

# Instalación de Linux Desde Cero

## Contenido

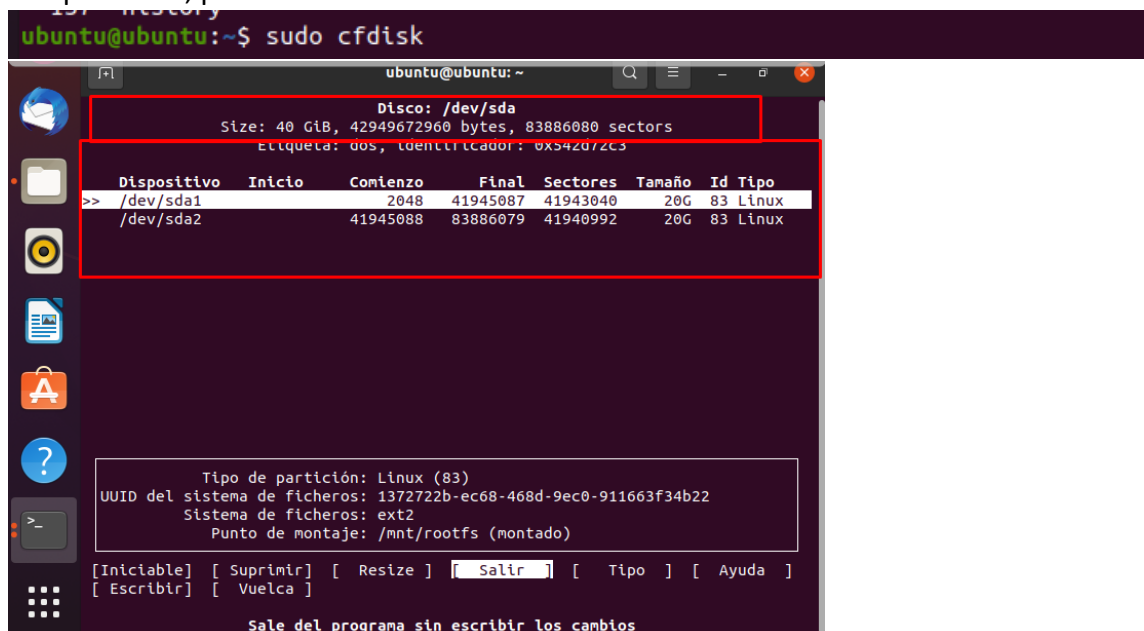
Redimensionamiento de los discos para la maquina virtual.....	3
Instalación de los paquetes base .....	5
Configuración de del LILO el Bash Shell .....	6
Entorno Básico .....	8
Seguimos con paquetes base no tan importantes como los anteriores.....	11
Instalación de los últimos paquetes y modificación del LILO.....	15

## Redimensionamiento de los discos para la maquina virtual

1. Arrancamos una máquina virtual en Ubuntu y hacemos Ubuntu Live para proseguir con la instalación, con el espacio que hemos delimitado anteriormente, en mi caso son 40 GB de memoria.



2. Ahora tenemos que dividir nuestro disco en dos particiones sda1 y sda2 ambas son primarias, para ello utilizamos el comando sudo cfdisk.



3. Hacemos un `sudo cfdisk /dev/sda` para comprobar que todo esta correctamente.

```

Disco: /dev/sda
Size: 40 GiB, 42949672960 bytes, 83886080 sectors
Etiqueta: dos, identificador: 0x542d72c3

Dispositivo  Inicio  Comienzo  Final  Sectores  Tamaño  Id Tipo
--> /dev/sda1      2048    41945087  41943040    20G    83 Linux
    /dev/sda2      41945088  83886079  41940992    20G    83 Linux

Ayuda
Tipo de partición: Linux (83)
UUID del sistema de ficheros: 1372722b-ec68-468d-9ec0-911663f34b22
Sistema de ficheros: ext2
Punto de montaje: /mnt/rootfs (montado)

[Inicial] [Suprimir] [Resize] [Salir] [Tipo] [Ayuda]
[Escribir] [Vuelca]

Sale del programa sin escribir los cambios
  
