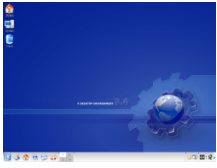


2 April 2006

By: Marius Nestor, Linux Editor



## [Gentoo Linux Review](#)

*To compile or not to compile, that is the question.*

**What is this Gentoo?** Linux is an open source operating system and almost all of the software packages are open source. Some are distributed under licenses that can be considered even more “free” than the GNU GPL licenses, while some are given away without any kind of license. Having the source code of the Linux kernel and the source code of thousands of software packages is the obvious reason why anyone can start putting together their own personalized Linux distribution by compiling it from source code. Perhaps there’s no need to go that far, thanks to Gentoo. Gentoo is a source distribution. While there are various “stages”, which have different levels of already compiled binary packages included, the main idea is building every package from source code. There are GRP packages, which reduce the need of compiling certain packages that take a very long time to compile. The creation of this distribution was inspired by the ports system of FreeBSD, there’s even an experimental stage with a FreeBSD kernel and the Gentoo “portage” called Gentoo/FreeBSD. Well, so far we’ve talked about the fact that Gentoo can be compiled from the source code. There are numerous advantages, but there are also problems. Among the advantages, compiling from source code and installing packages only with the desired dependencies can reduce the size of the operating system and might increase the overall speed. The fact that packages are compiled from the source code can also be a disadvantage, for example, compiling an entire system, with a desktop environment and all other programs can take a very long time, from a few hours up to a few days or even more, depending on how much software you want to install. A way of getting around this issue is the usage of ccache and distcc. While Ccache caches the compiled object files, distcc distributes the compilation process across many computers, which can save a lot of time, depending on the number of machines, their CPU speeds and the number of processors. This can be very useful, but mixing high end computers with very old computers, like a few computers which have Pentium 4 CPUs and older machines which only have Pentium 2 class CPUs, won’t speed the process up, actually, it will slow it down. Beside this problem, compiling packages with distcc and mixing different GCC versions isn’t recommended, using GCC 3.3.6 on some machines and GCC 3.4.4 on others will often result in broken or unstable packages. Only versions from the same release series can be mixed, for example, it would be fine if there are some machines that use GCC 3.3.5 and others use 3.3.6. Ccache can be described as a tool that automatically and transparently saves the work that has already been done for later retrieval and usage. Distcc uses the parallelization part of GNU Make and by specifying the right number of jobs (-jN, where N is the number of CPUs that are compiling with distcc) to compile simultaneously on many computers. Gentoo is great in case you want to build a custom operating system with the packages that you want and without anything else. In case you want to learn more about the development of the Linux distributions, Gentoo should provide both tools and information on how to get most things done. Some want to install Gentoo because they heard it is very fast, others want to learn more about the development and the inner workings of Linux distributions, while some are hardcore tweekers. There are lots of HOW-TO guides on Gentoo’s Wiki system. For more information and help, questions can be asked on the Gentoo forums. Before the release of Gentoo 2006.0, the only way to install it was by doing many things manually, without any kind of help from a graphical user interface and that’s one of the things that some users didn’t like and why they didn’t go for a Gentoo install on their system. Gentoo was an operating system that would be a bit hard to start with for the first time, but as soon as it was installed and working, things would get a lot easier. Installing additional software

