Amithlon Kernel

Starter Guide
Documentation by Tom Crews (SnkBitten)

This is less of a guide and more of a "how I did it" to get you started. I'm not a linux user nor am I a programmer. I had previously pulled Google's source code for Android and modified it for my own custom Rom on my phone back in the Android 2.2x days.   That required a linux system and highlights the extent of my linux knowledge :)

Requirements:
Linux system with the appropriate (older) compiler tools. GCC 3.2.3, I believe from my readings, is the last known version to compile without errors.   I use Oracle's Virtualbox and CentOS 3 i386 as it had everything at the appropriate levels to compile Milan's (milanca) Kernel 4 tree without much work. If you are using Virtualbox, make sure the HD is configured as IDE and not SATA, it will not see the SATA device when you boot the CD to install CentOS..

CentOS 3 i386 https://mirror.hmc.edu/centos/3/isos/i386/
Install CentOS 3 using the .ISO's you downloaded and then update it. I used a custom install and added the kernel development package.

Boot of the first CD (or .iso) and press Enter to boot the graphical install
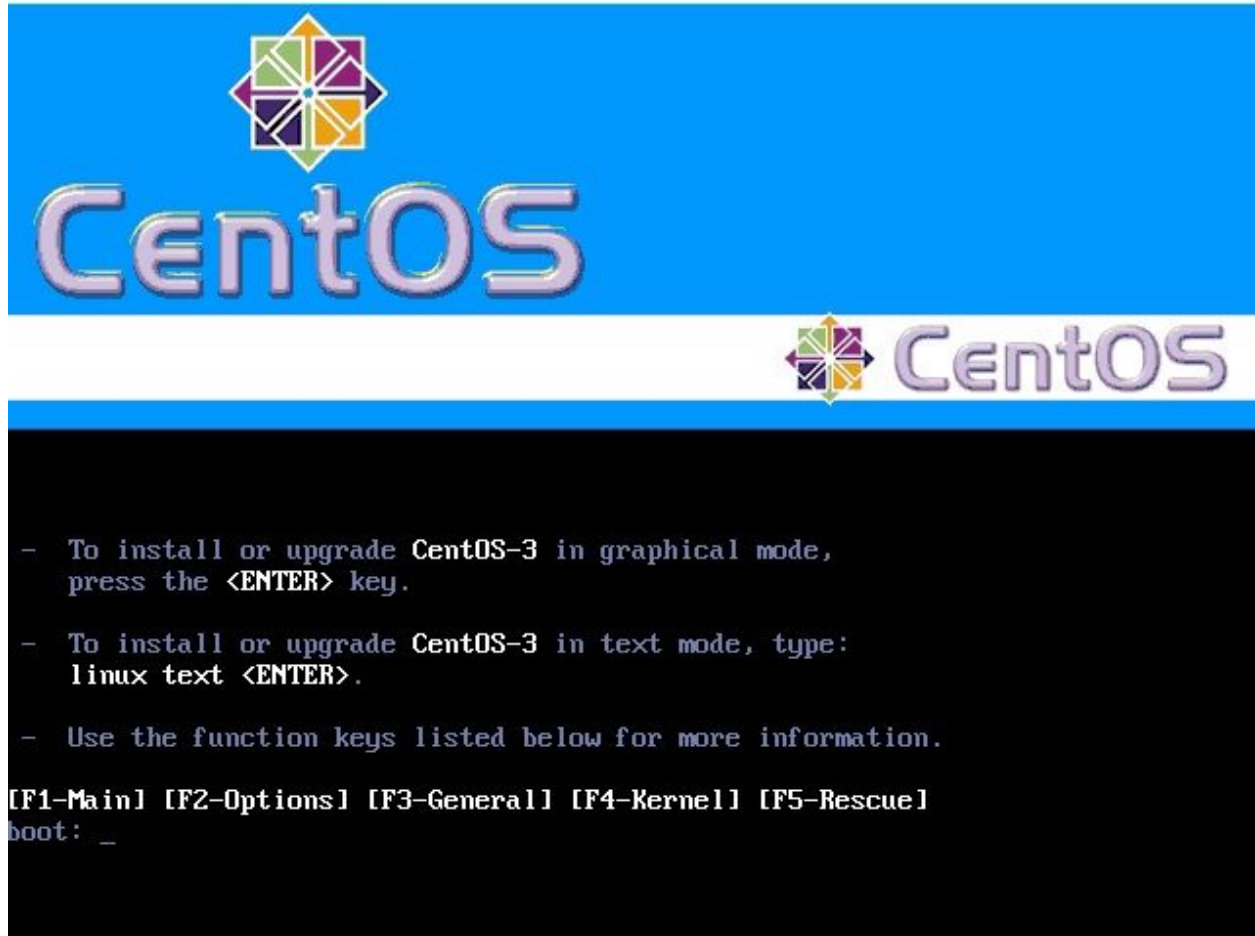
(first boot off CentOS-3 disk(.iso) 1 of 3, press enter to choose the first option)

