DA TRADURRE

Configure your new Slackware System

We'll assume you've read the Installation Guide, and you have a clean install of Slackware on your machine that you're happy with.

This beginner's guide is meant to put you firmly on the Slackware path. If you installed Slackware for the first time, you may be daunted by the sight of the blinking cursor at a console login. Let this page guide you through the initial configuration of a freshly installed Slackware system.

Before we continue, it is important to realize that the Slackware package manager does not perform any dependency checks. If you are new to Slackware, then performing a full installation (with the possible exception of the KDEI series) could prevent a lot of problems later on.
The official Slackware recommendation [1)] is "*If you have the disk space, we encourage you to do a full installation for best results*".

Post Installation Overview

When Slackware starts for the first time after completing the installation and rebooting, you will notice that it boots to a console log in screen - not the graphical login screen you may expect from using other distributions. Do not let that discourage you. It is the first stage in a learning experience which will make you a *lot* more knowledgeable in Linux after as little as a few weeks.

The installation did not offer to create a user account. At this stage, there is only the "`root`" account. You should remember the root password, which you set at the very end of the installation procedure. Login as "root" now - you will find yourself at a "**#**" console root-prompt.

So now what? The "`root`" user is not the account which you are going to use as a matter of routine. Root is meant for system maintenance and configuration, software upgrades and the like.
The first thing to do is create a fresh user account for yourself, without the root privileges. After that, it is time to start considering the installation of "Proprietary Graphics Drivers" (if you own a Nvidia or Ati card), setting up a wireless network connection or starting a graphical desktop environment. There is a lot that you can do with Slackware! Let's start with the basics.

Create a User Account

The first thing you will need to do is create your own non-root user account. There are two ways you can do this, both from the console. The recommended way is to use Slackware's own interactive `adduser` script, thus:

```
# adduser
```

and follow the prompts. Read the user management page for more detail on the `adduser` script. You

can use the non-interactive standard Linux program `useradd` too:

```
# useradd -m -g users -G
wheel,floppy,audio,video,cdrom,plugdev,power,netdev,lp,scanner -s /bin/bash
slacker
```

Once that's done you can log in to your user account.
Log out of the root account (type `logout` at the root prompt) and then login using the new account you just created. Now come the really interesting adventures!

Configure a Package Manager

Now that you have Slackware running, you should consider spending a bit of time caring for your computer's good health. The software which was installed as part of the Slackware release you are running, may develop vulnerabilities over time. When those vulnerabilities are critical to the health of your computer, then Slackware will usually publish a patched version of the software package. These patched packages are made available online (in the `/patches` directory of the release) and announced on the Slackware Security mailing list.

You have various options in order to keep your Slackware installation up-to-date. It's not advised to make the process of applying security updates fully automatic, but it is possible to do so using a cron job.

slackpkg

Your best option is to use slackpkg, which is a package manager on top of Slackware's own pkgtools. Before you can use `slackpkg` you will need to define an online mirror from which it will download updates to your computer.

A list of available mirrors for your Slackware version can be found in this file:

```
/etc/slackpkg/mirrors
```

Open the file in a text editor such as `nano` or `vi` and uncomment a single mirror URL. Make sure that the URL mentions the release number for the version of Slackware you are running! Also, pick a mirror which is close to you or of which you know it is fast. When you have done that, you need to initialize slackpkg's database by running

```
# slackpkg update gpg
# slackpkg update
```

Note that package management is done as the "`root`" user!

You will need to update the `slackpkg` database from time to time, when you learn about the availability of new patches for your distribution. After updating the database you can let it download and install the updates. Again, see the slackpkg page for guidelines about the use of this tool.

Watching for Updated Packages

The Slackware Essentials book has a chapter about keeping up to date. It would be good if you read it now if you have not done so already.

- One way to look out for updated packages (patches) is to subscribe yourself to the Slackware Security mailing list and act when you read about new patches.
- Another way is to setup a script to check for updates once a day and make the script email you when updates are available.
  For this to work you need to have sendmail configured (although it usually runs out of the box) and know how to create a cron job. And of course, have a script that does the work.
  An example of such a script is rsync_slackware_patches.sh which watches the Slackware ChangeLog.txt for updates. You download the script, edit it to use your favorite mirror server and make it executable so that it can be used in a cron job:

```
# wget http://www.slackware.com/~alien/tools/rsync_slackware_patches.sh
-O /usr/local/bin/rsync_slackware_patches.sh
# chmod +x /usr/local/bin/rsync_slackware_patches.sh
```