packages was easy, usually, the only thing that has to be done in order to install an app is to issue a command like this: “emerge package-name” and the install procedure would start, the source code is retrieved, unpacked, patched, if there are any patches for it, configured and then installed. Some packages might also need additional configuration editing, so emerging a package isn't always the whole picture, for example, if you install the Apache web server, you'll have to configure it in order to have it set up the way you need it to be, some virtual hosts might have to be added and other such things. In other cases, it's the other way, you install a package that needs certain variables to be set, at the configuration phase, compilation or install phase and portage won't be asking you to set those variables. For example, if you install GCC by using emerge gcc, portage will run ./configure make and make install with the right parameters. That's because the Gentoo developers that handle the package creating tasks have taken care of all this; they create the so called ebuild files which configure, compile and install the packages for you. Furthermore, you don't even have to check dependencies, portage checks them for you and also installs other required packages. **Gentoo is hard to install, what about the new installer?** The classic way of installing Gentoo wasn't very appealing for beginners and for some of them, getting it to work was a big issue. After they would go through the entire install procedure and tried to boot the resulting system, their installed Gentoo didn't boot at all or errors showed up. Obviously, they did something wrong at a certain point, but they don't always realize where. This and the fact that people complained about the hard to perform install process and other reasons have led to the development of a graphical and text mode GLI. GLI stands for Gentoo Linux Installer. This GLI can be found on the Live CD disc that contains the completely compiled stages (the so called “stage 3”) and the precompiled software packages. This Live CD contains a Gnome desktop environment and after the computer boots and starts Gnome, two shortcuts appear. These are the two versions of GLI, the text mode and the graphical user mode. The Gentoo Linux Installer, the version which has a GTK+ front end is the graphical user interface for the GLI. As soon as you click on the corresponding shortcut on the desktop, it starts and it displays a greeting message and some information about the GLI and about the GTK+ front end. Press forward once and the installer asks you to configure the network, a configuration that is going to be used during the install process. Configure your network adapters and, in case you use a proxy, the proxy information, then press forward once more. Before doing so, you might want to load additional modules, if you use a SCSI controller or other hardware that you might have to load a module for in the “misc.” section. The following section enables you to partition your hard disk. If you have other partitions on your computer on which you have data which you want to keep, you will have to take care when partitioning the hard drive. If you also have other operating systems installed, there's even more, you will either have to keep the current boot loader in the MBR (master boot record) or install a new boot loader and configure it to handle the booting of the other operating systems. In case you started with a clean hard drive, without any data or with a hard drive which you want to be wiped entirely, you can just press “recommended layout”. The recommended layout contains 3 partitions, a boot, a swap and a root partition. Obviously, the boot partition is small, around 100 MB, the swap partition's size can be up to 512 MB and the remaining space is allocated to the root (/) partition. The following section is the network mounts section, here you can define NFS mounts. This can be used to mount one or more remote directories by using NFS. As the developers have said, one of the many uses that it has is mounting an /usr/portage from a different computer and avoiding running “emerge --sync” locally, thus saving space. This can have many advantages, you don't have to do emerge --sync on each machine and you need to run emerge --sync only once for many computers. Beside the mentioned advantages, this would also respect the Gentoo netiquete, which says you should “emerge --sync” only once a day from a mirror, to save bandwidth and let others synchronize their portage tree. The next section is important, here's where you get the chance to decide how long it will take to install Gentoo on your PC. The stage 3 install with GRP Install and