You can skip the CD media test to start the installation.
I'm using .iso files in Virtualbox so I'm not too worried about checking the media of my my install files. You can do the check if you like but I think most would be fine skipping and moving directly to installing CentOS-3.

Choose "Custom" as your install type.
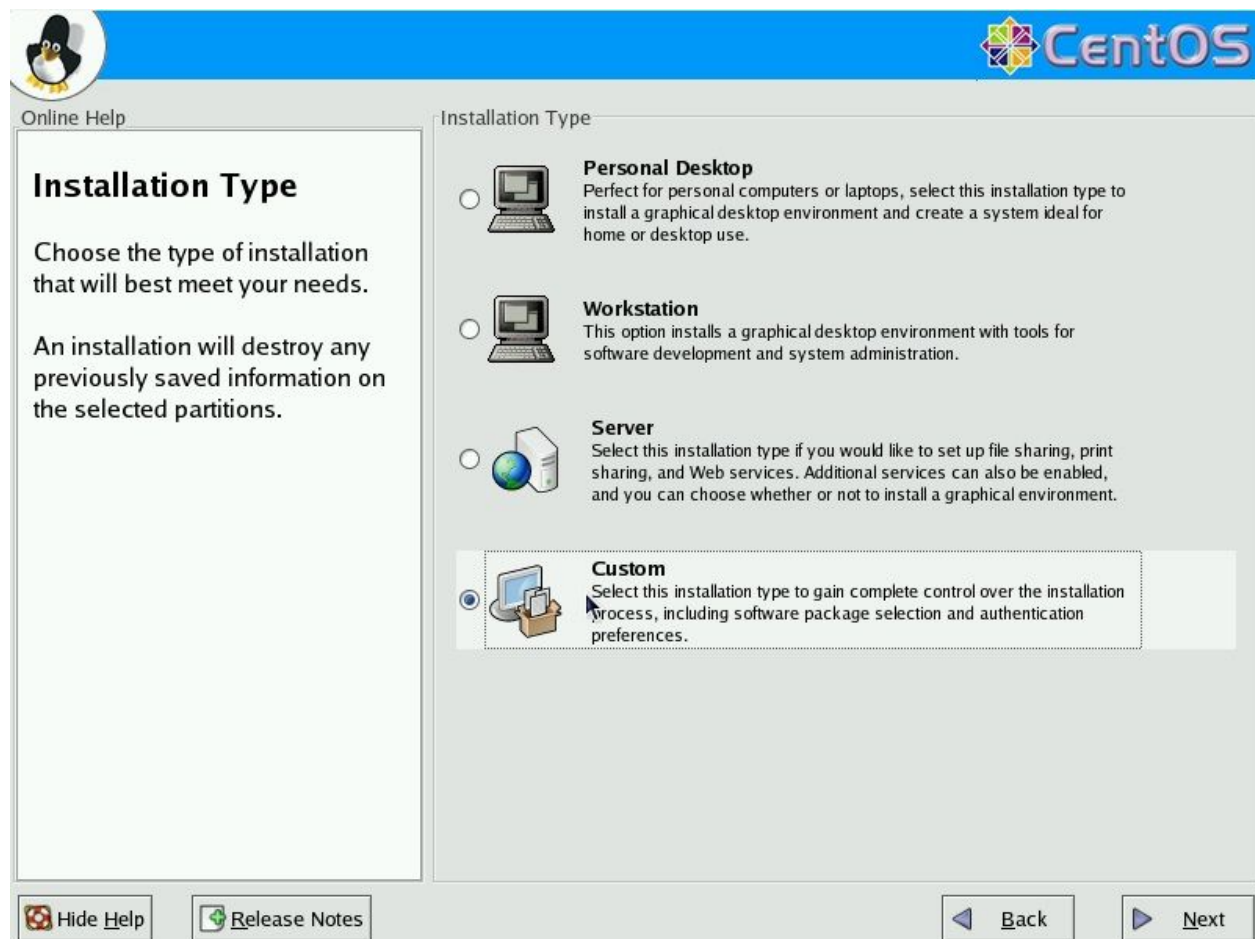
(choosing Custom for the install type)

I let it automatically create my partitions for me as I just want to deal with compiling a kernel for Amithlon. I select "Next" leaving the defaults and eventually get to the Network Configuration screen. I again use the defaults and choose "Next", I however, do turn off the firewall as this is not a system that runs and I want to make sure I have connectivity to my NAS box, the web and anything else I may need quickly.

