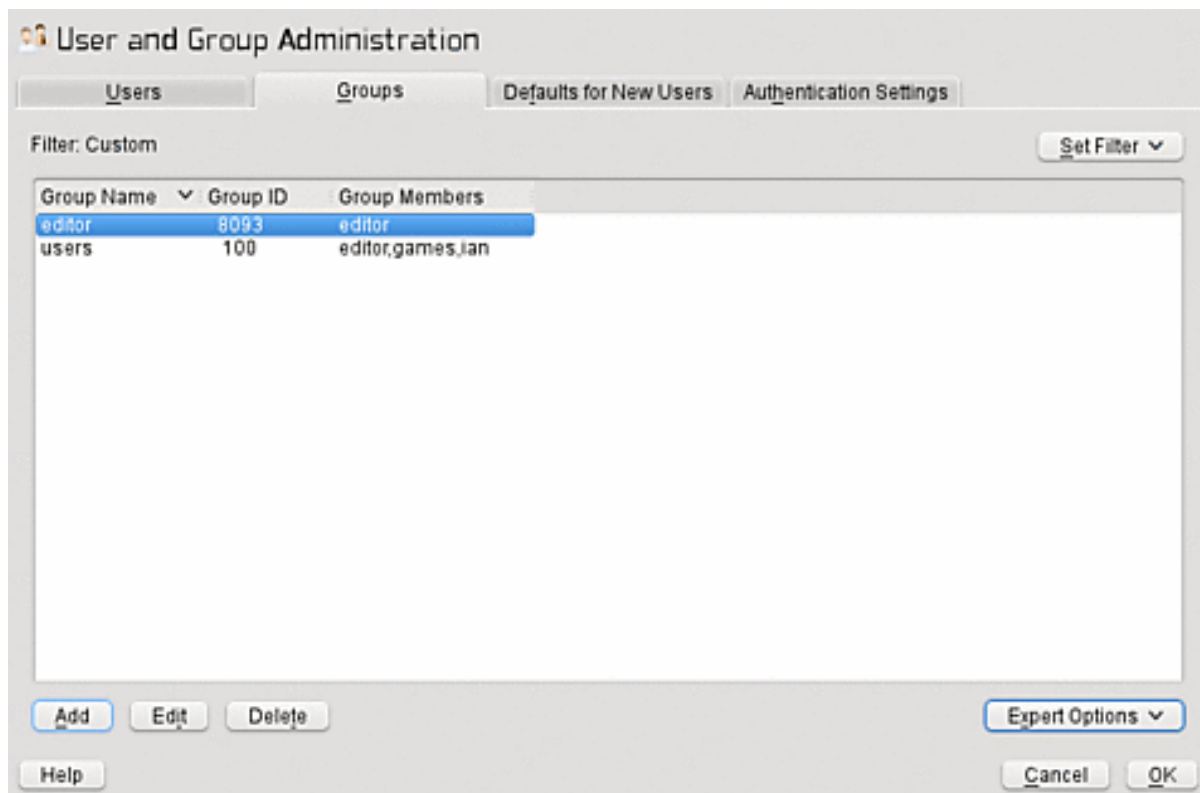
Open this application and click **Security and users** in the left panel to view the tasks shown in the main panel of Figure 33.