Dynamic turned on will be the fastest install. It's obvious that installing from stage 2 takes longer than installing from stage 3 and installing from stage 1 takes even more. In case you decide to perform that stage 3 with GRP Install and Dynamic turned on, you should be aware that you'll be needing a lot of patience, don't perform anything else other than this install if you need to use your PC as soon as possible. For example, if you choose to compile OpenOffice.org from sources later in this installer, you will have to wait for a very long time, depending on the speed of your CPU, whether it is or not an SMP machine and if it is, on the number of CPUs. On slow single CPU machines it can even take two or three days. If your PC is very slow, like a Pentium 2, the install process takes far too long to complete and you shouldn't perform anything else than the stage 3 that I recommended. There's one way of getting around these long compilation periods, I've mentioned it earlier, using ccache and distcc. More about that will follow later. After you chose the install stage of your choice, you get to the Portage tree selection. Here's where you get to say which portage tree snapshot should be used. Obviously, you have to put the link to the desired snapshot in the "Portage URI" box. That link doesn't necessarily have to be a link to the Internet, it can also be a local file. Those who took my advice and chose the stage 3 installed with the two options enabled should skip this section, as it is completely disabled, all the settings have been made based on your stage 3 choice, the files are going to be used directly from the CD. The next section is "make.conf". Here's where you're going to perform the so called optimization of the system and specify things like the USE flags, CFLAGS, ccache and distcc. The architecture is, obviously, the CPU type you have in your PC, if you have a Pentium 4 CPU, you can choose Pentium 4 as the architecture; if you have an Athlon XP CPU, you can choose the Athlon XP architecture. If you don't know what CPU you have or if you want to migrate this Gentoo install to a different computer with a different CPU type, you might want to leave it on "i686". The following setting is the "Optimizations". You shouldn't use O3 as it takes longer to compile, the generated code will be sometimes faster, sometimes slower and almost always larger. For example, a program that was compiled with the O2 level of optimization is 250KB large and is quite fast, the same program compiled with the O3 level of optimization can take more time to compile, produce a larger binary, depending on the code, even 500-700KB large and it can be slower or faster. Therefore, the O3 optimization level isn't recommended. Use O2 if you need the best optimization without adding the risk of obtaining large binaries, which need a long time to be loaded before being executed and are not always faster. Using the "unstable arch" is not recommended, as it can ruin your Gentoo install. Build binary packages can be used to generate and store binary packages for later install on a different machine. Distcc handles the distribution of the compilation process and ccache caches the compiled source files. I recommend using ccache, I would also recommend using distcc, but it hasn't been implemented in this GTK+ front end, distcc can only be used if you use the command line GLI. That means that ticking the distcc box won't make any difference, it's not working yet, perhaps it will work in the next release of Gentoo, 2006.1. The makeopts option should not be ignored, especially if you know you have an SMP machine. For example, if you have a dual core CPU, you might want to specify the number of simultaneous jobs. This means that you can compile more than one source file at once and compile packages faster if you have more than one CPU. The setting that affects the number of simultaneous jobs is "-jN", where N is number of CPUs. That means you write -j2 if you have a dual core computer. It's obvious that the compilation process is even shorter if you have many CPUs. The "custom" text box shouldn't be ignored either, I personally recommend leaving -pipe where it is, this disables the creation of temporary files during the compilation and generates only the final compiled object files. Instead of the temporary files, it will use pipes. What other things should end up in this custom box? Well, thinking that the more options you add, the faster the system will be, won't do any good. Probably it will be very fast, but extremely unstable, which isn't a desirable thing. Using some additional custom flags like -s -DNDEBUG would be a great thing. The -s flag strips additional data from the binary object

file, things that won't affect the functionality of the resulting binary object, the `-DNDEBUG` flag prevents the compiler from generating debugging information for the binary object. This improves performance because the debugging routines won't be included and the program is likely to execute a lot faster. Some may wonder why I mentioned the `-s` flag if I included the `-DNDEBUG` flag, well, the debugging information isn't all that the `-s` flag strips, there are more sections inside the object file, like comment and version, which can be removed in order to save some space and gain some extra performance. Some might also say that not generating debugging information is useless, because it can be stripped by using the `strip` flags variable in `make.conf` and that some programs will lose all the information that could be used to track down a bug. Generating the debugging information takes time, because the compiler has to compile the debugging code, so it's better to not compile it at all rather than stripping it later. The fact that the resulting binaries won't be quite debuggable isn't an issue to normal users, it affects developers - normal users usually don't have anything to do with the debugging process. The USE flags, which can be found above all of these settings, are disabled if you perform a stage 3 install like I recommended. These USE flags configure packages, meaning that you can instruct your Gentoo system to build or not to build certain features of some packages. For example, you can specify in the USE flags that you want to have mpeg2 support in applications that can support it, include support for special CPU instructions, like MMX, SSE, SSE 2 and many other things. The following section is about selecting the kernel. Here you can choose the kernel that should be installed onto your machine. Those who are concerned by security can choose from `grsecurity-sources` and `hardened-sources`. These two kernel sources are patched for security and offer better security; this is not normally required for a desktop computer, they're more like server oriented. The other three options are `gentoo-sources`, `vanilla-sources` and `livecd-kernel`. The vanilla kernel sources represent kernel sources without any other patches, just like it's available for download from [www.kernel.org](http://www.kernel.org). The Gentoo kernel sources represent the same vanilla sources with a lot of patches for performance. The third one just copies the kernel from the live CD, without compiling it or including the sources in the installed system. If security isn't a concern, you can go for the `gentoo-sources`, a kernel built from these sources will be quite stable and fast. There's no point in choosing a hardened or `grsecurity` kernel if you have a desktop computer. If you don't want to wait at all for the kernel to be compiled, the live cd kernel can be copied and you're done. This choice has a catch, you won't have the kernel sources in your system. You can also make this live CD kernel choice if you plan to emerge some sources later, patch them and compile manually your own kernel. If you liked that little boot splash that was displayed during the start up of the live cd, you can include support for such boot splashes into your kernel just by turning the setting on, "enable boot splash support". The following section is delicate, you have to take great care here because you can render your old operating system, which was installed on your PC, unbootable, obtain an unbootable installation of Gentoo, or both. Depending on your hardware and the way your data is stored on your hard drive(s), you can install the boot loader into the MBR or not. If you added a new empty hard drive for the new Gentoo system, you can install the boot loader into the MBR of that hard drive, if you install Gentoo on a hard drive that had a different operating system, you might want to consider if you should install it to the MBR or not. Both `lilo` and `grub` support other operating systems, even those which are proprietary, therefore installing the boot loader to the MBR and configuring it to boot the other operating system as well is the way to go. Options can be passed to the kernel; for example, you might want to pass the option "quiet", which would force the kernel to stay silent, it won't make any output regarding hardware and all of the messages you would normally get before the Gentoo scripts start running. This results in smaller boot times and in the lack of information regarding the current status of the kernel - this might prevent you from realizing what went wrong during the boot phase. After you made up your mind, you can go on and choose the timezone in the following section. If your location is available on the list, you can choose the nearest location. The following section regards