Package Group Selection:
Desktops:
Choose what you like for Desktops, I use the defaults.
Applications:
I add Editors and leave the rest at default settings
Servers: (default, no selections)
Development:
I add Development Tools and Kernel Development
System:
I add System Tools and leave the rest as default

Miscellaneous: (default, no selections)

Now complete the installs, swapping disc (.iso) as needed.

Graphical Interface
(X) Configuration
I use the VESA driver, choose an unprobed monitor LCD that fits the monitor I'm using and use the Graphical boot option (with just a 1024x768x24 bit screen).

(first boot screen)

I don't create an account as I login as root, but you can if you prefer. This again is not my primary system and is only used when compiling a kernel. When asked for Additional CDs I just click Next.

CentOS will reboot and present the login screen, I type root but if you want to use an account you created, enter that here.
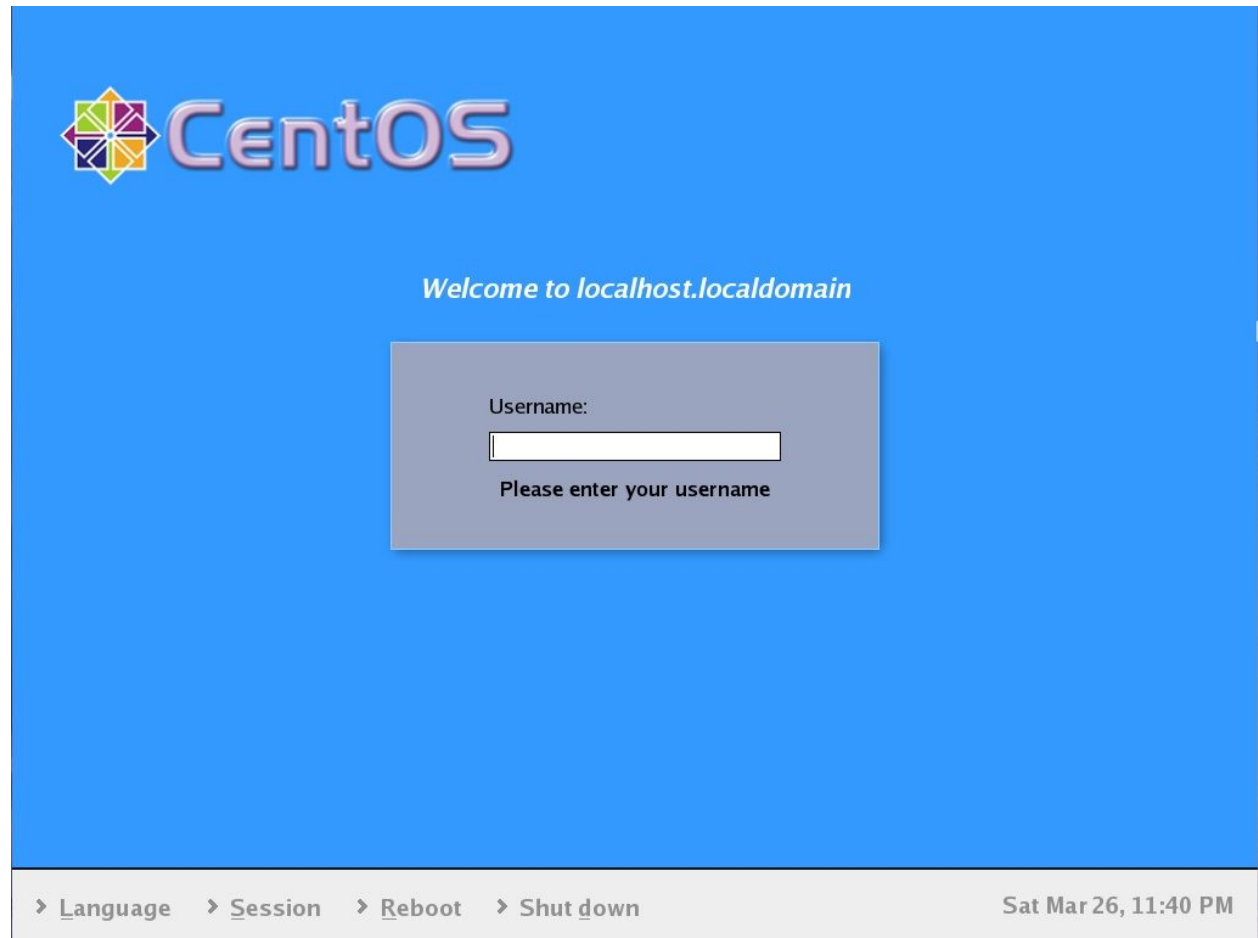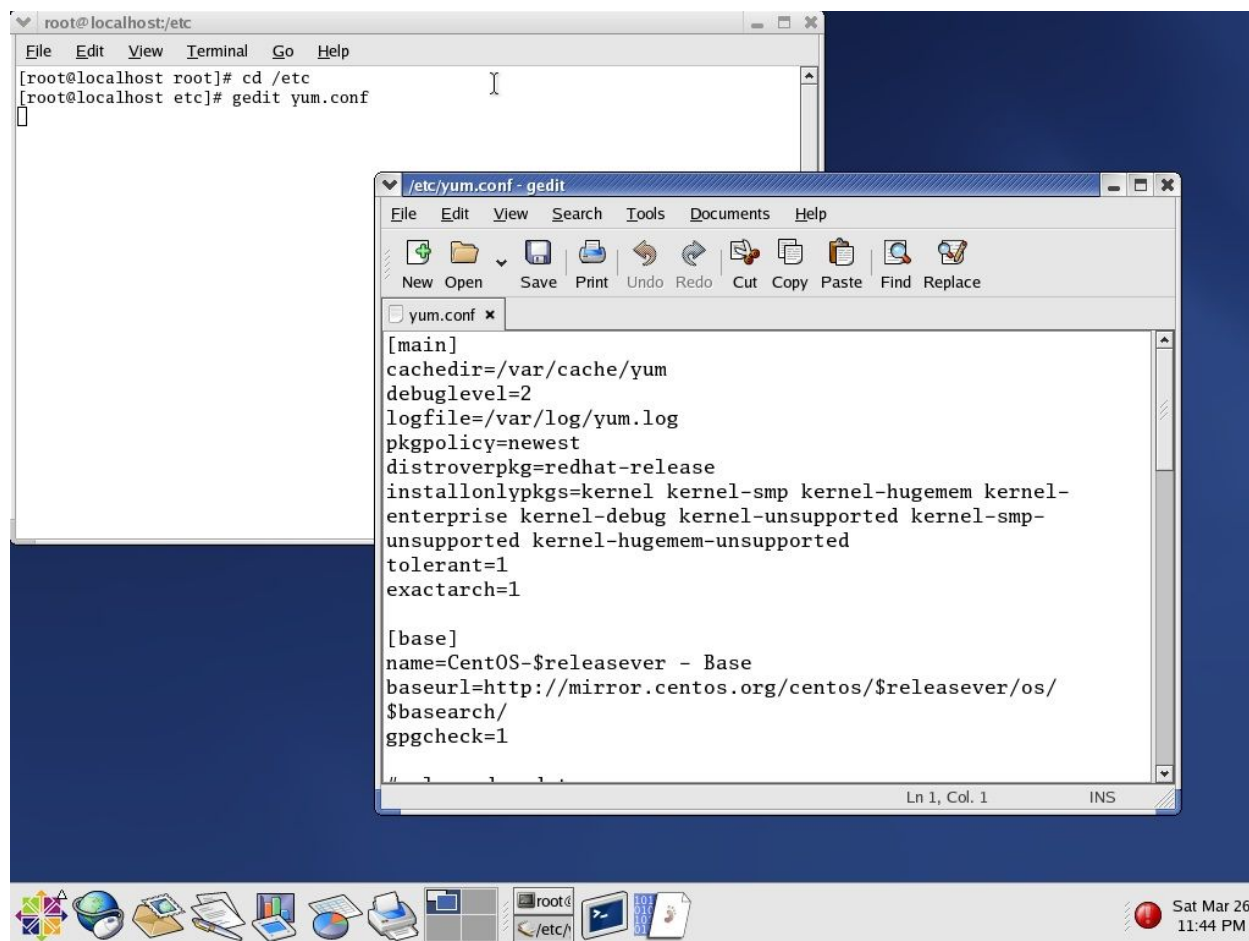
(CentOS login screen)