Figure 33. YaST2 Control Center



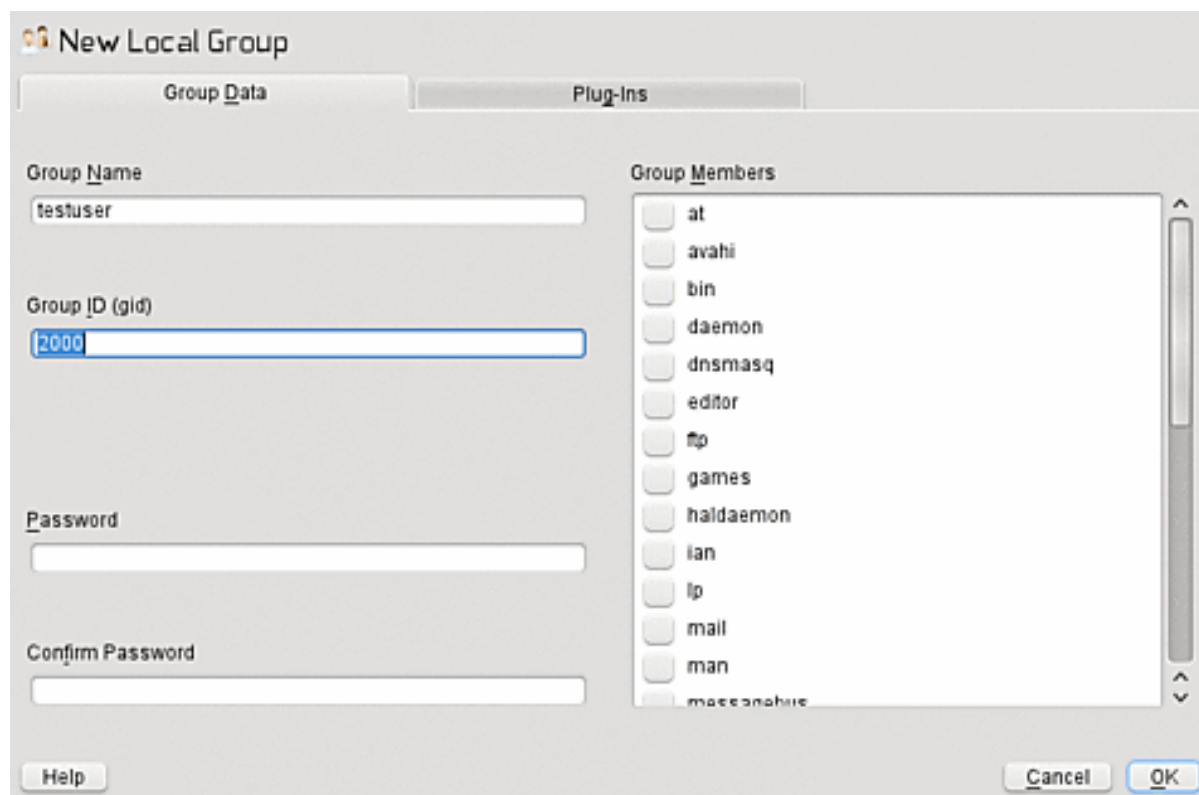
Select **User and Group Management**. If you have not recently received root authority, you will be prompted for the root password. On the next screen, you will see any existing users. Select the **Groups** tab, and you will see something like Figure 34.

