ASK KLAUS!

Klaus Knopper is the creator of Knoppix and co-founder of the LinuxTag expo. He currently works as a teacher, programmer, and consultant. If you have a configuration problem, or if you just want to learn more about how Linux works, send your questions to: klaus@linux-magazine.com.

Are Gnome and KDE equal?

? Gnome and KDE are always presented as equal alternatives. Are they really equal, or are there situations where one is better than the other? Can I configure my Linux system so that it will run BOTH KDE apps and Gnome apps?

Technically speaking, both Gnome and KDE are highly configurable desktop systems with object-oriented internal design, and even for non-technical users, these desktops (if configured appropriately) are very easy to use.

I'm trying to be neutral about which desktop system is "better" in general. I have been using Xfce 3 for quite a while, and even with its very reduced features, it has always been sufficient for me.

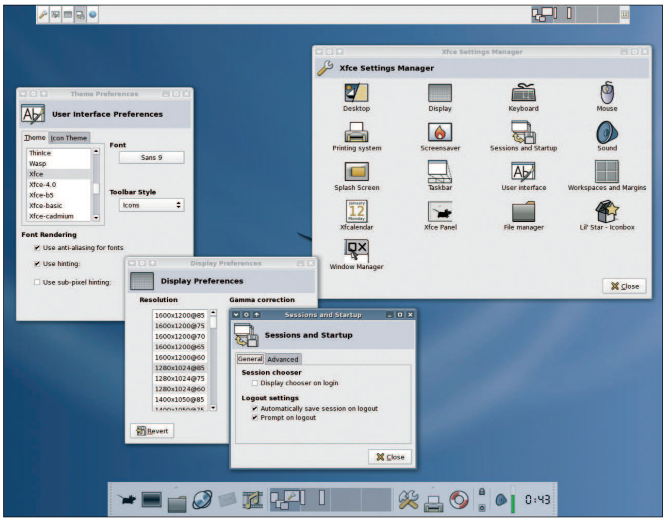
There were quarrels between developers and users about KDE being "non-free" in the old days, because of the qt license (for the library KDE is based on); that it was more memory-consuming than Gnome; or that its graphics were poorer. None of these are genuine problems today, and I don't know if the rumors about graphics and memory consumption ever were true, or whether they were just part of the creative competition between different groups and philosophies.

Nowadays, Gnome consumes the same amount of system resources as KDE (if not more), performance benchmarks are equal, graphics are excellent for both, and their features (at least those that you will most likely ever use) are more or less identical.

Gnome and KDE do, however, use an entirely different set of dependencies in the form of daemons, services, and libraries. I have seen Gnome programs that require 40 library dependencies or more. It sometimes seems impossible to use a Gnome or KDE program that doesn't require additional components that are about 10 times the size of the actual program. (Static linking of KDE or Gnome programs is more likely fail because of this modular structure than for any other reason.)

This can get you into some problems. If you use KDE regularly, and you want to install a single Gnome program such as Gnopernicus, you end up installing the entire Gnome software suite because you won't be able to to get the program to run otherwise. In very bad cases, Gnome dependencies try to deinstall parts of KDE (or vice versa) because of conflicting services that are present in both (a very rare case, though).

OK, I cheated a little in this example. Gnopernicus is more like a suite of plugins and addons for Gnome than like a standalone application. (It is a graphical screenreader and accessibility enhancement for Gnome.) But the same thing can also happen with smaller applications, such as the great GnomeMeeting video conferencing client, gnumeric, and others that I surely would like to have



Theme Preferences – User Interface Preferences

Theme: Xfce (ThinIce, Wasp, Xfce, Xfce-4.0, Xfce-b5, Xfce-basic, Xfce-cadmium)

Font: Sans 9

Toolbar Style: Icons

Font Rendering

☑ Use anti-aliasing for fonts

☑ Use hinting:

☐ Use sub-pixel hinting:

Display Preferences

Resolution: 1600x1200@85, 1600x1200@75, 1600x1200@70, 1600x1200@65, 1600x1200@60, 1280x1024@85, 1280x1024@75, 1280x1024@60, 1400x1050@85

Gamma correction

Sessions and Startup

Session chooser

☐ Display chooser on login

Logout settings

☑ Automatically save session on logout

☑ Prompt on logout

Figure 1: Xfce is a fast and simple alternative to Gnome and KDE.

available on the KDE-based Knoppix CD edition, but they just won't fit with all their library dependencies. It's the same thing the other way round if you want to install a KDE application on a plain Gnome system. But at least KDE 3.x programs try to avoid unnecessary dependencies if possible.