networking configuration. Take your time to fully configure the network and double check it, you don't want to have any bad surprises when you'll try to install some package and realize that you can't because the network wasn't properly configured. Next, you get to choose to choose the cron and system logger daemons. It's recommended that you stick to the default ones, vixie-cron and syslog-ng. Another important section follows - extra packages. If you're performing a desktop install, you'll want to check xorg-x11, at least one window manager and some of the other custom packages. You might also want to install some of the software packages that are GUI applications for X11, because you also need functionality from your operating system, not just a desktop environment to stare at. More packages can be installed after the install has been done, even here, you can install extra packages by adding them to the text box below the extra packages selection box. Please note that this will take additional time to compile and will have to be downloaded. Those who are doing the stage 3 install I keep talking about can see GRP near the name of some packages, that means the package is available on the live cd as a precompiled package, ready to be installed and used. As a general rule, you should also install UFED, which is a USE flag editor, it's the app that I mentioned earlier, it can be easily used to change the flags and recompile the system later with added or removed features. Please note that the GNOME environment is available for installation from a binary package. KDE fans might want to compile KDE later, because compiling KDE takes a long time. The next section is the place where you get to configure the start up services, in other words, the daemons that are started each time your computer boots up. If the system you're installing Gentoo on is a desktop one, you should choose alsasound to be started automatically. The next section is used to set up some xorg related settings. You can set the display manager, which is that nice window which appears and kindly asks you to login into the system, the clock, whether it is UTC or local time, the keymap and the default XSession. The default XSession is the default window manager that is started when you start the X.org X11 server, those who install the gnome environment should choose gnome, those who install KDE should choose KDE and those who chose to install something else should choose whatever window manager they installed. The next section is the users section, here's where you can set the root password and create other users. Please note that you should create at least one other user beside root, because it's not safe to use your system as the superuser. You can use su whenever you need root rights. The passwords that you choose should be easy enough for you to remember and complex enough in order to be very hard to be guessed, especially if you're installing on a computer which will be used as a server. This is it, the last section before the install process; you can save your chosen settings to a file for later usage or review, or for performing another install with these settings or slightly modified ones. Before pressing the Install button, make sure that you configured the system correctly and that you considered each setting you made and that you double checked each, you don't want to obtain a fresh non-working Gentoo install, lose all of your data, or both, do you? This is all, you press install and the process of getting Gentoo on your computer is started. Depending on your choices, sooner or later you should obtain a working Gentoo install. The other Gentoo installer version, the GLI in text mode looks like an ncurses-based application. It can be used in two modes: load the configuration from xml files or create a new one. If you create a new one, you can go through a wizard or set your options one by one in dialog boxes. In order to use distcc and ccache, you should enable ccache and distcc features, in this order, "ccache distcc" and then set the hosts on which you'd like to compile this system. Please note that you cannot mix various versions of GCC and that you should burn multiple copies of the Gentoo LIVE CD in order to be sure that you have exactly the same version of GCC as the one which will be used to compile your new Gentoo system. You will also have to configure a small network, the other computers should be up and running, and distccd started. In order to configure distccd, you'll have to edit the allow hosts line in the /etc/conf.d/distccd file and insert the network from which they will be accessed in order to perform the distributed compilation, then distccd should be started by issuing the command:

“/etc/init.d/distccd start”.The classic way of installing Gentoo can still be used, those who don't like the installer, can install Gentoo the same way they did before, the way it's described in the classic Gentoo handbook.In order to update your Gentoo system you must do the following: synchronize the portage tree, emerge world. Issue the “emerge --sync” command in order to update the portage tree, “emerge --ud world” (this will take quite some time, make sure you're using ccache, if you can use some disk space for the cached data). The next step would be “emerge --depclean” and “revdep-rebuild” (which is a part of the gentoolkit, it has to be emerged before it is used).If you changed your USE flags, you'll have to “emerge -ud --newuse” instead of “emerge --ud world”, so support is added or removed, depending on the new set of USE flags.When I installed Gentoo, I used ccache and distcc to speed up the compilation process, still, the duration of the compilation process of KDE was very long, even with two computers, one Pentium 4 with hyper threading and one Athlon XP, which both had 1GB of RAM memory. Sometimes, distcc would compile on the second PC, but most of the time, it didn't compile anything. Passing the localhost on the second place in the list of hosts for distcc didn't seem to help either. It's best to exclude localhost and compile on other computers, that way the main computer will always be ready to serve new files for compilation.The KDE version that was available for compilation was 3.4.3, which is a lot older than the version most distributions use.**Why Gentoo?**Those who are busy, impatient or just don't want to go through all of this, should choose a ready to be installed distro. Gentoo may be easier to install now, but it's still not as up to date as some other Linux distributions and it still takes a lot of time in order to be updated. Those who like to perform updates daily and have lots of packages installed will have to get used to updating less often then they used to do on binary Linux distributions.Gentoo can sometimes be unstable, depending on the settings you make and the packages that you choose to install. Installing experimental packages, beta versions and other such things can make your system unstable. Trying to over-optimize your Gentoo system by tweaking the CFLAGS and adding unsafe, insane flags will definitely make your system unstable, errors and strange things can happen in this case.If you want to learn more about the development of Linux distributions and start developing yourself such a distribution, Gentoo can be the place where you can learn some of those things. Those who are just wondering what Gentoo is and can consider themselves as impatient persons, should not choose it. If you just want to test drive Gentoo and see how it is like, I don't recommend messing with your computer just for this. You could add another hard drive and perform all of these tests, or install Gentoo into a virtual machine software. The second option is recommended if you don't intend to run Gentoo as a real operating system.People who like to use Linux distributions which are up to date, distros that have GCC 4.1, the newest kernels, the newest KDE and don't want to wait for the packages to be compiled should use something like Fedora Core 5. The fans of Gentoo, old and new ones, people who want to learn and all those who want to start using Gentoo can start installing it right away. While they download this Live CD or the minimal CD, they can create a new user account on the Gentoo forums and, last but not least, have fun using Gentoo.**Distro target:** Desktop computers, Servers, OS Development, others**Supported platforms:** - Minimal cd: amd64, hppa, alpha, PPC 32bit, PPC 64bit, sparc64, x86- Live CD: x86**Recommended to:** Home users, Linux beginners, Developers, Advanced users, Experts**Operating system features:** Great **Performance:** Good**Hardware support:** Good**Operating System configuration:** Good **Documentation:** Great