```

4. Siguiendo con la practica montamos los discos, para darles formato, para ello usaremos los siguientes comandos:

```

-sudo mkdir /mnt/rootfs/

- sudo mkdir /mnt/packages/

-sudo mount /dev/sda1
- sudo mount /dev/sda1 /mnt/rootfs/
-sudo mount /dev/sda2 /mnt/packages/
-sudo mkfs /dev/sda1
- sudo mkfs /dev/sda2
  
```

Capturas de los comandos.

```

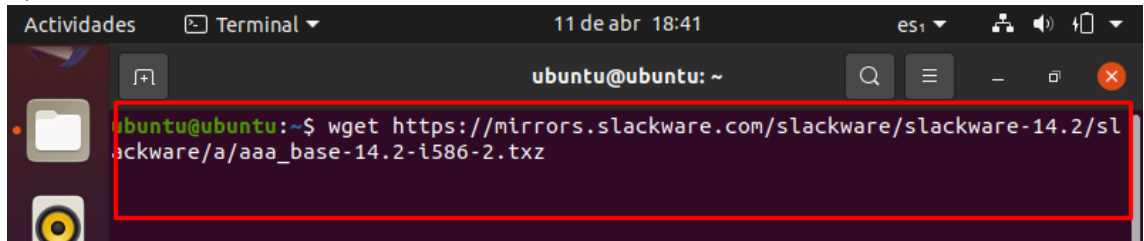
sudo mount /dev/sda1
sudo mount /dev/sda1 /mnt/rootfs/
sudo mkfs /dev/sda1
sudo mkfs /dev/sda2

ubuntu@ubuntu:~$ sudo mkdir /mnt/rootfs/

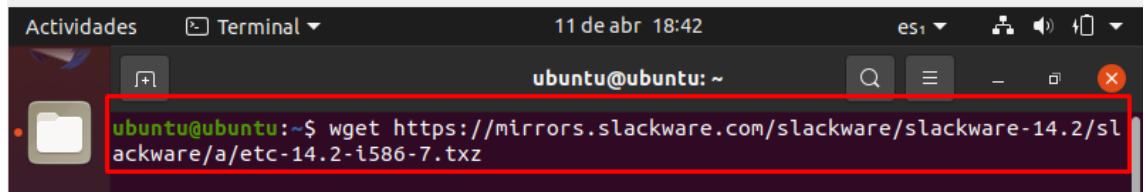
ubuntu@ubuntu:~$ sudo mkdir /mnt/packages/
  
```

## Instalación de los paquetes base

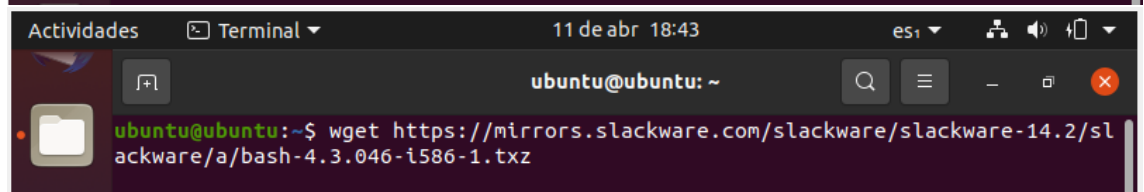
1. Tenemos que hacer un WGET de todos los paquetes a instalar, que no son muchos en este punto, después los descomprimos los archivos con la herramienta TAR con las opciones Jxvf.