Now we need to get update….
You'll need to edit the config file /etc/yum.conf for where the sources/updates are pulled from,
as the official mirrors don't have this old version still live.
Launch a terminal and change directory to /etc
cd /etc
gedit yum.conf

(gedit /etc/yum.conf file to pull updates)

Replace the default of "base", "updates", "addons" and "extras"
http://mirror.centos.org/centos/$releasever/ with

https://mirror.hmc.edu/centos/3/os/$basearch/
https://mirror.hmc.edu/centos/3/addons/$basearch/
https://mirror.hmc.edu/centos3/extras/$basearch/
Make note that you are changing from http:// to https://

then
su -c 'rpm --import http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-3'

Next type
sudo yum update

This will check for updates from the location we configured in yum.conf
You will have to answer "y" after it checks for all the updates to install them.
This does take a while so do something else for a bit or just be patient.

I would reboot after this, it may not be needed but why not.

Download Milan's (milanca) 2.4.37.9 [kernel tree](#) and extract it to your home folder.

I login as root so I just created folders in /home, ie… /home/amithlon.
Using the above directory, I downloaded the kernel tree to /root and then in the terminal typed the following:

cd /home/amithlon
tar -xvjf /root/linux-2.4.37.9-amithlon

This moves you to the home folder and then extracts the downloaded linux-2.4.37.9-amithlon.tar.bz2 file to a folder named linux-2.4.37.9-amithlon in the folder /home/amithlon/.

cd linux-2.4.37.9-amithlon
Do this to save you some time and effort…
cp .config ../
This copies the .config file to the amithlon folder
make clean mrproper
make clean config
make mrproper
cp ../.config ./
This copies the .config back to our current directory. This way we don't have to walk through and make all the choices (proper choices) to build the kernel.
Now type
make oldconfig
This will create a .config file for your build using the settings configured in the "old" .config file.

__Note - I went through this process manually until this occurred to me. Manual process detailed below which uses 'make config' .__
Make config will create a new .config file for the options used in compiling.   This is a long and boring process. You step through each option and choose, Y, N or M (Yes, No, Module) depending on its option and your choice. If you choose wrong, you have to start over.
*I used the [diff file](#) from Milan to view the changes made in .config, I scrolled the the diff file in a text editor while I ran "make config" on the linux box and made sure I selected the same options.***

Once you have a .config file created, you can use 'make menuconfig' which will bring up a menu interface to make changes. This allows you to pick individual options vs. having to walk through the entire list of options, saving you lots of frustration. Be knowledgeable about your choices!

Once you have completed the .config process do the following.

make dep

To build the dependencies.

Now let's compile the kernel.

make bzImage

This compiles the kernel and creates the bzImage file which will be located in /home/amithlon/linux-2.4.37.9-amithlon/arch/i386/boot, you can rename this to what you want, mine is "snkkern" and it would replace "kernel4" in your "small" file. Of course make sure you copy your "bzImage" (or whatever you named it) file to your Amithlon's DOS boot partition.

Modules for the kernel haven't been built, so now you can do a

make modules

This builds the modules.

Once completed building the modules do a

make modules_install

which will create the module files to be used by insmod (or pciinsmod) at /lib/2.4.37.9/modules/kernel/drivers

In this folder here should be a net, sound and char folder created with the appropriate modules inside.   Zip them up and copy them over to your Amithlon system (they go in devs:pcidrivers)

That's it. You've compiled the kernel and tested, if it's working successfully you can start changing what is included in the kernel and what is set as a module or just include what you specifically have in your Amithlon system and don't add/build any modules…(config and choose "*" not "m" as the option for devices you want, "*" is built in, "m" is build as module) for your particular Nic and sound card. Boot with this kernel and you won't need any drivers for your network card or soundcard, no need for pciinsmod or insmod use.

You should spend a lot of time looking at the .config file (just cd /home/amithlon/linux-2.4.37.9-amithlon and then gedit .config but don't make or save change, just look at the file. There are a LOT of options but removing or adding just one can make your compiled kernel fail to boot Amithlon.