Figure 34. Group list in YaST2 Control Center



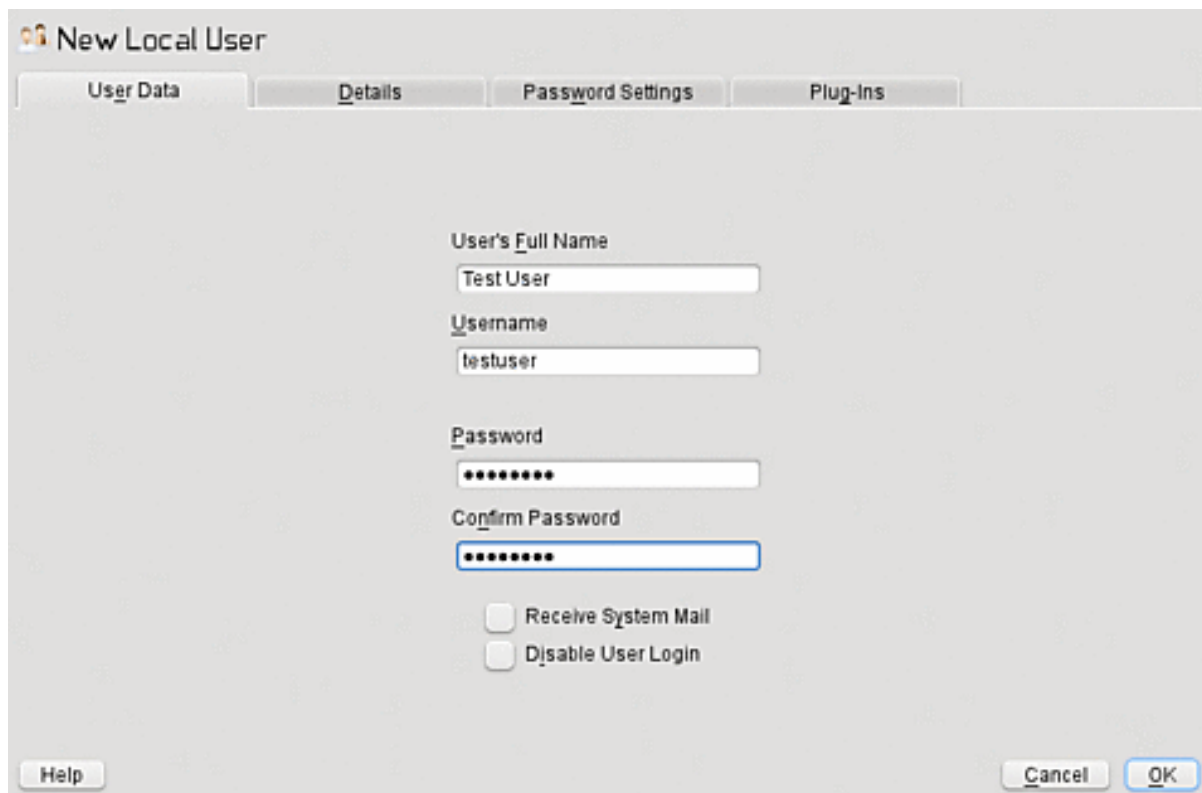
Click the **Add** button to add a new group. You will see something like Figure 35. Note that there are a number of groups that were created when you installed your system. Enter 'testuser' for the **Group Name** and '2000' for the **Group ID**. Click **OK** to return to the group list display and see the new group listed. At this point your group has not yet been saved to the system, so it will be lost if you cancel.

Figure 35. Adding a group in YaST2 Control Center



Click the **Users** tab to return to the user display, then click **Add** to add a new user. Enter 'Test User' for the **User's Full Name**, 'testuser' for the **User Name**, then type and retype an initial password for the user. See Figure 36.

Figure 36. Adding a user in YaST2 Control Center



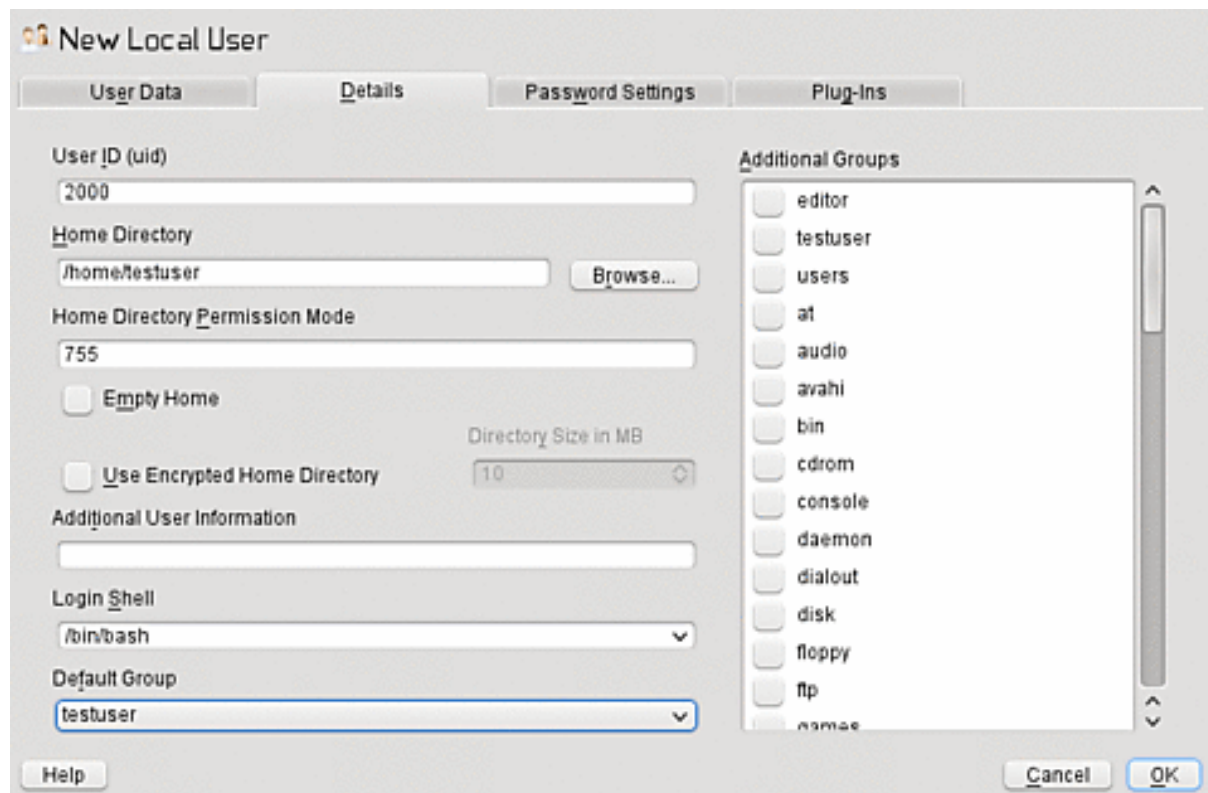
The screenshot shows the 'New Local User' dialog box with the 'Details' tab selected. The 'User Data' tab is also visible. The 'Details' tab contains the following fields and options:

- User's Full Name:** Test User
- Username:** testuser
- Password:** (masked with dots)
- Confirm Password:** (masked with dots)
- ☐ Receive System Mail
- ☐ Disable User Login

Buttons at the bottom: Help, Cancel, and OK.

Click on the **Details** tab and enter '2000' for the **User ID (uid)**, and select 'testuser' from the **Default Group** drop-down menu. This panel is where you can change the default home directory and default login shell, among other items. You can also select additional groups that this user will be a member of. When you have finished, click **OK** to return to the list of users where you will see your new user. Click OK, and YaST will process all your changes and save them to the system.

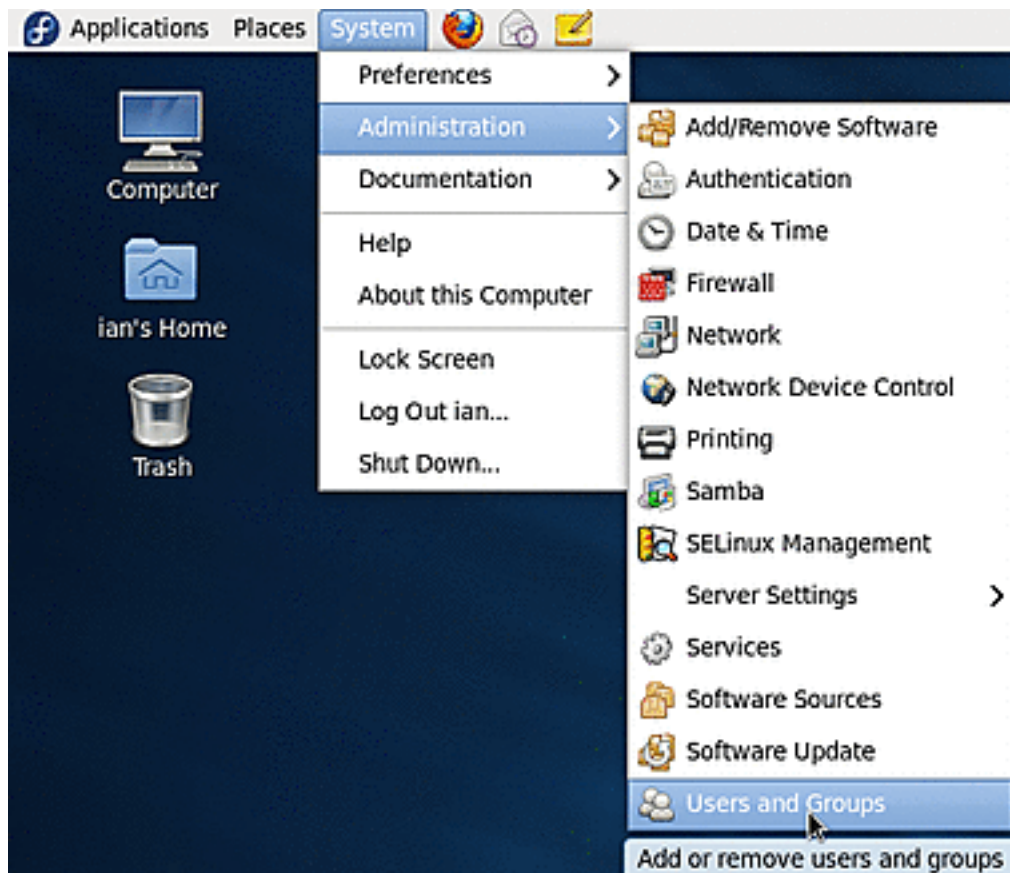
Figure 37. User details in YaST2 Control Center