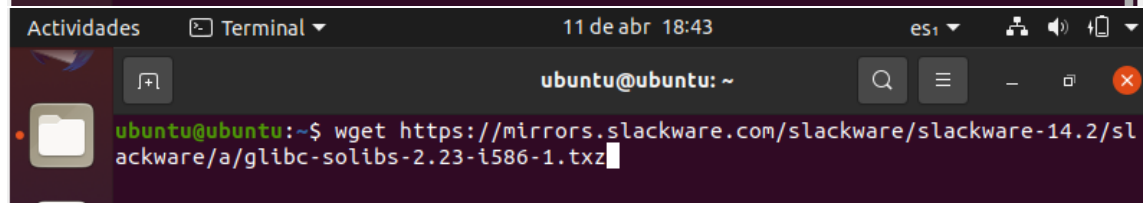
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/aaa_base-14.2-i586-2.txz
```



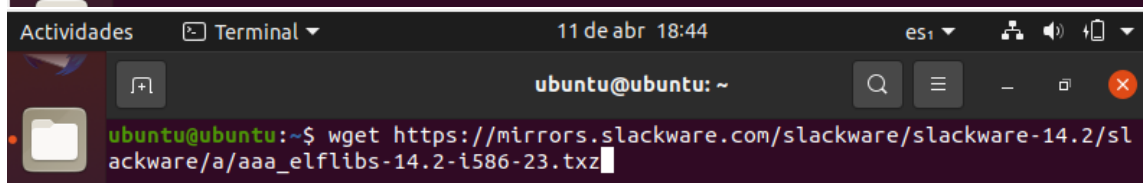
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/etc-14.2-i586-7.txz
```



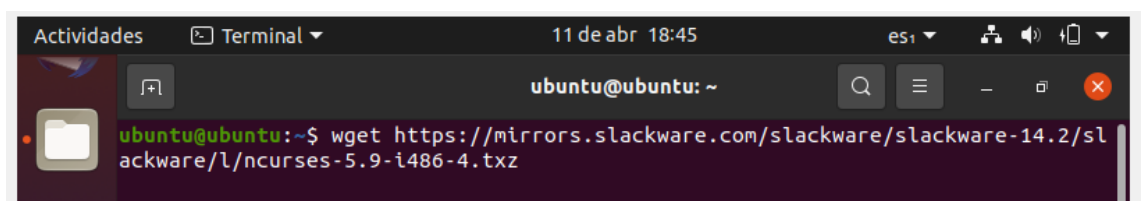
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/bash-4.3.046-i586-1.txz
```



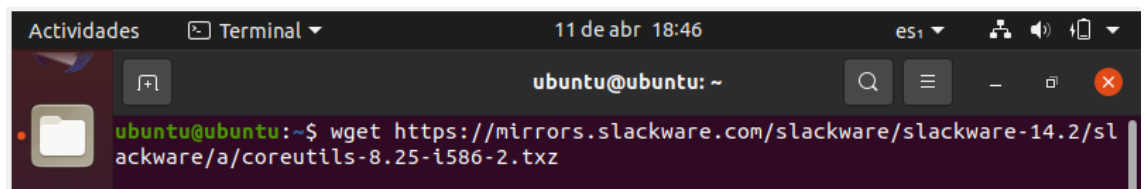
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/glibc-solibs-2.23-i586-1.txz
```



```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/aaa_elflibs-14.2-i586-23.txz
```

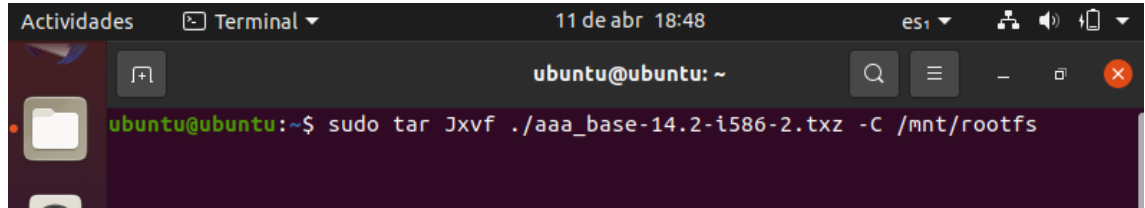


```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/l/ncurses-5.9-i486-4.txz
```