In any case, when using KDE with Gnome applications, or KDE applications on Gnome, you will always end up with a lot of libraries and background services, object brokers, or preloaders from one or the other system. Luckily, both Desktop systems try not to fight each other (on the runtime level) or keep the other desktop system from running its own stuff. So yes, it's possible to freely mix Gnome and KDE applications, except in a few cases when both insist on starting their very own (sometimes device-blocking) sound daemon, and you end up wondering why your sound card suddenly fails to respond until you kill a few running processes launched by the "other" desktop system. If you encounter such problems, you can check who's blocking your sound by entering

```
fuser -v /dev/dsp
```

which will identify the offending sound daemon process in most cases.

Partitioning for Experts

? Linux installers have gotten much better over the years, but the section on partitions is just as confusing as it always was. Now, at least, the wizard "suggests" a partition configuration, but it still asks me whether I approve of the settings, and I have no idea what I am approving. There is usually an "Experts Only" button, which leads to a new page where everything is expressed in terms of partition names and locations on the disk. Why would an expert want to use a different partition configuration, and if so, is there a reason why I would want to do so? Is there an advantage to changing the size of the swap file or the location of the partitions on the disk?

It is an advantage to create partitions in a way that is optimal for the desired purpose of the specific GNU/Linux installation. If this is your first-time Linux installation, and you just want to do everyday work or play, you don't usually have to worry about "expert" options and can just go with the defaults.

Manual configuration is preferable if you plan to do some more advanced tasks with your computer, like realtime streaming, video processing, or other heavy duty data processing that requires a more sophisticated setup. It also matters whether or not the system will be a home to a lot of different users, or whether it will be a single user system.

I usually use a lot of partitions in order to have some separate test installations, encrypted partitions, and additional swapspace areas whenever I need them.

A GNU/Linux system can fit on a single partition (which can even be a virtual partition located inside a file). Swapspace is not required in order to run GNU/Linux at all (though some installers insist on having a swap partition).

So, let's discuss swap first.

Swap adds space to the "virtual memory" management of the Linux kernel. This means, you can run more and bigger applications than would fit into your computers RAM. The question of how much swap space you need depends on what you plan to do. *Real* RAM is always better, of course, but in most cases, swap will also work by temporarily storing running programs and data that are not currently needed.

A fine Linux performance-boosting feature is the fact that unused RAM will be used as a dynamic (self-adjusting) file system buffer. Files that have been read once will stay in the fast RAM cache (and just be read from there) until the RAM is needed more urgently by something else (such as a running program that requests memory for graphics, for example).

As an example of how memory usage works, OpenOffice will need about 100 megs of RAM when you start it for the first time. As your working document grows and contains elements from different parts of OpenOffice, other components will be needed, like a spreadsheet or drawing or presentation component. You can end up using 300 megs or more by just running OpenOffice for a while. The KDE desktop also needs about 100 MB for itself, and if you open the Firefox browser, the Gimp graphics manipulation program, and other potentially memory-hungry programs, you are already at about 500-1000 MB that need to be kept somewhere.

If you manually calculate the required swapspace, try to estimate the memory used by programs. (The *free* command is helpful for this, but watch the values WITHOUT file system buffers), and add another third of this number just to be safe. If your system runs out of memory, everything starts being very slow and responsiveness goes down to a point where you want to leave the room to get some coffee or tea. Running applications may also be terminated if there is no way they can get the memory they need. But usually, before this happens, you are already considering using the reset button to restart the system, knowing that this is not the preferred way to free memory. So, it is good to think about swapsize in advance.

"...it is good to think about swapsize in advance"

If you notice later that your swap partition is unsufficient, there is an easy way to add more swap. Just create a file, mark it as swap, and add it to the virtual memory like a partition. The following example shows this with a 100MB file on */var*:

```
dd if=/dev/zero of=/var/swap ➘
bs=1024k count=100
mkswap /var/swap
swapon /var/swap
```

and now, you have 100MB more swap. To make this permanent, put this line into */etc/fstab*:

```
/var/swap  none  swap  sw  0  0
```

Dedicated swap partitions are much faster than swap files, since there is less filesystem overhead. (The kernel writes directly to the partition, instead of calling file system functions.)

