*By GHALEB*
Published: 2008-06-12 19:31

# Installing An Ubuntu Hardy 8.04 LTS DNS Server With BIND

***Pleasenote that my main reference and source is Falko's article "The Perfect Server - Ubuntu 8.04 LTS" here*** with more DNS details.

***Version 1.0***
***Author: Mohamed Ghaleb <Mohamed_Ghaleb [at] msn [dot]com> (English and German only please)***
***Last edited 06/03/2008***

This tutorial shows how to set up an Ubuntu Hardy Heron(Ubuntu 8.04LTS) based server that offers DNS services. This tutorial is writtenfor the32-bit version of Ubuntu 8.04 LTS, but should apply to the 64-bitversion.

I will use the following software:

- DNS Server: BIND9

I want to say first that this is not the only way of settingup sucha system. There are many ways of achieving this goal but this is theway I take. I do not issue any guarantee that this will work for you!

## *1 Requirements*

To install such a system you will need the following:

- the Ubuntu 8.04 LTS server CD, available here: ***ftp://releases.ubuntu.com/releases/hardy/ubuntu-8.04-server-i386.iso***
- a fast internet connection.

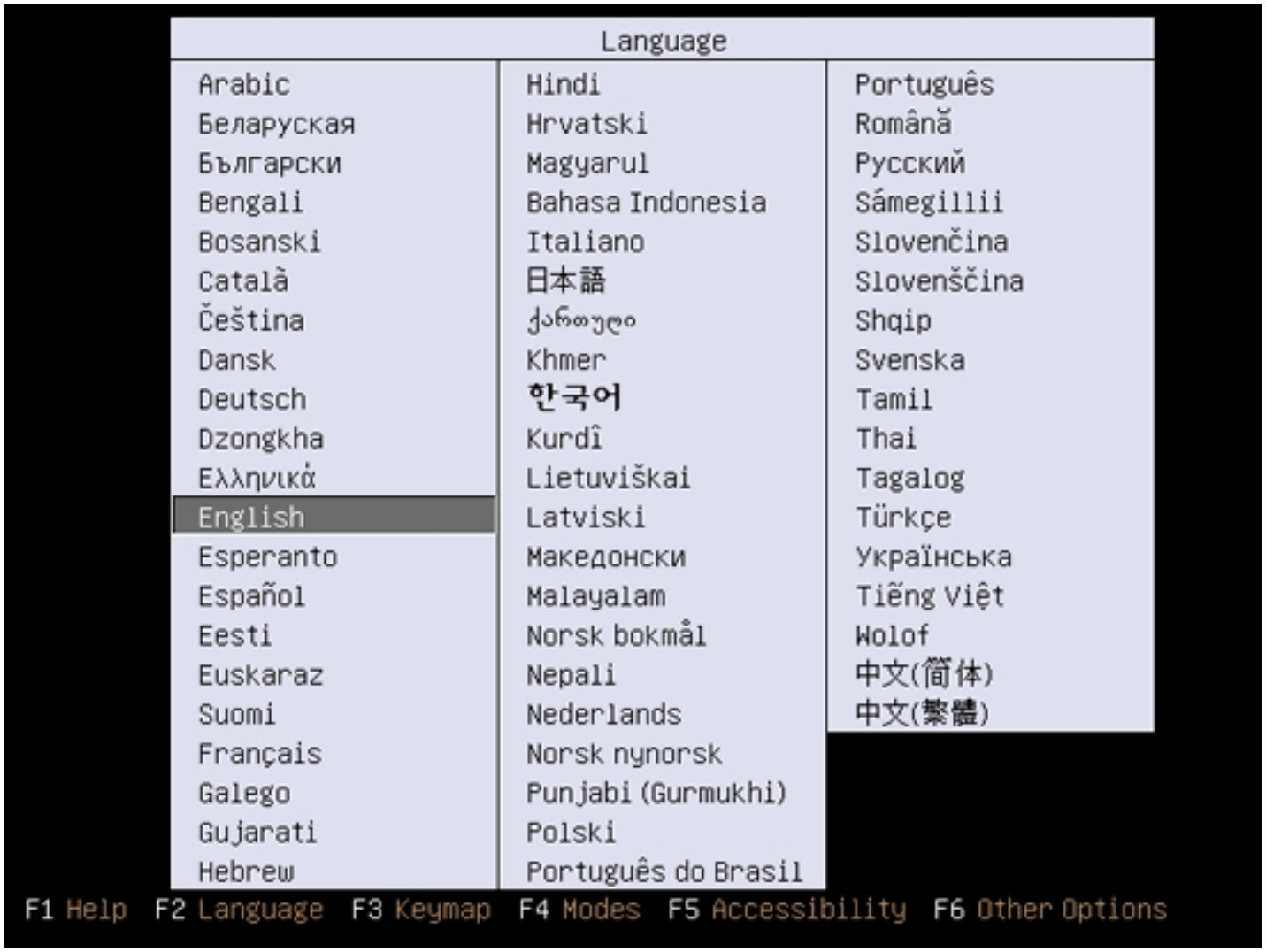
## *2 Preliminary Note*

In this tutorial I use the hostname `server1.tm.local` with the IPaddress `192.168.0.100` andthe gateway `192.168.0.1`..These settings might differ for

you, so you have to replace them whereappropriate.

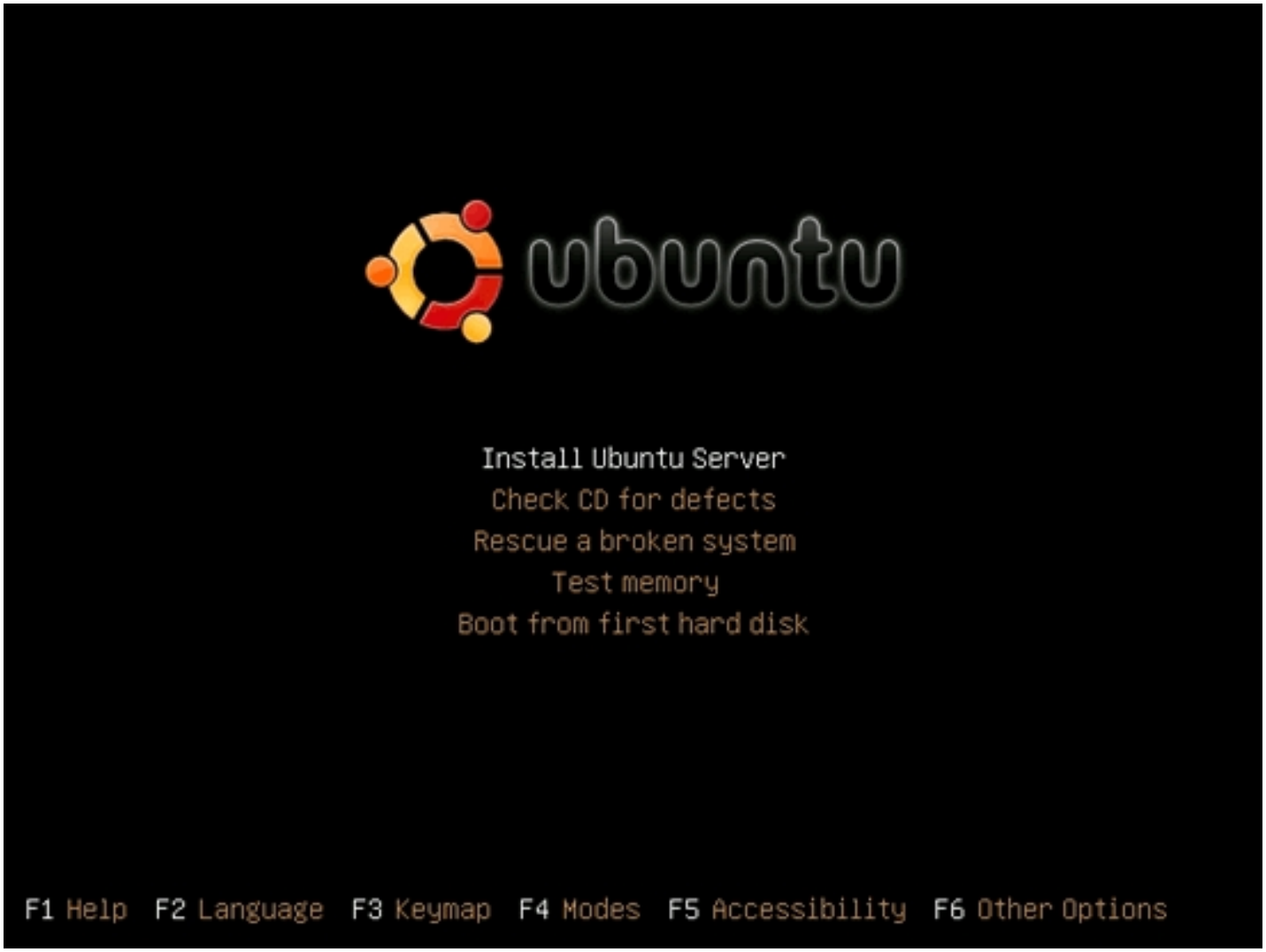
## 3 The Base System

Insert your Ubuntu install CD into your system and boot fromit. Select your language:



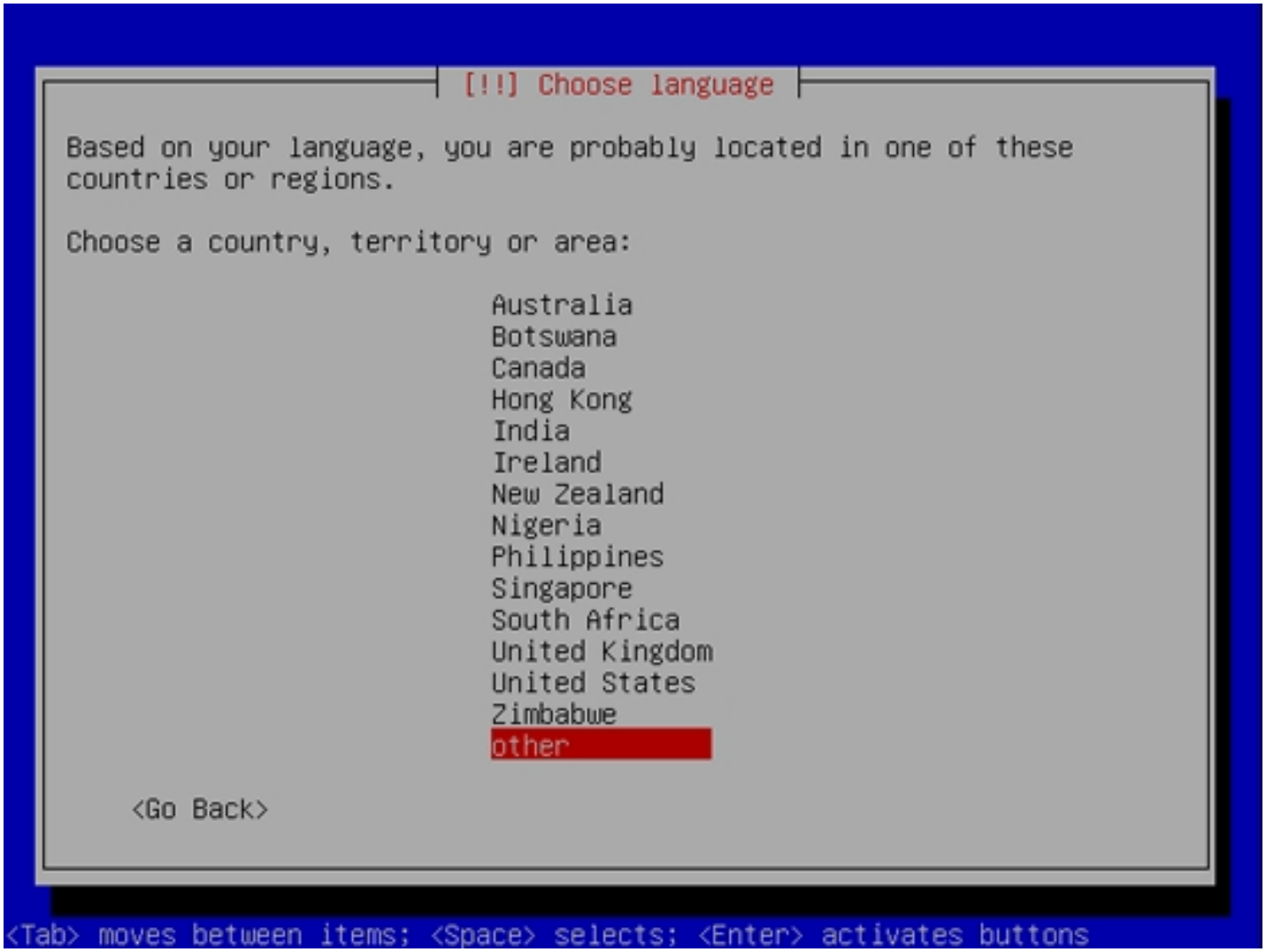

Then select *Install UbuntuServer*:

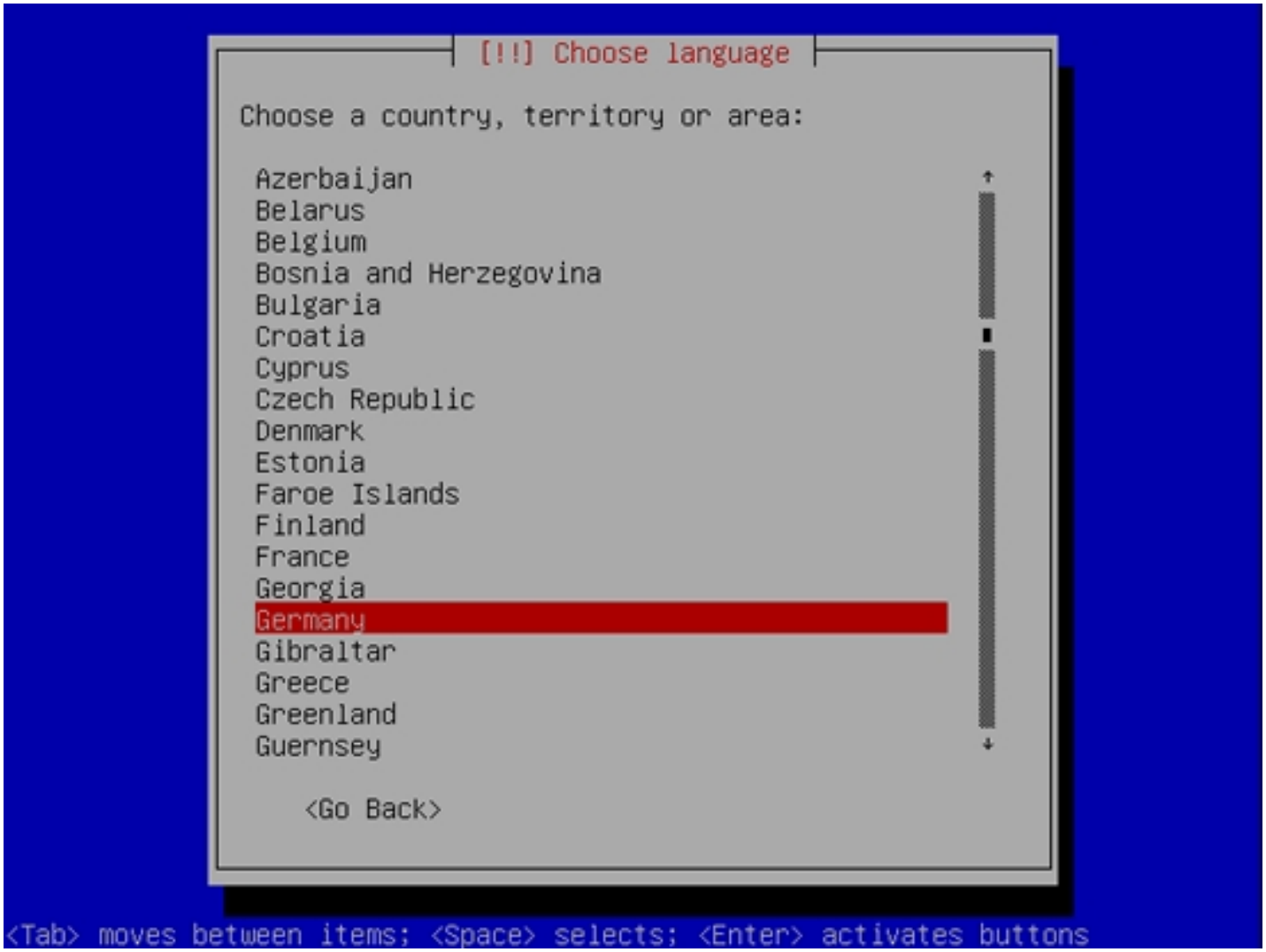
 



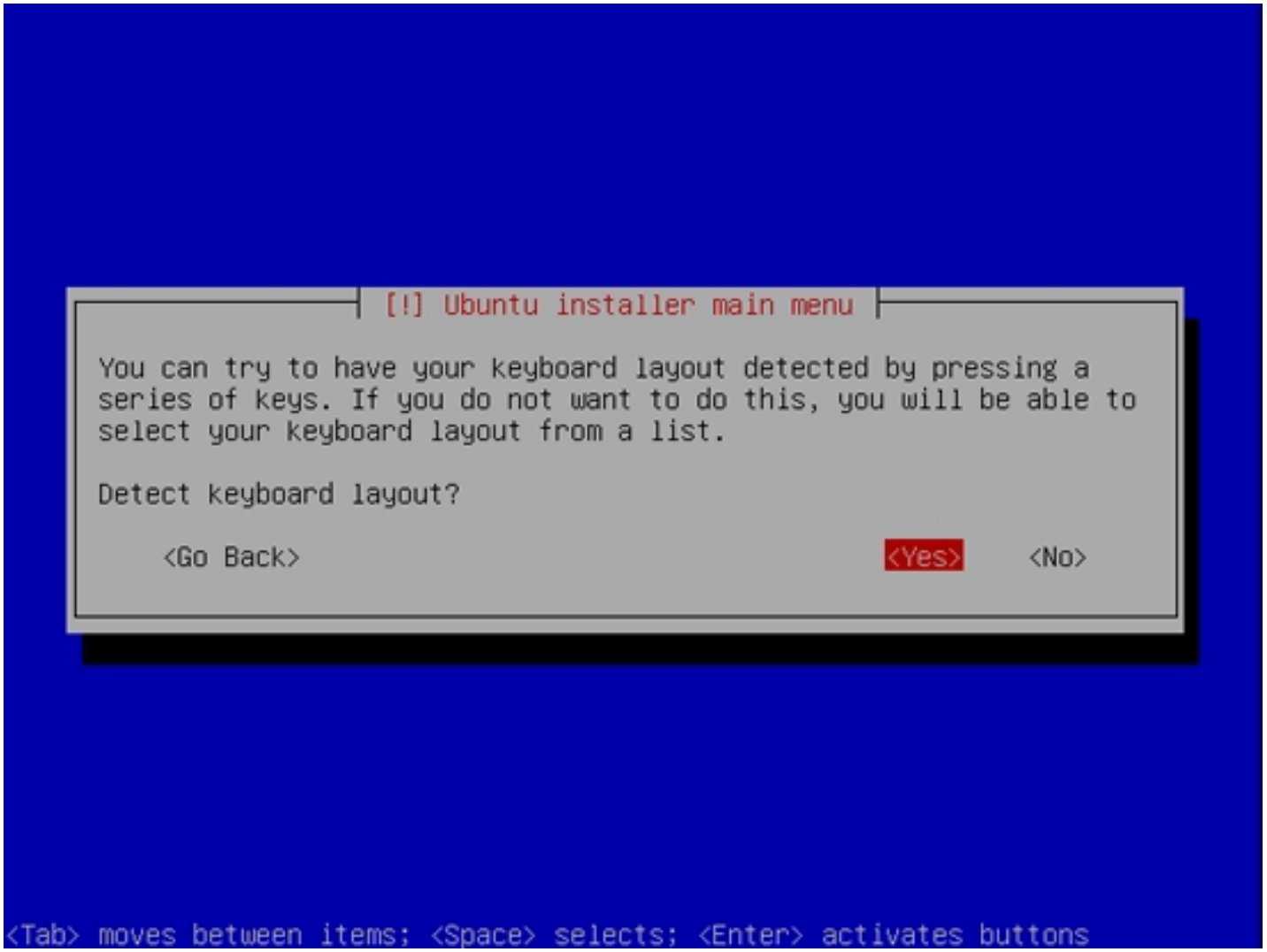

Choose your language again (?):

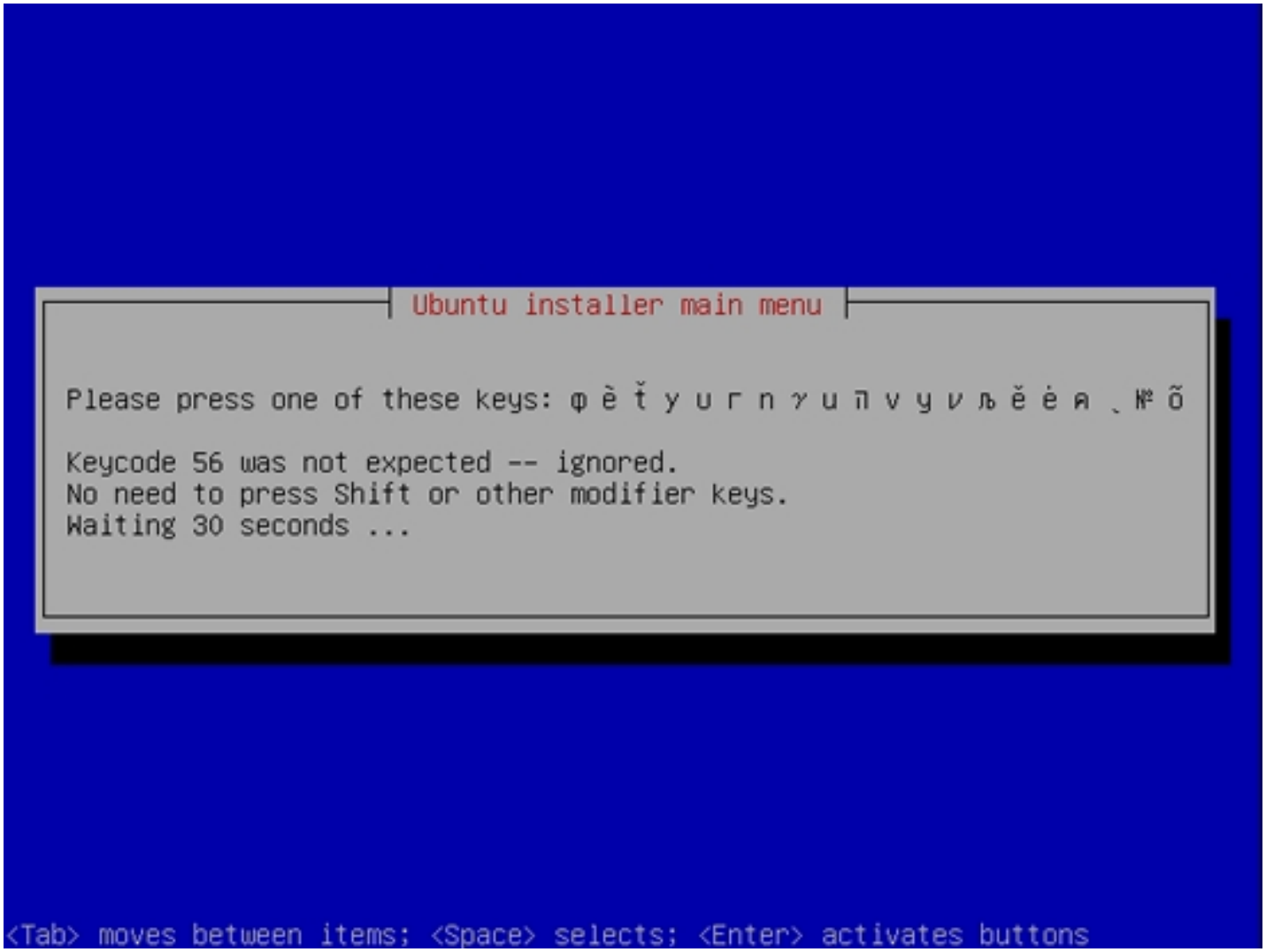

Then select your location:

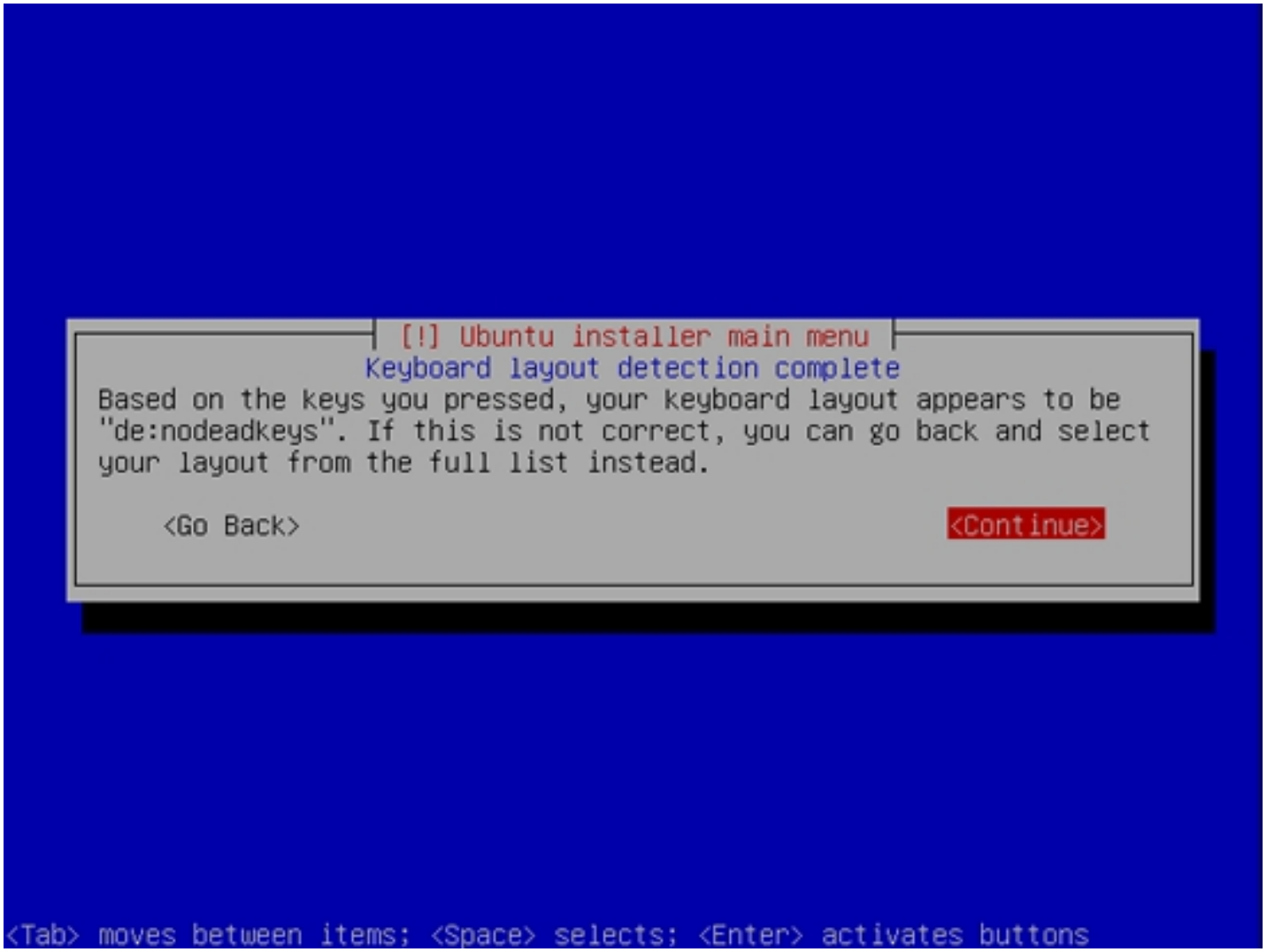




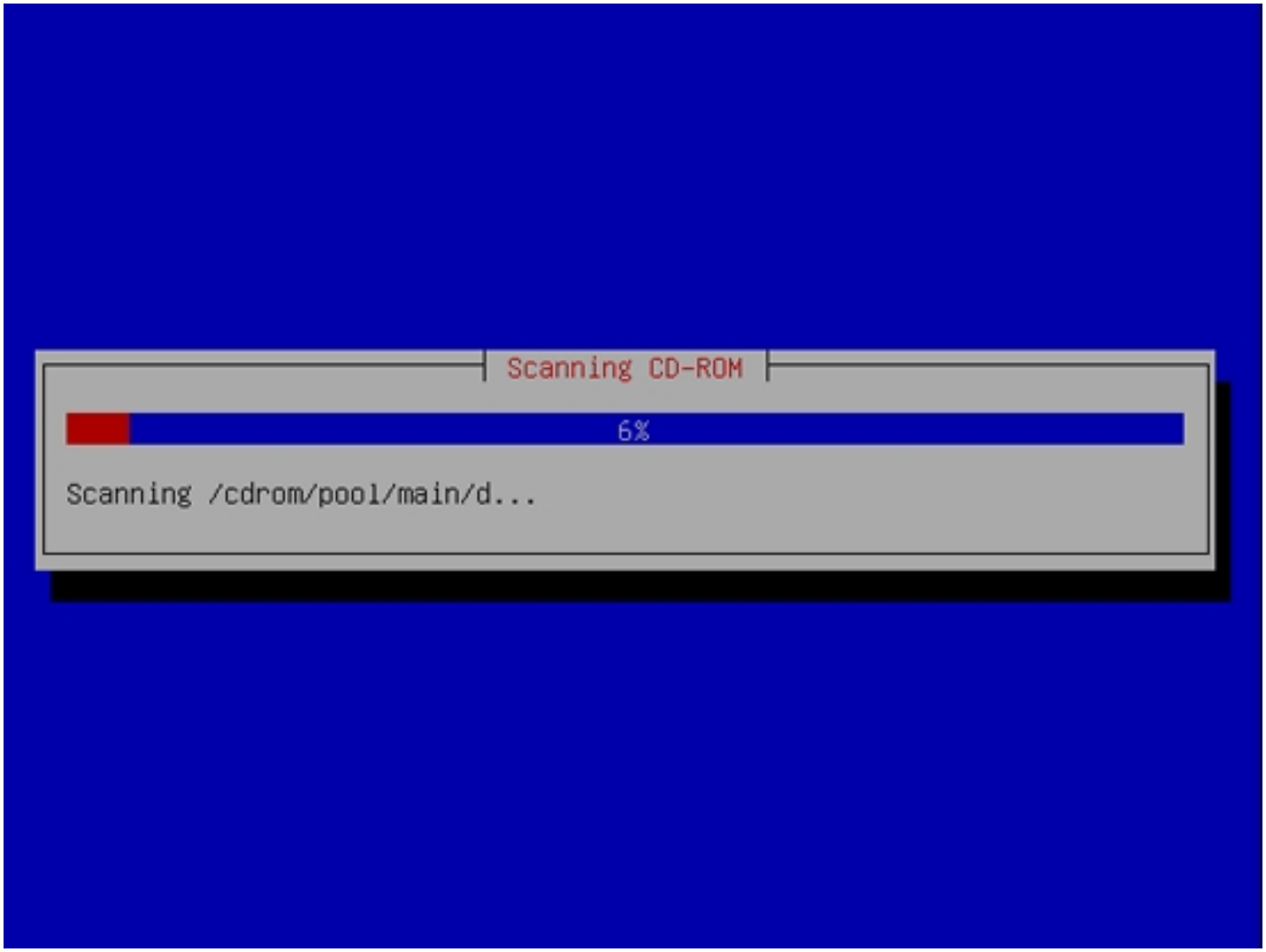

Choose a keyboard layout (you will be asked to press a fewkeys, andthe installer will try to detect your keyboard layout based on the keysyou pressed):