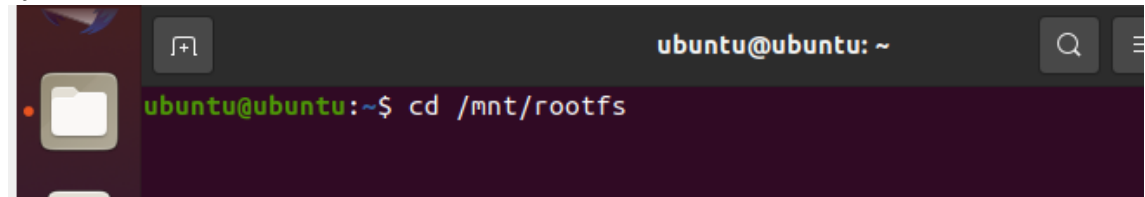
```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/coreutils-8.25-i586-2.txz
```

Aquí tienes como se descomprimen los archivos con tar

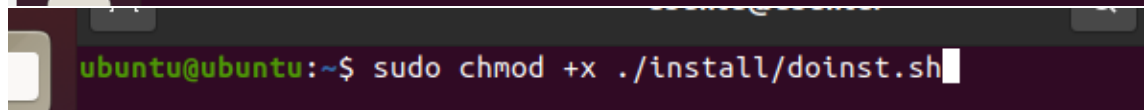


```
Actividades Terminal 11 de abr 18:48 es1
ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ sudo tar Jxvf ./aaa_base-14.2-i586-2.txz -C /mnt/rootfs
```

En este punto tenemos que ir descomprimiendo las cosas cambiando el nombre de los archivos y entrando en la ruta /mnt/rootfs y dándole permisos de ejecución y ejecutándolo como administrador con sudo.

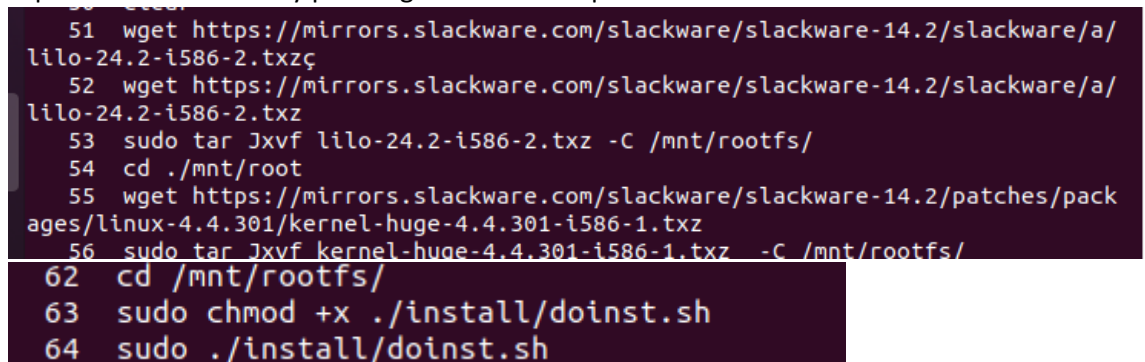


```
ubuntu@ubuntu:~$ cd /mnt/rootfs
```



```
ubuntu@ubuntu:~$ sudo chmod +x ./install/doinst.sh
```

Aquí instalamos el LILO y para el gestor de arranque.



```
51 wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/
lilo-24.2-i586-2.txz
52 wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/
lilo-24.2-i586-2.txz
53 sudo tar Jxvf lilo-24.2-i586-2.txz -C /mnt/rootfs/
54 cd /mnt/root
55 wget https://mirrors.slackware.com/slackware/slackware-14.2/patches/pack
ages/linux-4.4.301/kernel-huge-4.4.301-i586-1.txz
56 sudo tar Jxvf kernel-huge-4.4.301-i586-1.txz -C /mnt/rootfs/
62 cd /mnt/rootfs/
63 sudo chmod +x ./install/doinst.sh
64 sudo ./install/doinst.sh
```