For file system partitions, separating “often written” data from “mostly static” data can be useful. This precaution ensures that one overflowing partition (with logfiles, for instance) won’t affect others, and you will still be able to login and solve the problem (at least, in most cases) without having to switch to an administrative runlevel. Having everything on a single partition, on the other hand, has the advantage that the available space is shared by everything and you don’t have to worry about resizing partitions when one of them is running out of space.

I usually use one partition for the “installed system files” (containing the root file system and */usr*), one partition for the frequently written-to */var* file system (also containing logfiles and */tmp*), one encrypted partition for authentication keys, passwords, and backups of customer data, one partition for my */home* directory, and one partition for experiments and “unimportant data” like test installations.

Something like this:

```
/dev/hda1    5GB /
/dev/hda5    2GB /var
/dev/hda6    1GB /crypt
/dev/hda7    8GB /home
/dev/hda8   20GB /mnt/scratch
```

Of course, this setup is very individual, and you will have to repartition and reconfigure every once in a while when one or the other partition is filled up, or whenever a disk upgrade is planned.

If you have several disks, it also makes sense to distribute partitions that contain frequently used data across different controllers (for example, */dev/hda1* contains the system files, and */dev/hdc1* the home directory, */tmp* or */var*). This reduces disk head movement in parallel read/write operations. For video data processing, spreading the partitions across different controllers can be essential to achieve the throughput necessary to encode videos in real time.

Printer Problems

I have an HP 1200 Business Inkjet printer. The box for the printer did not specifically advertise Linux compatibility, but I was able to set up the printer easily in my KDE-based Linux system using a PPD file I downloaded from the web. The printer works fine for printing documents. The one problem is that I don’t get the usual messages from the printer if something goes wrong (paper jam, out of ink, etc.). I just get a standard message saying something like “Printer Error.” Is there any way to receive the same feedback messages from the printer that (according to the manual) Windows users receive?

CUPS messages usually go to */var/log/cups/error_log* and */var/log/cups/access_log*. Their verbosity depends on the setting of *LogLevel* in */etc/cups/cupsd.conf* (see the description of log levels in Listing 1).

So, you may get more information about the driver internals if you set the LogLevel to *debug* or even *debug2* in */etc/cups/cupsd.conf*. (Don’t forget to restart CUPS using */etc/init.d/cupsys reload* in Debian.)

The kind and visibility of information given by a PPD file really depends on its structure and configuration, so it may well be that certain features of that printer cannot be accessed directly from within CUPS.

For a table of supported printers and capabilities, *http://www.linuxprinting.org/* is a good address for research.

More specifically for the printer we have been discussing, linuxprinting.org says at *http://www.linuxprinting.org/show_printer.cgi?recnum=HP-Business_Inkjet_1200* that this printer works perfectly with Linux/CUPS, therefore you should not need a custom PPD, but only the installed hpijs driver (and a current CUPS version).

If you are searching for a new printer that is well supported by CUPS, *http://www.linuxprinting.org/suggested.html* gives you an overview of printers which have been shown to work extremely well.

In some cases, the printer vendor has a special status monitor program, such as a “Printer LCD display monitor” for Linux, which works independently of CUPS and fetches information (read-only) directly from the printer connector. Especially for printer/scanner combinations, third-party add-on programs like this sometimes let you access or at least view the “hidden features” of the printer. I personally don’t like them much if they come only in binary form and without a license that allows me to analyze or modify them.

Just check the driver CD that comes with the printer for any native Linux userspace programs, or again, you can check linuxprinting.org, which sometimes has links to third-party or vendor-supplied addon programs for specific printer models.

For your HP 1200 printer, linuxprinting.org recommends checking out the hplip driver on sourceforce *http://hpinkjet.sourceforge.net/* for extended features such as status and maintenance reports. ■

Listing 1: CUPS Log Levels

```
# Log level (LogLevel)
#
# Controls the number of messages logged to the ErrorLog
# file and can be one of the following:
#
#     debug2:    Log everything.
#     debug:     Log almost everything.
#     info:      Log all requests and state changes.
#     warn:      Log errors and warnings.
#     error:     Log only errors.
#     none:      Log nothing.
#
# ex: info
#
# Default: LogLevel info
```