Adding users and groups to your Fedora or Ubuntu system

On GNOME systems, such as our Fedora and Ubuntu systems, you start user management from the **System->Administration->Users and Groups** menu as shown in Figure 38.

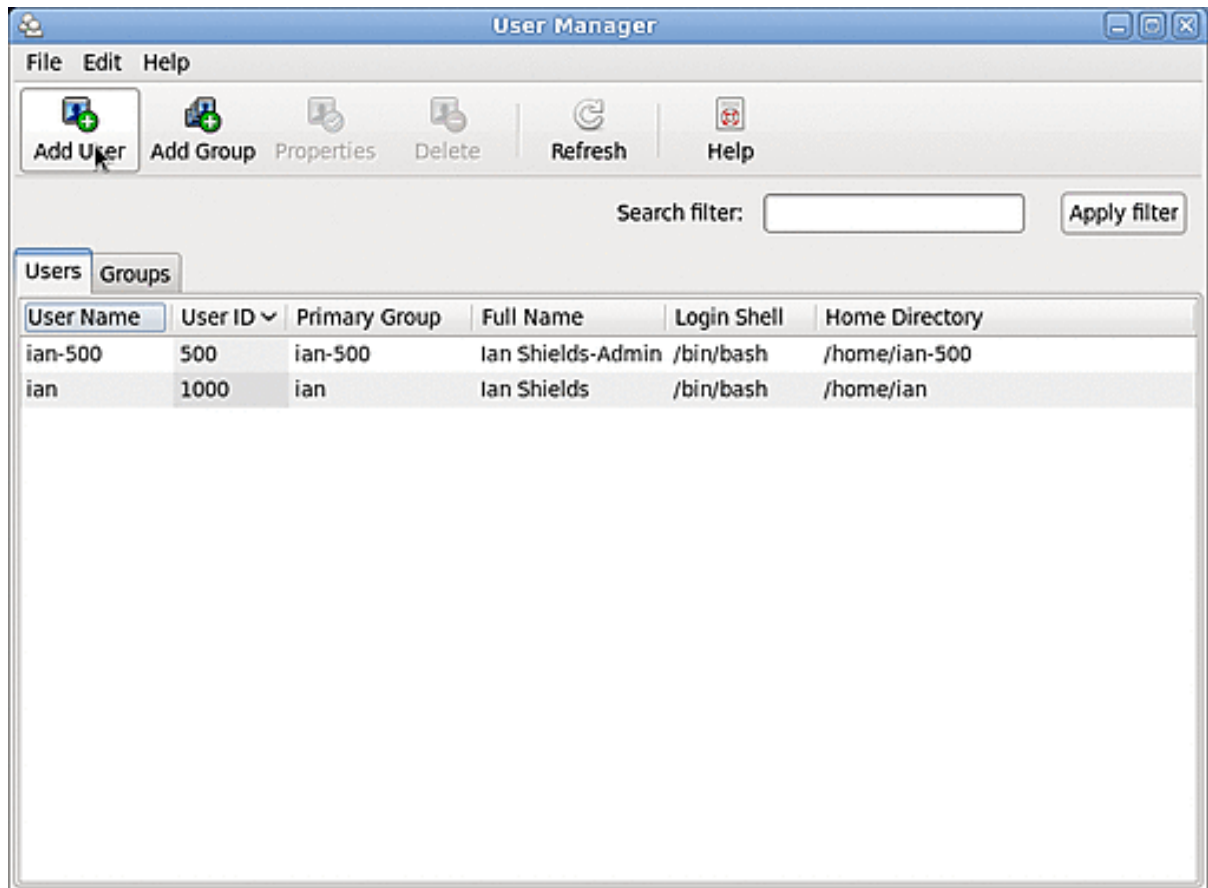
Figure 38. Starting user and group management on GNOME