## Configuración de del LILO el Bash Shell

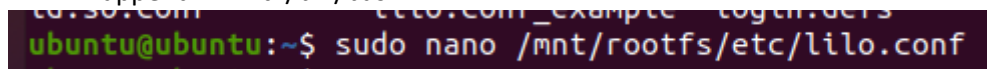
1. Primero entramos en el archivo de configuración de lilo.conf, para ello usaremos el sudo nano con la ruta del mismo, además tendremos que ir al final del archivo y escribir lo siguiente:

Image = /boot/vmlinuz-huge

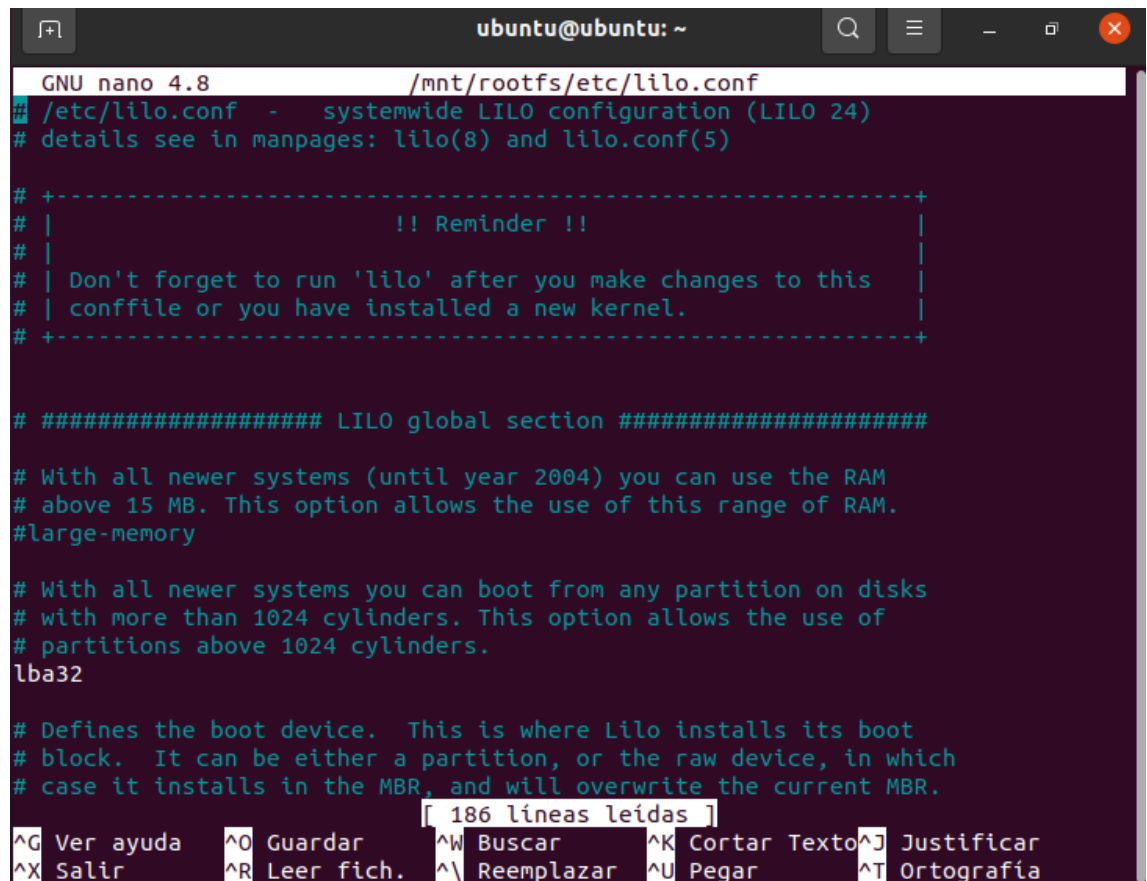
label = "Bash Shell"