  The script uses a couple of defaults which you may want to change to suit your environment - such as the location where the script will download the patches to.
  Simply run the script once, and see what it reports:

```
# /usr/local/bin/rsync_slackware_patches.sh
[rsync_slackware_patches.sh:] Syncing patches for slackware version
'13.37'.
[rsync_slackware_patches.sh:] Target directory
/home/ftp/pub/Linux/Slackware/slackware-13.37/patches does not exist!
[rsync_slackware_patches.sh:] Please create it first, and then re-run
this script.
```

  You notice that you will have to edit the script and define a local directory (*and create that directory too!*) for the script to use. When that is done, you should run the script once - for a first-time download of patches.
  Then you can use cron to run the script once a day. For instance, schedule the script to run at 05:33 every day, and let it check for updates to the 64-bit version of Slackware-13.37. Open the crontab editor by typing

```
crontab -e
```

  and then you add the following line to your cron table:

```
33 5 * * *      /usr/local/sbin/rsync_slackware_patches.sh -q -r 13.37
-a x86_64
```

  This command will be executed silently (meaning you will not get emailed) if no new patches are found. However when the script finds updates it will download them and email you the script's output. You will get an email like this:

```
[rsync_slackware_patches.sh:] New patches have arrived for Slackware
13.37 (x86_64)!

.....................................................................

0a1,10
> Mon Sep 10 20:26:44 UTC 2012
> patches/packages/seamonkey-2.12.1-x86_64-1_slack13.37.txz:  Upgraded.
>   This is a bugfix release.
> patches/packages/seamonkey-solibs-2.12.1-x86_64-1_slack13.37.txz:
Upgraded.
>   This is a bugfix release.
> +--------------------------+
> Sun Sep  9 19:11:35 UTC 2012
> patches/packages/mozilla-thunderbird-15.0.1-x86_64-1_slack13.37.txz:
Upgraded.
>   This is a bugfix release.
> +--------------------------+
```

And then you know you have to update slackpkg and make it install the latest patches. This gives you control over your updates (you decide when you update) while being automatically warned about the availability of new patches (which will already have been downloaded for you).

Configure your Network

If you installed the network packages, then at the end of the Slackware installation, you will have been asked a couple of simple questions, like:

- do you use DHCP;
- or else, what IP address do you want to use;
- what is your computer's hostname;
- do you have a (DNS) nameserver in the network?

All of these questions have resulted in the setup of a few network related configuration files.

- `/etc/rc.d/rc.inet1.conf`
  This is where the details for your network interfaces go. Slackware's `netconfig` tool will only configure your `eth0` interface. If you have additional network interfaces, you can edit the file with a text editor such as `nano` or `vi` and add you configuration details. There is a man page for this:

  ```
  man rc.inet1.conf
  ```

- `/etc/resolv.conf`
  This is where your nameserver and domain search list are added. If you use DHCP then the DHCP client will update the file. If you use static IP addresses, then you are supposed to edit the file yourself. There is a man page for this:

```
man resolv.conf
```

- `/etc/HOSTNAME`
  This is where your computer's hostname is defined.
- `/etc/hosts`
  This is where you will find a definition for your loopback interface which connects that to your hostname. You can add further hostname-to-IP-address mappings in this file if you do not use a DNS server or if you need specific mappings which the DNS server does not provide. There is a man page for this:

  ```
  man hosts
  ```

If you want to read in more detail about how to configure your network, have a look at this online comprehensive guide to networking in Slackware.

Traditional Network Configuration

Wired Network

To configure your wired network interface `eth0`, run (as root)

```
# netconfig
```

The same script which was run during the installation process.

`netconfig` only deals with the wired connection for `eth0`.

On the assumption that you configured your wired connection with `netconfig`, your network should be connected automatically without the need for any post-installation configuration.

If you didn't enter your network configuration details during installation, just run `netconfig` as root; then run

```
# /etc/rc.d/rc.inet1 eth0_start
```

and you should have a working network connection instantaneously.

Wireless

Most common wireless hardware is supported by Linux these days. You can search online if your wireless hardware is supported by 3rd parties that have written Linux drivers. If you want to know if your computer recognizes your wireless card, simply run

```
# iwconfig
```

as root. If that tool reports "*no wireless extensions*" for all your network interfaces then the kernel does not have a driver for your wireless card and you'll have to find one online.

As with the wired network interfaces, your wireless card is traditionally configured in the file `/etc/rc.d/rc.inet1.conf`. You can read a lot more about it in this wireless configuration guide. There is also the man page:

```
# man rc.inet1.conf
```

You will also need to take steps to include wireless security, whether WEP or WPA2. Unencrypted wireless connections are strongly discouraged. Note that WPA/WPA2 encryption is not configured just in `/etc/rc.d/rc.inet1.conf`, you will also need to edit `/etc/wpa_supplicant.conf` and add an encryption key there.