However, Fedora and Ubuntu have different dialogs once you open the User and Group management. Fedora starts the `system-config-users` application, while Ubuntu starts the `users-admin` application. We'll illustrate the Fedora usage here and then summarize the differences for Ubuntu.

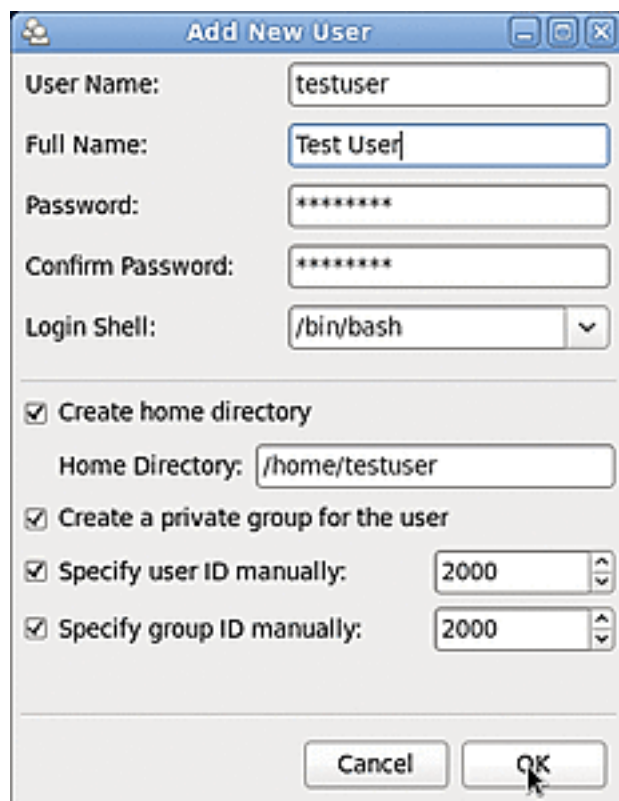
If you are not logged in or recently authenticated as root, you will need to provide the root password when prompted. You will then see the User Manager screen, opened at the **Users** tab as in [Figure 39](#). By default, only normal users and groups are shown. To view system users and groups, uncheck the **Hide system users and groups** checkbox under **Edit->Preferences**.

Figure 39. Fedora User Manager



We could do as we did above for the OpenSUSE system and define our groups first. However, the Red Hat User Manager has a convenient feature that lets you create a private group for a user with the group name being the same as the user name. So click the **Add User** button and fill in the details for the `testuser` user as for OpenSUSE above. However, this time, check the **Specify user ID manually** and **Specify group ID manually** check boxes and fill in 2000 for each of these values. Our screen now looks like Figure 40.

Figure 40. Adding a user in Fedora



The screenshot shows a window titled "Add New User" with the following fields and options:

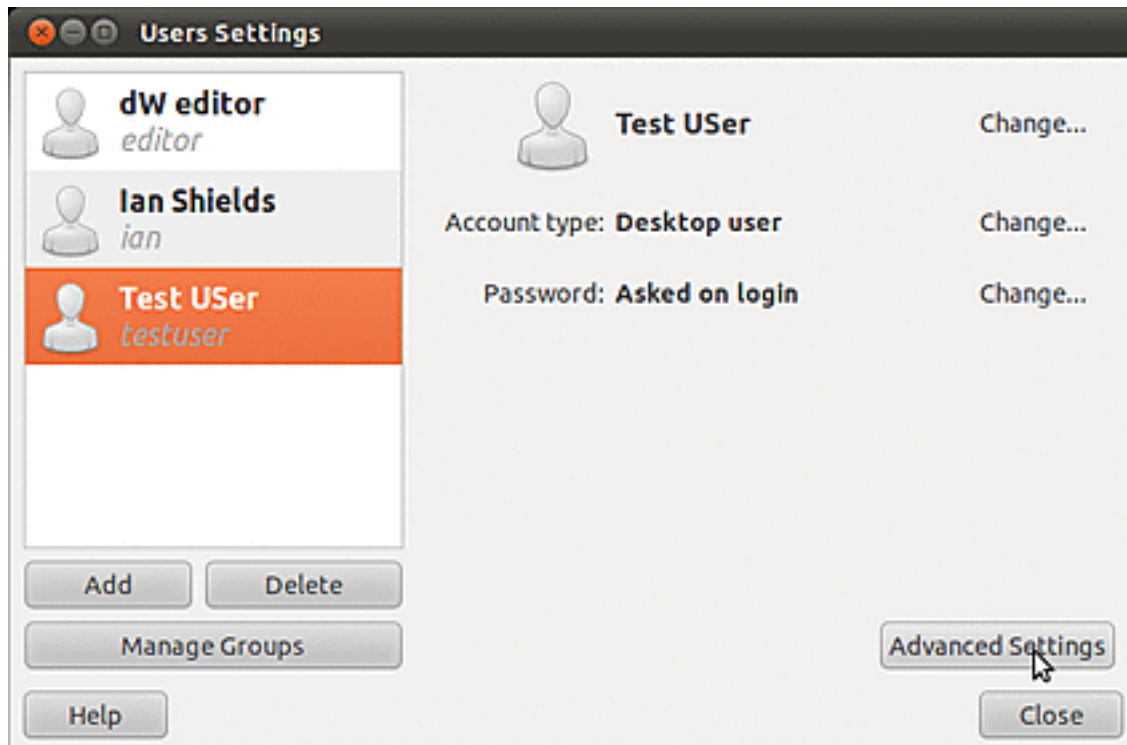
- User Name: testuser
- Full Name: Test User
- Password: masked with asterisks
- Confirm Password: masked with asterisks
- Login Shell: /bin/bash (dropdown menu)
- ☒ Create home directory
 - Home Directory: /home/testuser
- ☒ Create a private group for the user
- ☒ Specify user ID manually: 2000 (spin box)
- ☒ Specify group ID manually: 2000 (spin box)
- Buttons: Cancel, OK

After you click **OK**, you will return to the User Manager. Your new user will already be added to the system, unlike the case for OpenSUSE. Click **Add Group** to add any additional groups you may need. To make users members of additional groups, you can either select a group and use its properties to add users or select a user and use the properties to add groups. When you are done with properties, click **OK** to return to User Manager and then **File->Quit** to close the User Manager.

Ubuntu user manager

Now that you are familiar with adding users on both OpenSUSE and Fedora, you will be able to manage the Ubuntu process. In general, Ubuntu will lead you through the process of adding a user and set up the user with a default id and group. Once you have added `testuser` in this way, you should see a screen like Figure 41.

Figure 41. User management in Ubuntu



At this point you will need to use the **Manage Groups** button to add the new group (or you could add it before you add the user). Once you have added the `testuser` group, you will need to come back to the screen of Figure 41, select the `testuser` id, and click the **Advanced** button to change both the id number and primary group for the user.

Adding users and groups using the command line

You can add or change users and groups from the command line. These tasks require root authority.