root = /dev/sda1

append = "init=/bin/bash"



```
ubuntu@ubuntu:~$ sudo nano /mnt/rootfs/etc/lilo.conf
```



```
GNU nano 4.8 /mnt/rootfs/etc/lilo.conf
/etc/lilo.conf - systemwide LILO configuration (LILO 24)
# details see in manpages: lilo(8) and lilo.conf(5)

# +-----+
# |                !! Reminder !!                |
# | Don't forget to run 'lilo' after you make changes to this |
# | conf file or you have installed a new kernel.          |
# +-----+

# ##### LILO global section #####

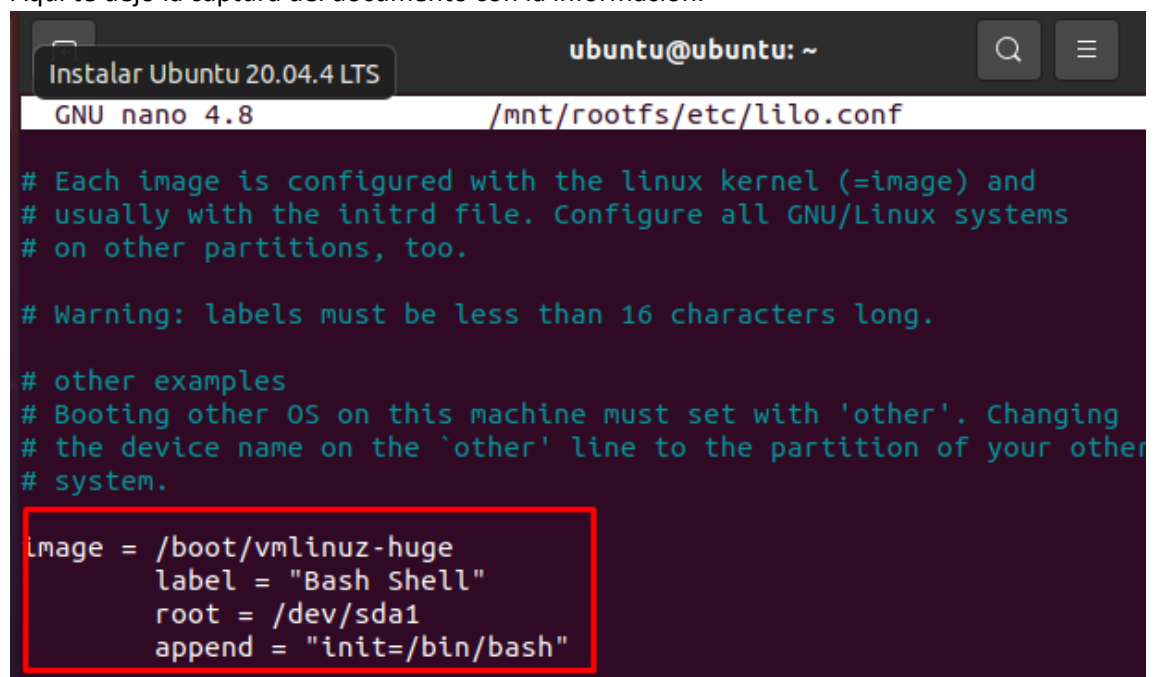
# With all newer systems (until year 2004) you can use the RAM
# above 15 MB. This option allows the use of this range of RAM.
#large-memory

# With all newer systems you can boot from any partition on disks
# with more than 1024 cylinders. This option allows the use of
# partitions above 1024 cylinders.
lba32

# Defines the boot device. This is where Lilo installs its boot
# block. It can be either a partition, or the raw device, in which
# case it installs in the MBR, and will overwrite the current MBR.

^G Ver ayuda  ^O Guardar   ^W Buscar    ^K Cortar Texto ^J Justificar
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar       ^T Ortografia
```