Wireless encryption issues, in particular for WPA, can be hard to troubleshoot. Some basic troubleshooting steps are detailed in the above networking guide, just in case you do not get your computer associated to the Access Point.

Graphical Network Configuration Services

Slackware currently has some alternatives to configure and monitor your network connections. These install a daemon (aka a background service) which will allow you to switch between wired and wireless connections easily. That makes them perfectly suited for mobile users. They come with graphical configuration utilities and do not depend on the traditional Slackware configuration files - *in fact, those files will cause conflicts if they contain network configuration*.

- You will find wicd in the *extra* section of the Slackware release tree (the word *extra* means that it is not part of the core distribution and will not have been installed as part of a full installation).
  After installing the wicd package, you have to make its init script executable so that the network daemon automatically starts at boot:

  ```
  # chmod +x /etc/rc.d/rc.wicd
  ```

  You can then configure your network using the graphical tool `wicd-client` or if you are running Slackware 14 you can use the KDE widget for wicd instead. For console lovers, there is also `wicd-curses` which offers the same configuration capabilities as the X-based counterparts.

- Starting with Slackware 14, there is also Networkmanager. It will be installed as part of a full install, but the network daemon will not be started by default. As with wicd, you have to make its init script executable:

  ```
  # chmod +x /etc/rc.d/rc.networkmanager
  ```

  which will make NetworkManager start at boot. You will have to configure NetworkManager using an X-based graphical utility.
  Slackware 14 includes a KDE widget for Networkmanager. If you are using another Desktop Environment like XFCE, you can install the Gnome network-manager-applet from SlackBuilds.org.

Switch to a generic kernel

It's recommended that you switch to Slackware's *generic* kernel. This is easy to do but there are a few steps to follow.

What is the difference between a “generic” kernel and the “huge” kernel which has been installed as the default kernel?
The “huge” kernel is essentially a kernel which has every hardware driver built in which you might need for a successful installation of your computer. Think of storage and (wired) network drivers, filesystem and encryption drivers and a lot more. All these built-in drivers result in a big kernel image (hence the name “huge”). When this kernel boots it will use up a lot your RAM (relatively speaking… with 1 GB of RAM you will not really be troubled by a few MB less RAM).
The “generic” kernel on the other hand, is a kernel which has virtually no drivers built in. All drivers will be loaded into RAM on demand. This will make your kernel's memory consumption lower and the boot process a bit faster. The smaller size allows for the use of an initial RAM disk or “initrd”. An initial RAMdisk is required in certain configurations, like software RAID, or a fully encrypted hard drive.
For now, you need to remember that a “huge” kernel will not support an intial RAM disk, but the “generic” kernel will. We go for maximum flexibility and use a “generic” kernel.

- You will need to create an initial RAM disk (“*initrd*” for short). The *initrd* functions as a temporary root file system during the intial stage of the kernel booting, and it helps get the actual root system mounted when your system boots. Run this, as root:

  ```
  # /usr/share/mkinitrd/mkinitrd_command_generator.sh
  ```

  This command will not actually *do* anything. It is informational only, and will output something like this - depending on your kernel version, your hardware configuration, the root filesystem you chose when you installed Slackware and so on:

  ```
  #
  # mkinitrd_command_generator.sh revision 1.45
  #
  # This script will now make a recommendation about the command to use
  # in case you require an initrd image to boot a kernel that does not
  # have support for your storage or root filesystem built in
  # (such as the Slackware 'generic' kernels').
  # A suitable 'mkinitrd' command will be:

  mkinitrd -c -k 3.2.29 -f ext4 -r /dev/sdb2 -m usb-storage:ehci-
  hcd:usbhid:ohci-hcd:mbcache:jbd2:ext4 -u -o /boot/initrd.gz
  ```

  Run the script's suggested `mkinitrd` commandline (as root) to generate the `initrd.gz` image.