Information on groups is stored as a flat file in `/etc/group`. You may use the `groupadd` command to add a new group. This is fairly simple. Adding a new user is a little more complex as there are more parameters, and you will need the numerical number of the user's group. Let's use the `groupadd` command to add our `testuser` group, with group id 2000, and then use the `grep` command to search `/etc/group` and verify the settings. **Note:** If you do not provide a group id, the system will assign the next one that is higher than any existing group id.

```
root@pinguino:~# groupadd -g 2000 testuser
root@pinguino:~# grep testuser /etc/group
testuser:x:2000:
```

As you see, the `testuser` group is 2000. Now let's use the `useradd` command to

add the `testuser` user. The `-c` option allows us to specify a comment that is usually a user's real name. The `-u` option allows us to specify the numerical id (2000) for the user. The `-d` option allows us to specify the home directory for the user. The `-g` option specifies the user's primary group. Here we use 2000, which is the `testuser` group we just created. The last option we use is the `-G` option to specify additional groups for this user. Here we can use the group name. In this case, we'll add `testuser` to the group `ian`.

Once you have added the user, you can use the `grep` command again, and you will see that user `testuser` has been added to the `testuser` and `ian` groups. At this point you have created a new user, but the user does not have a password and cannot log on to the system. Some users do not need to log on, so that would be alright for those users. The root user has the authority to set (or reset) passwords for other users. To do this, you use the `passwd` command and give the username as a parameter. You will be prompted for the new password, and then you will be prompted to retype it for verification.

```
root@pinguino:~# useradd -c"Test User" -u 2000 -d/home/testuser -g 2000 -G ian \
> testuser
root@pinguino:~# grep testuser /etc/group
ian:x:1000:testuser
testuser:x:2000:
root@pinguino:~# passwd testuser
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Finally, you may need to add users to an existing group. You can use the `usermod` command to do this, but you need the list of existing groups for the user as you will replace the list of additional groups. It is often easier to edit `/etc/group` directly. Make a backup copy first, just in case you make a mistake. To add the editor user to both the `ian` and `testuser` groups, edit `/etc/group` and update the lines for `ian` and `testuser` so they look as follows:

```
ian:x:1000:testuser,editor
testuser:x:2000:editor
```

You will find much of the user information is stored in `/etc/passwd`, but you should not edit this file yourself. Use the `useradd`, `usermod`, and `userdel` commands instead. If you are not a full time system administrator, you will probably find it easier to do occasional manipulation of users and groups through the graphical interfaces.

Resources

Learn

- For help with more advanced Linux development tasks, take a look at the [developerWorks roadmap for LPIC-1](#). The roadmap leads to many developerWorks articles to help you study for LPIC-1 certification based on the April 2009 objectives.
- Customize your own prompts with "[Prompt magic](#)" (developerWorks, September 2000).
- Learn more about how devices are mounted in "[Learn Linux, 101: Control mounting and unmounting of filesystems](#)" (developerWorks, October 2010).
- In the [developerWorks Linux zone](#), find hundreds of [how-to articles and tutorials](#), as well as downloads, discussion forums, and a wealth of other resources for Linux developers and administrators.
- Stay current with [developerWorks technical events and webcasts](#) focused on a variety of IBM products and IT industry topics.
- Attend a [free developerWorks Live! briefing](#) to get up-to-speed quickly on IBM products and tools, as well as IT industry trends.
- Watch [developerWorks on-demand demos](#) ranging from product installation and setup demos for beginners, to advanced functionality for experienced developers.
- Follow [developerWorks on Twitter](#), or subscribe to a feed of [Linux tweets on developerWorks](#).

Get products and technologies

- [Evaluate IBM products](#) in the way that suits you best: Download a product trial, try a product online, use a product in a cloud environment, or spend a few hours in the [SOA Sandbox](#) learning how to implement Service Oriented Architecture efficiently.

Discuss

- Get involved in the [My developerWorks community](#). Connect with other developerWorks users while exploring the developer-driven blogs, forums, groups, and wikis.

About the author

Ian Shields

Ian Shields works on a multitude of Linux projects for the developerWorks Linux zone. He is a Senior Programmer at IBM at the Research Triangle Park, NC. He joined IBM in Canberra, Australia, as a Systems Engineer in 1973, and has since worked on communications systems and pervasive computing in Montreal, Canada, and RTP, NC. He has several patents. His undergraduate degree is in pure mathematics and philosophy from the Australian National University. He has an M.S. and Ph.D. in computer science from North Carolina State University. You can contact Ian at ishields@us.ibm.com.