Aquí te dejo la captura del documento con la información.



```
Instalar Ubuntu 20.04.4 LTS
GNU nano 4.8 /mnt/rootfs/etc/lilo.conf

# Each image is configured with the linux kernel (=image) and
# usually with the initrd file. Configure all GNU/Linux systems
# on other partitions, too.

# Warning: labels must be less than 16 characters long.

# other examples
# Booting other OS on this machine must set with 'other'. Changing
# the device name on the 'other' line to the partition of your other
# system.

image = /boot/vmlinuz-huge
label = "Bash Shell"
root = /dev/sda1
append = "init=/bin/bash"
```

Después de esto tenemos que hacer un mount `--bind /dev /mnt/rootfs/dev` después el `proc` y seguidamente `sys` tal y como en la captura.

```
sudo mount --bind /dev /mnt/rootfs/dev/  
sudo mount --bind /proc /mnt/rootfs/proc  
sudo mount --bind /sys /mnt/rootfs/sys
```

## Entorno Básico

Para este punto lo que hice fue hacer un script para hacer los `wget` de cada paquete que es tedioso.

```
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/aaa_base-14.2-i586-2.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/etc-14.2-i586-7.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/bash-4.3.046-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/glibc-solibs-2.23-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/aaa_elflibs-14.2-i586-23.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/l/ncurses-5.9-i486-4.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/coreutils-8.25-i586-2.txz
```

Esto es los paquetes en orden de descomprimir, descomprimir el archivo `kernel-firmware`, descomprimir el archivo `kmod`, y todos los paquetes que tenemos encima para instalarlos son los mismos pasos siempre con cuidado ya que sino podemos cargarnos la máquina.

```
sudo tar Jxvf kernel-modules-4.4.14-i586-1.txz -C /mnt/rootfs
```

```
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu
```

```
sudo tar Jxvf kernel-firmware-20160628git-noarch-1.txz -C /mnt/rootfs
```

```
#install the binary kernel firmware  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu
```

```
sudo tar Jxvf kmod-22-i586-1.txz -C /mnt/rootfs
```

```
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu
```



```
sudo tar Jxvf sysvinit-2.88dsf-i486-4.txz -C /mnt/rootfs
```

```
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
```

```
sudo tar Jxvf shadow-4.2.1-i486-1.txz -C /mnt/rootfs
```

```
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
```

```
sudo tar Jxvf getty-ps-2.1.0b-i486-2.txz -C /mnt/rootfs
```

```
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
```

Configuramos el lilo.conf.

Image = /boot/vmlinuz-huge

Label = "Linux huge"

Root = /dev/sda1

```
image = /boot/vmlinuz-huge
label = "linux huge"
root = /dev/sda1
```

```
sudo mount --bind /dev /mnt/rootfs/dev/
sudo mount --bind /proc /mnt/rootfs/proc
sudo mount --bind /sys /mnt/rootfs/sys
```

Ahora vamos a instalar los paquetes de red.

```
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware-
ls-1.60.20120726git-i486-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackwar
-scripts-14.2-noarch-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackwar
-s20140519-i586-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackwar
2-4.4.0-i586-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackwar
6.8.2-i586-2.txz
```

```
sudo tar Jxvf net-tools-1.60.20120726git-i486-1.txz -C /mnt/rootfs
```

```
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
```

```
#descompilamos el archivo network-scripts
sudo tar Jxvf network-scripts-14.2-noarch-1.txz -C /mnt/rootfs
```

```
#instalamos el archivo network-scripts
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
```

```
sudo tar Jxvf iputils-s20140519-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf iproute2-4.4.0-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf dhcpcd-6.8.2-i586-2.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
```

Seguidamente vamos a editar los archivos de configuración de red para que podamos conectarnos bien a la descarga de cosas.

```
GNU nano 4.8 /mnt/rootfs/etc/rc.d/rc.modules.local
#!/bin/sh

# /etc/rc.d/rc.modules.local

# The Linux kernel source is the best place to look for documentation
# for the many available kernel modules. This can be found under
# /usr/src/linux-$VERSION/Documentation/.

# Almost all necessary modules are automatically loaded when needed,
# but there are a few exceptions. Here's a (not all-inclusive) list
# so uncomment any of the below entries or add others as needed:
# Note that you could also create/edit rc.modules-$version if you
# only wanted specific modules loaded for particular kernels.

#/sbin/modprobe tun           # Universal TUN/TAP device driver
#/sbin/modprobe sg            # Generic SCSI support for SATA DVD-R
/sbin/modprobe e1000
```

```
ubuntu@ubuntu:~$ sudo nano /mnt/rootfs/etc/rc.d/rc.inet.conf  
sudo nano /mnt/rootfs/etc/rc.d/rc.inet
```