- If you have installed LILO (the default bootloader of Slackware), then you will also need to make changes to its configuration file `/etc/lilo.conf` by adding a section to your Slackware entry as follows:

  ```
  image = /boot/vmlinuz-generic-3.2.29
    initrd = /boot/initrd.gz # add this line so that lilo sees initrd.gz
    root = /dev/sda1
  ```

```
  label = Slackware
  read-only
```

Actually, the "`mkinitrd_command_generator.sh`" script will show an example section which can be added to `/etc/lilo.conf` if you pass it the name of the generic kernel as an argument, like this:

```
# /usr/share/mkinitrd/mkinitrd_command_generator.sh -l /boot/vmlinuz-generic-3.2.29
```

Note that it is recommended to *add a new section* instead of editing the existing kernel image section. Assign a unique label to your new section. After reboot, LILO will give you two options: to boot into your freshly added generic kernel, or to boot into the failsafe huge kernel (of which you are certain that it will work).

- After making the changes to `/etc/lilo.conf` you have to save the file and then run

```
# lilo -v
```

to make your change permanent. Then, reboot.

- Have a look at `mkinitrd` manual page (`man mkinitrd`) for more information.
- If you use grub or another bootloader, then make changes which are applicable to the program you use.
- If you try to use the generic kernel without creating an `initrd.gz`, then booting will fail with a kernel panic.

Start a Graphical Desktop Environment

Configure X If Required

X.Org is the X-Window framework used in Slackware. The X server will usually auto-detect your graphics card and load applicable drivers. If auto-detect does not work (X crashes on startup), you will need to create a file `/etc/X11/xorg.conf` and set the correct options for your graphics card and display resolution. You can use

```
# X -configure
```

to generate a basic `xorg.conf` configuration file in your current directory. This file can then be customized and placed in the `/etc/X11/` directory. For a detailed overview of X configuration, check the `xorg.conf` manual page (`man xorg.conf`).

Non-free Display Drivers

Many people use computers with a modern graphics card powered by a Nvidia or Ati GPU (graphics processing unit). The vendors of these high-performance graphics card offer non-free (proprietary

binary-only) drivers for their cards. These binary-only drivers will boost your computer's graphical and in particular OpenGL performance. If you own such a card you may want to read our Wiki article "Proprietary Graphics Drivers".

Choosing a Desktop Environment/Window Manager

To choose the Window Manager or Desktop Environment you wish to use, run the `xwmconfig` utility:

```
$ xwmconfig
```

and select one of the available options. Note that you can run the `xwmconfig` command as the `root` user which will set a global default for all users. By running the same command as your ordinary user account, you override that global default and pick your own.
After making your choice you can simply run

```
$ startx
```

Your preferred Desktop Environment or Window Manager will then start up.

Graphical Login

To start with a graphical login screen on boot instead of Slackware's default console login, change the default runlevel to 4. Edit the file `/etc/inittab` and change the line that looks like

```
   id:3:initdefault:
```

to

```
   id:4:initdefault:
```

Note the difference from other Linux distributions; many of those use runlevel 5 for their graphical login. In Slackware, runlevel 5 is identical to runlevel 3 (console boot).

In the graphical runlevel, you will be greeted by one of the available display (login session) managers. Slackware will by default look for the availability of GDM (Gnome Display Manager), KDM (KDE Display Manager) and XDM (X Display Manager) - in that order. You can also install a third-party login manager like SliM but you will have to edit `/etc/rc.d/rc.4` and add a call to your new session manager all the way at the top.

Further Exploration

The Command Line

It may be of interest to new Linux users to explore the command line a bit more before installing a graphical desktop, just to learn some shell commands and applications available in non-graphical

mode. Slackware excels in having an abundance of command line programs for a wide range of tasks. For instance, web browsing can be done with lynx or links, which are console based web browsers. You can listen to music (even network audio streams) on the console using audio players like moc, mpg123, ogg123.

Mixing 64-bit with 32-bit

If you just installed the 64-bit version of Slackware (often called *slackware64* or *Slackware for x86_64*) you will soon discover that it will refuse to run 32-bit programs like Wine. You may want to read our page on adding multilib capabilities in that case.

Slackware Documentation

Even a Slackware user can benefit from good documentation (why else are you reading this?). Our suggestion is that you browse this Wiki for additional tips and HOWTOs. And don't forget to check out the root directory of the Slackware DVD or CD1! You'll find Slackware's own main documentation there. Every text file there is worth a read.

Upgrading the System

If you have been using Slackware for a while and want to upgrade to the next release once that becomes available, we have a nice HOWTO available here: Upgrading Slackware to a New Release

When tracking current, you should always read the latest ChangeLog.txt before upgrading the system, to see whether any additional steps are required to be performed before or after upgrading. For upgrades to a stable release, it is a good idea to read the `UPGRADE.TXT` and `CHANGES_AND_HINTS.TXT` files located on the CD/DVD or the official mirror.

slackware, beginners, guide

1) see the Slackware-HOWTO in the root of the DVD or CD1