Seguimos con paquetes base no tan importantes como los anteriores

En este momento vamos a instalar los mas opcionales entre ellos tenemos el TAR, el que descomprime archivos, el vim, python etc.

```
ubuntu@ubuntu:~$ sudo cat instalar_paquetes_opcionales.sh  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/ap/vim-7.4.1938-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/d/perl-5.22.2-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/d/python-2.7.11-i586-2.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/ap/nano-2.6.0-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/file-5.25-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/tar-1.29-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/gzip-1.8-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/bzip2-1.0.6-i486-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/xz-5.2.2-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/ap/man-1.6g-i586-3.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/ap/groff-1.22.3-i586-2.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/less-481-i586-1.txz  
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/ap/slackp
```

```
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/ap/slackp
kg-2.82.1-noarch-3.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/pkgtool
s-14.2-noarch-10.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/finduti
ls-4.4.2-i486-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/bin-11.
1-i486-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/dialog-
1.2_20130523-i486-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/sed-4.2
.2-i486-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/grep-2.
25-i586-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/which-2
.21-i586-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/gawk-4.
1.3-i586-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/n/gnupg-1
.4.20-i586-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/n/wget-1.
18-i586-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/l/libunis
tring-0.9.3-i486-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/a/util-li
nux-2.27.1-i586-1.txz
wget https://mirrors.slackware.com/slackware/slackware-14.2/slackware/n/openssl
-1.0.2h-i586-1.txz
```

Se descomprime los archivos y los ejecutamos

```
sudo tar Jxvf vim-7.4.1938-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf perl-5.22.2-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf python-2.7.11-i586-2.txz -C /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf nano-2.6.0-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf file-5.25-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf tar-1.29-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
```

```
sudo tar Jxvf gzip-1.8-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf bzip2-1.0.6-i486-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf xz-5.2.2-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf man-1.6g-i586-3.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf groff-1.22.3-i586-2.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf less-481-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
```

```
ubuntu@ubuntu: ~  
sudo tar Jxvf gawk-4.1.3-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf gnupg-1.4.20-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf wget-1.18-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf libunistring-0.9.3-i486-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf util-linux-2.27.1-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf openssl-1.0.2h-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu
```

```
ubuntu@ubuntu: ~  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf python-2.7.11-i586-2.txz -C /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf nano-2.6.0-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf file-5.25-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf tar-1.29-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf gzip-1.8-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf bzip2-1.0.6-i486-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo tar Jxvf kernel-generic-4.4.14-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf mkinitrd-1.4.10-i586-1_slack14.2.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
sudo tar Jxvf cpio-2.12-i586-1.txz -C /mnt/rootfs  
cd /mnt/rootfs  
sudo chmod +x ./install/doinst.sh  
sudo ./install/doinst.sh  
cd /home/ubuntu  
ubuntu@ubuntu:~$
```



## Instalación de los últimos paquetes y modificación del LILO.

En este punto debemos instalar los últimos paquetes necesarios para que la practica este terminada, para ello usaremos el wget, el tar los permisos de ejecución

```
sudo tar Jxvf e2fsprogs-1.43.1-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf procps-ng-3.3.11-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf pciutils-3.4.1-i586-2.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
sudo tar Jxvf usbutils-008-i586-1.txz -C /mnt/rootfs
cd /mnt/rootfs
sudo chmod +x ./install/doinst.sh
sudo ./install/doinst.sh
cd /home/ubuntu
ubuntu@ubuntu:~$
```