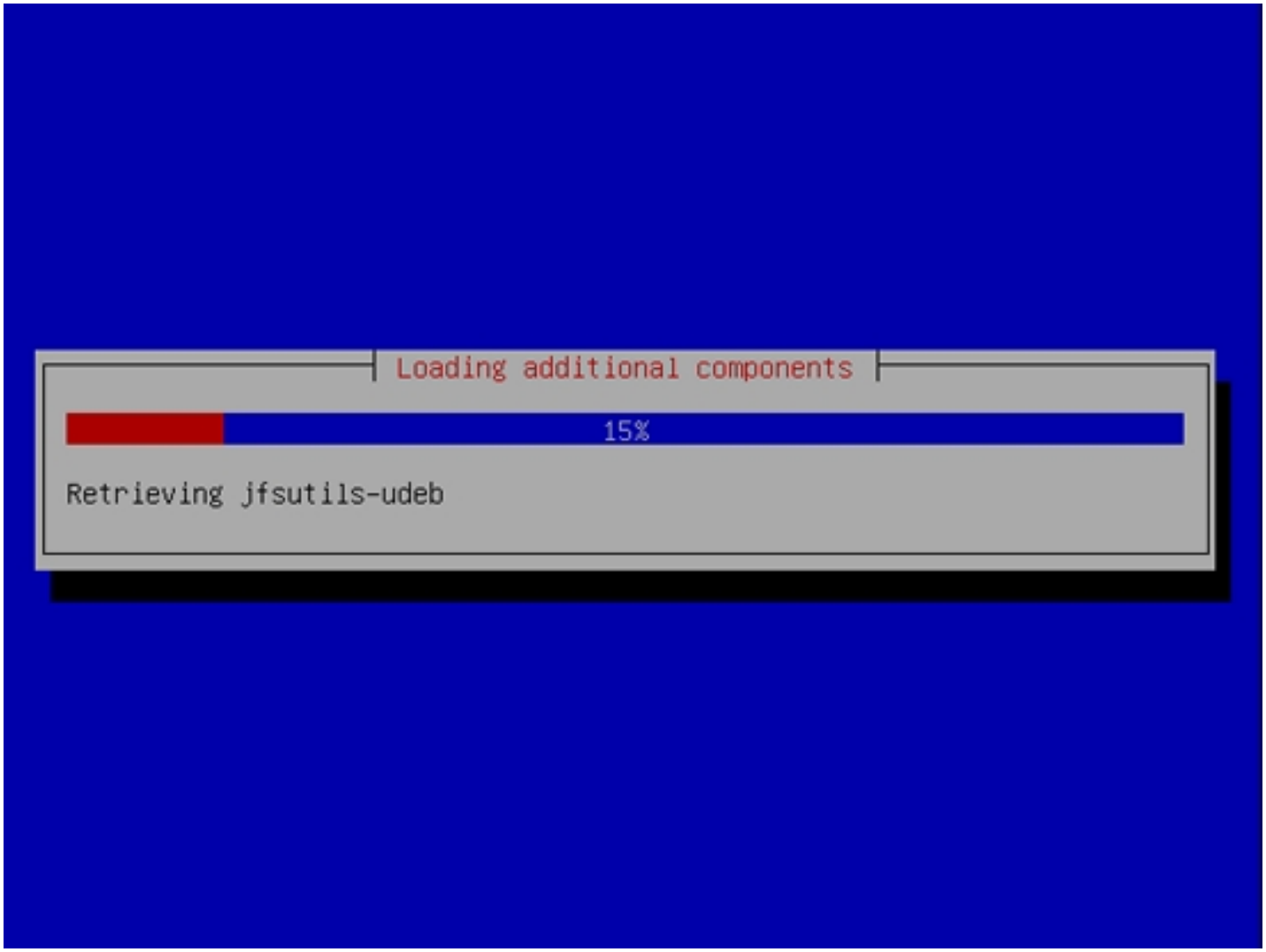


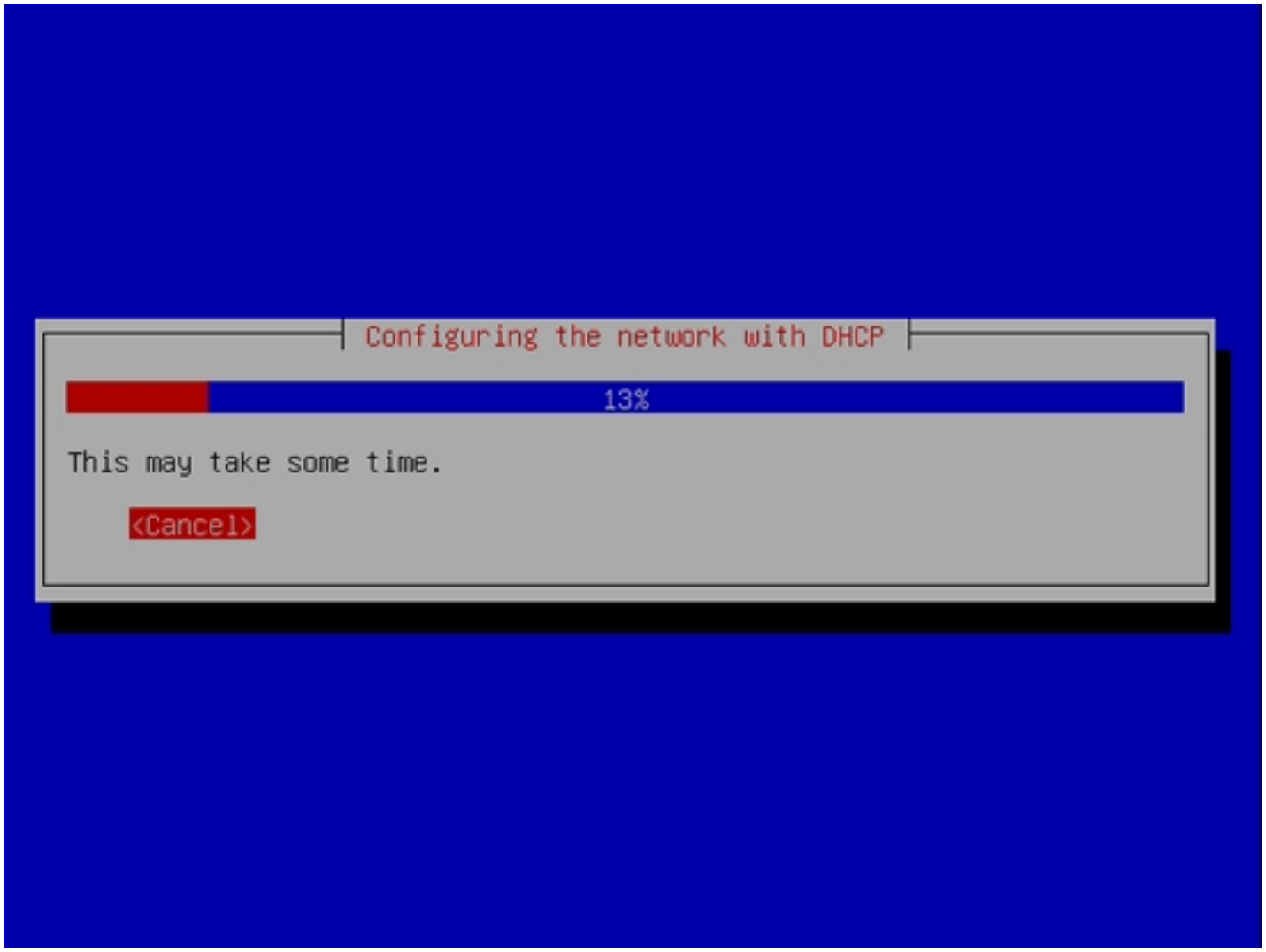




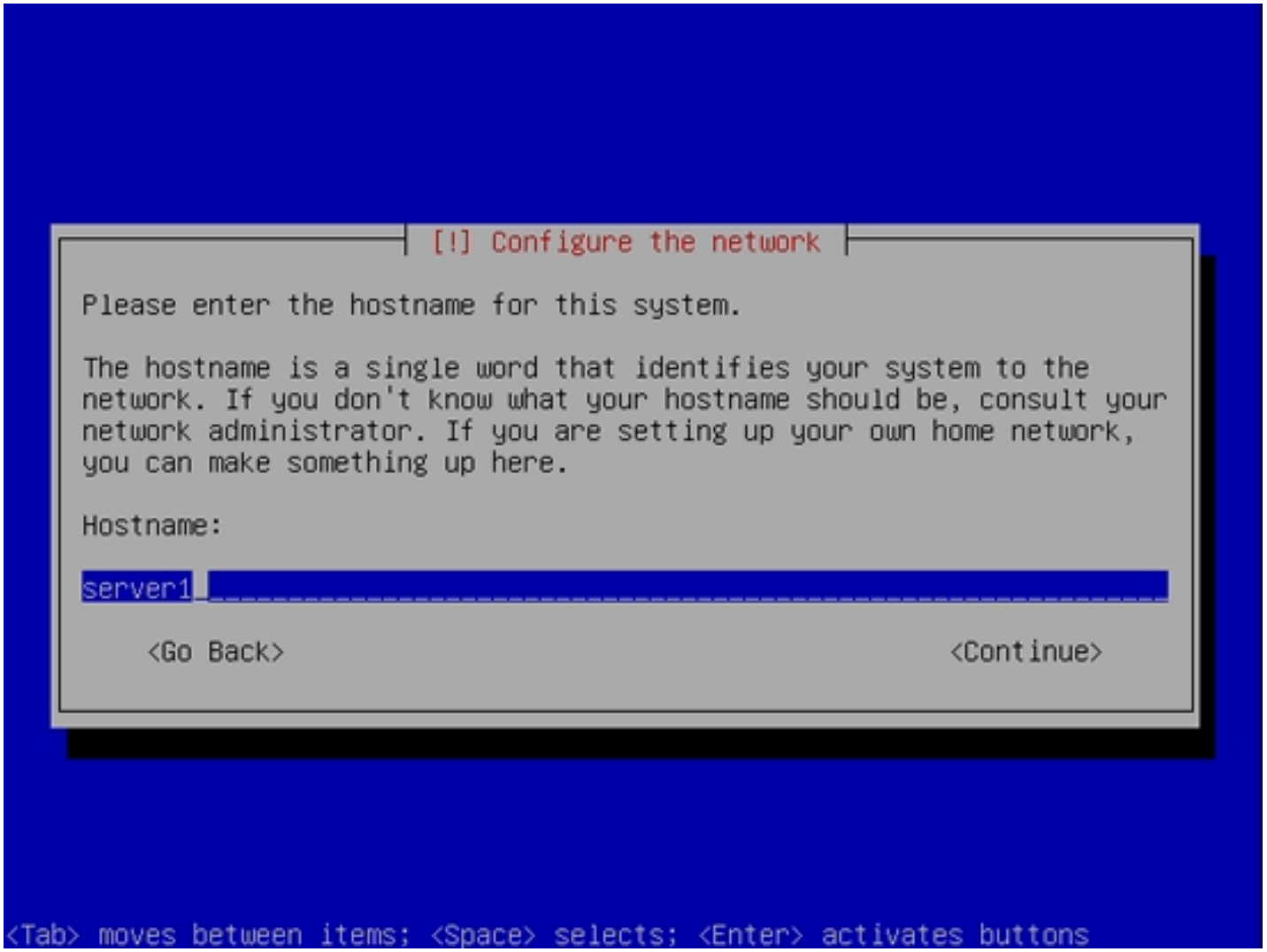

The installer checks the installation CD, your hardware, andconfigures the network with DHCP if there is a DHCP server in thenetwork:



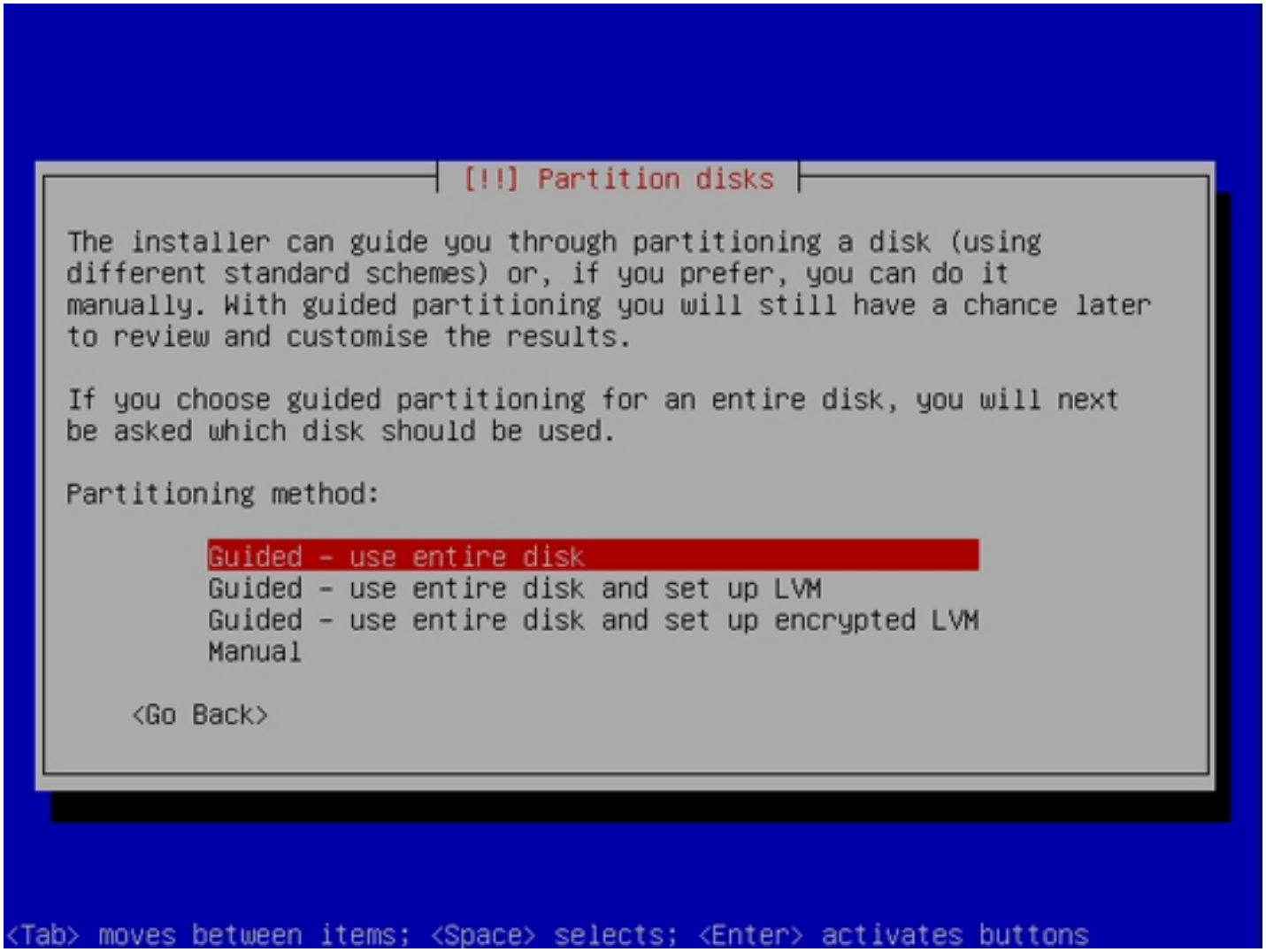




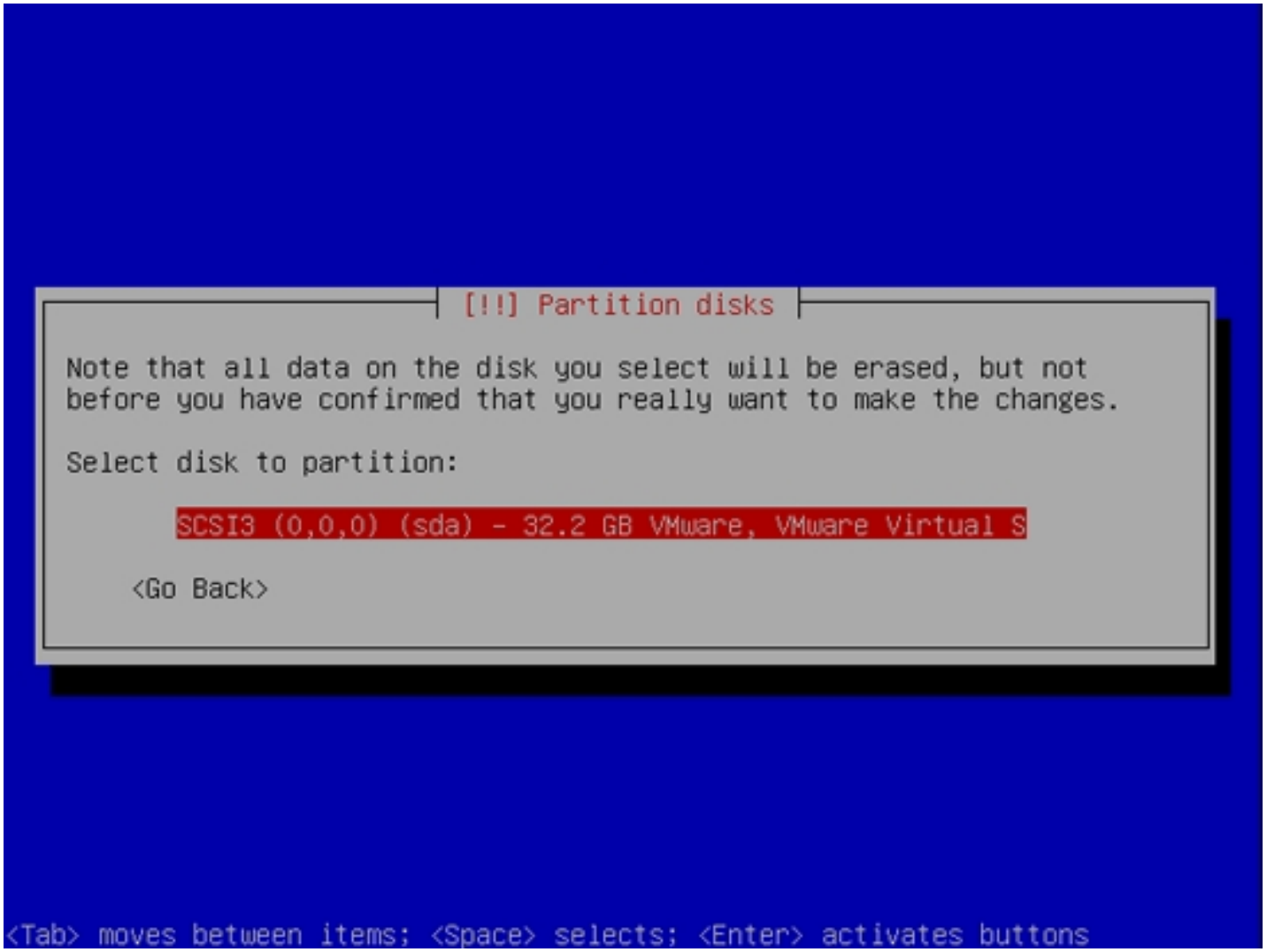

Enter the hostname. In this example, my system is called ***server1.tm.local,*** so I enter ***server1***:



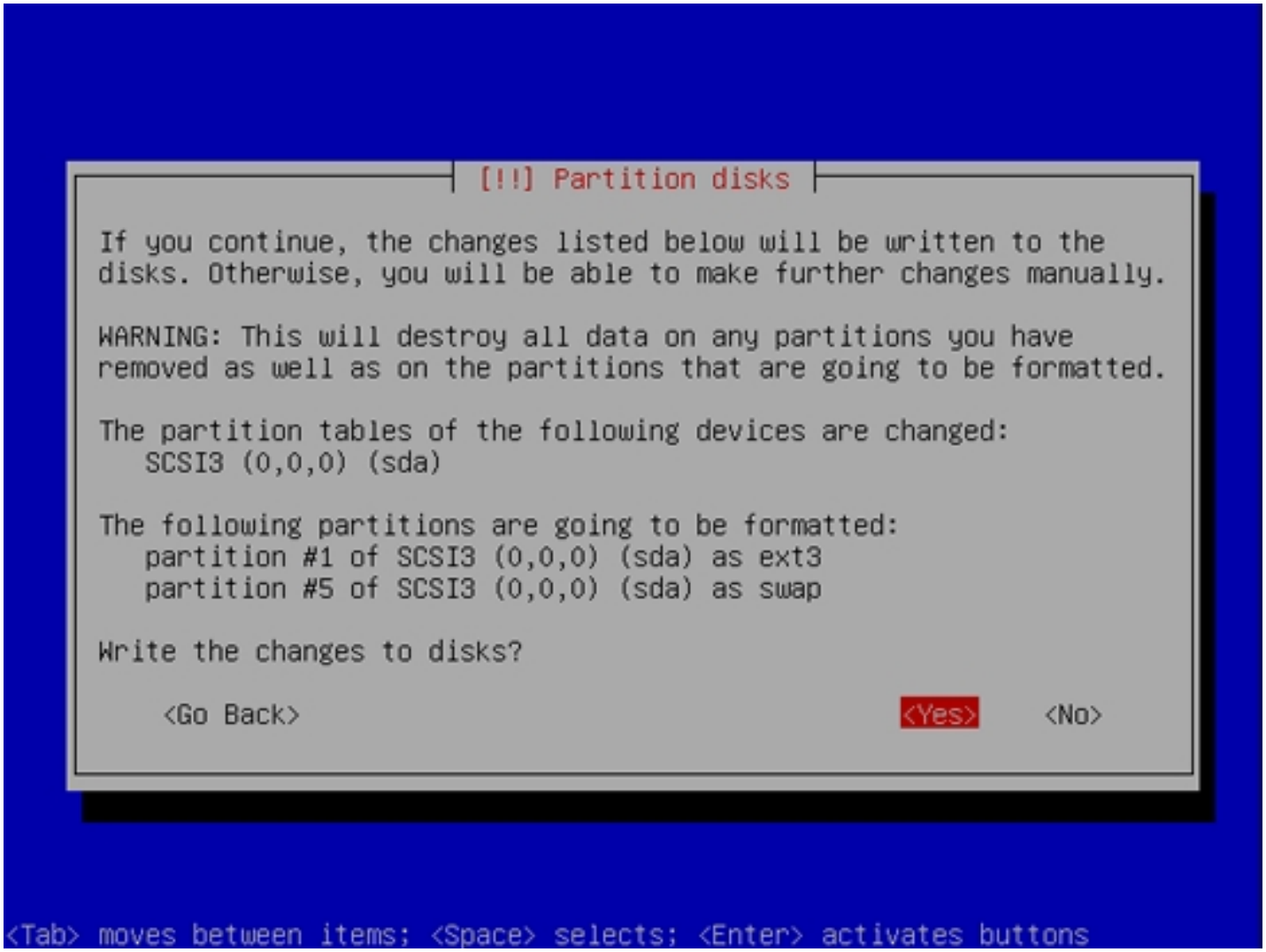

Now you have to partition your hard disk. For simplicity'ssake I will create one big partition (with the mount point */*) and a little swap partitionso I select *Guided - use entire disk*(of course, the partitioning is totally up to you - if you like, youcan create more than just one big partition, and you can also use LVM):




Select the disk that you want to partition:

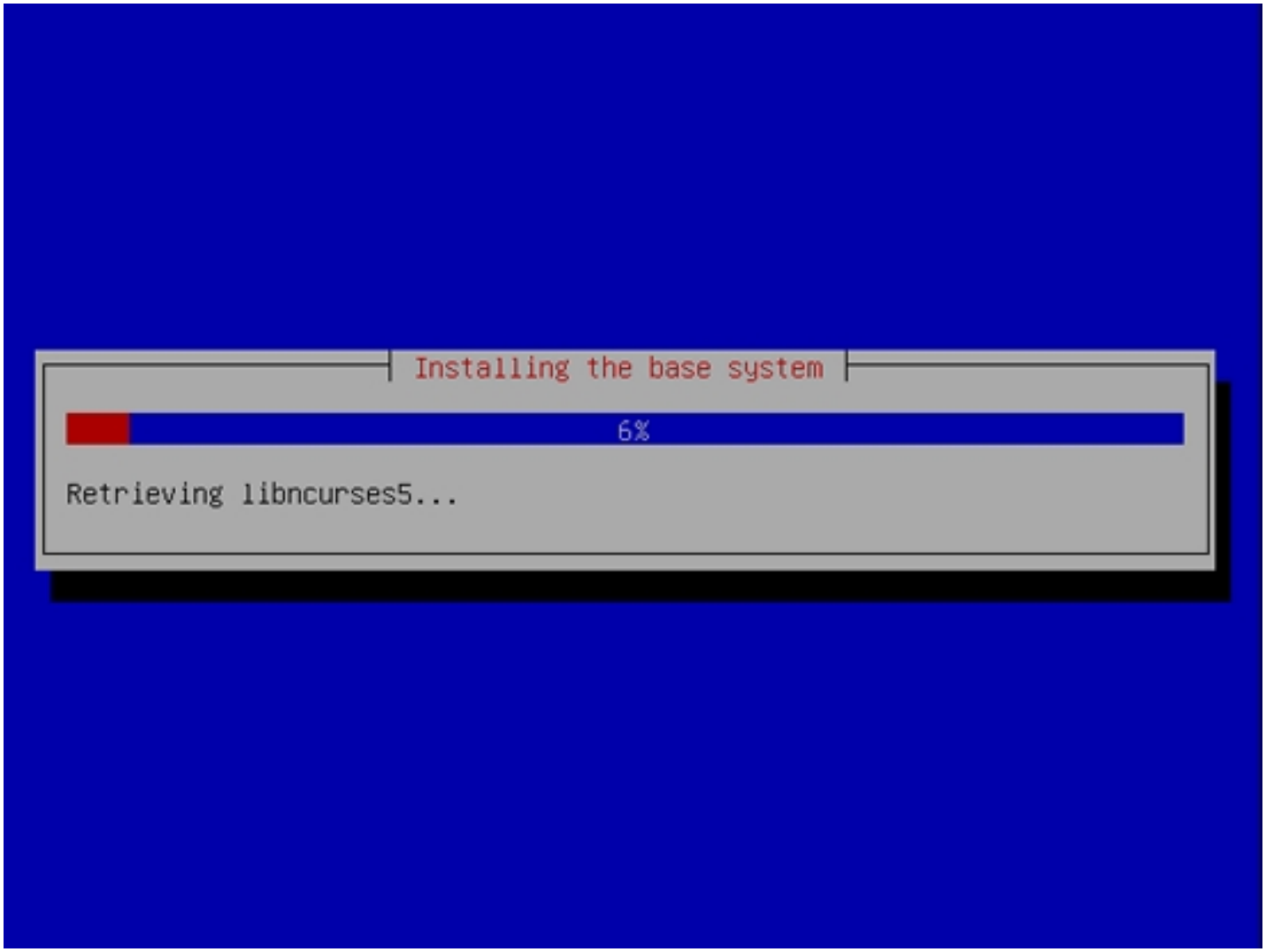


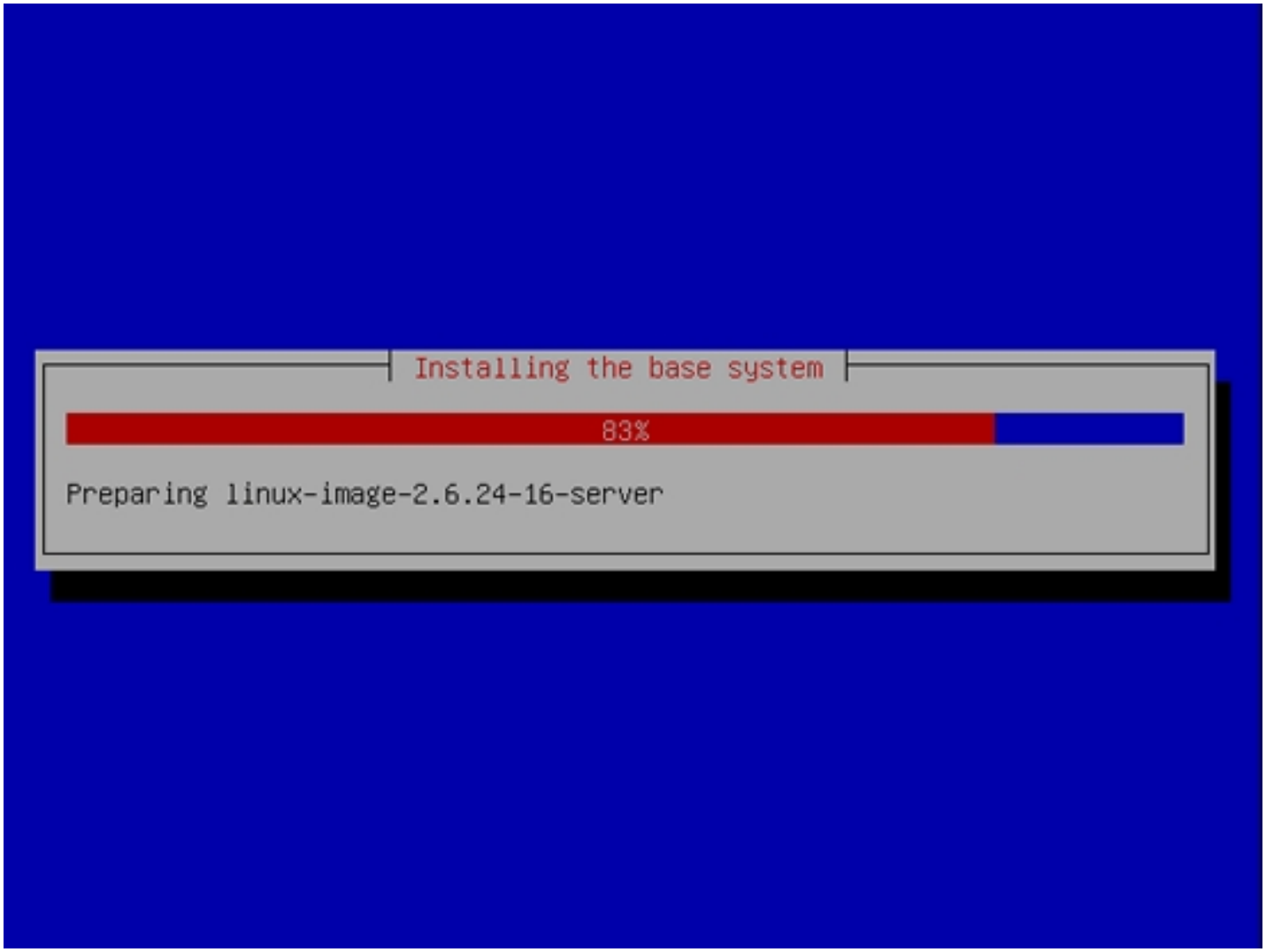

When you're finished, hit *Yes*when you're asked *Write the changesto disks?*:



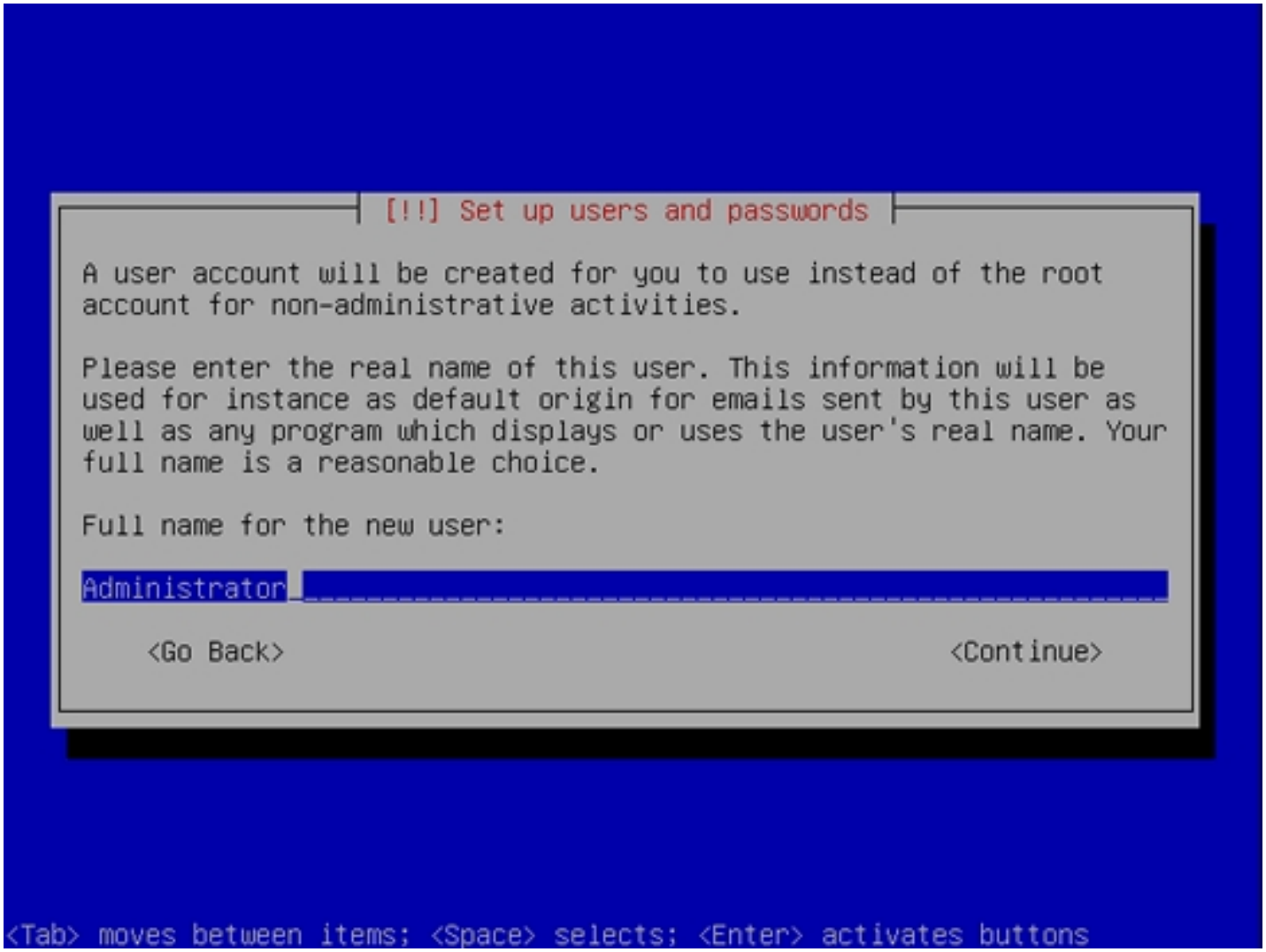

Afterwards, your new partitions are being created andformatted.

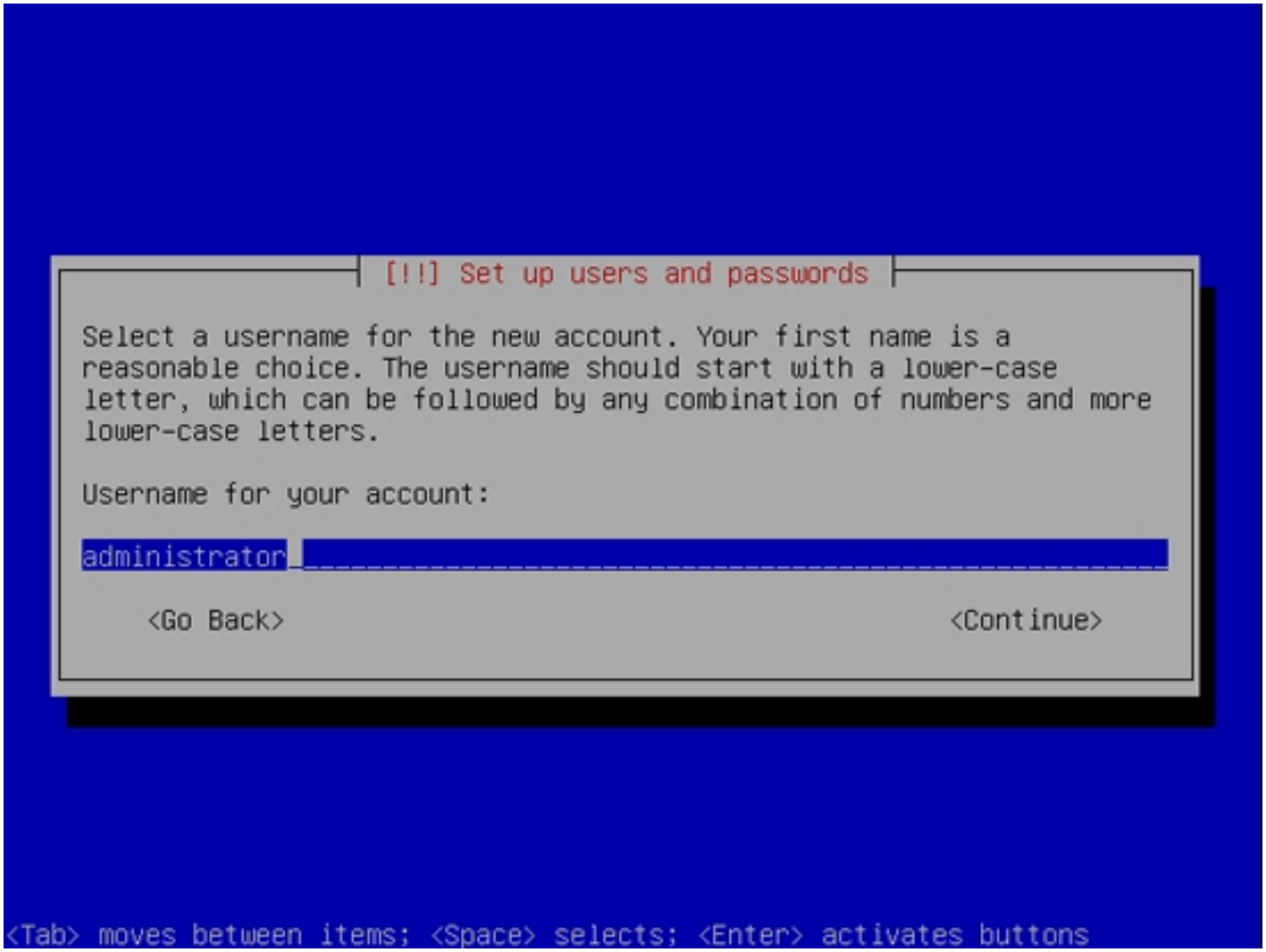
Now the base system is being installed:

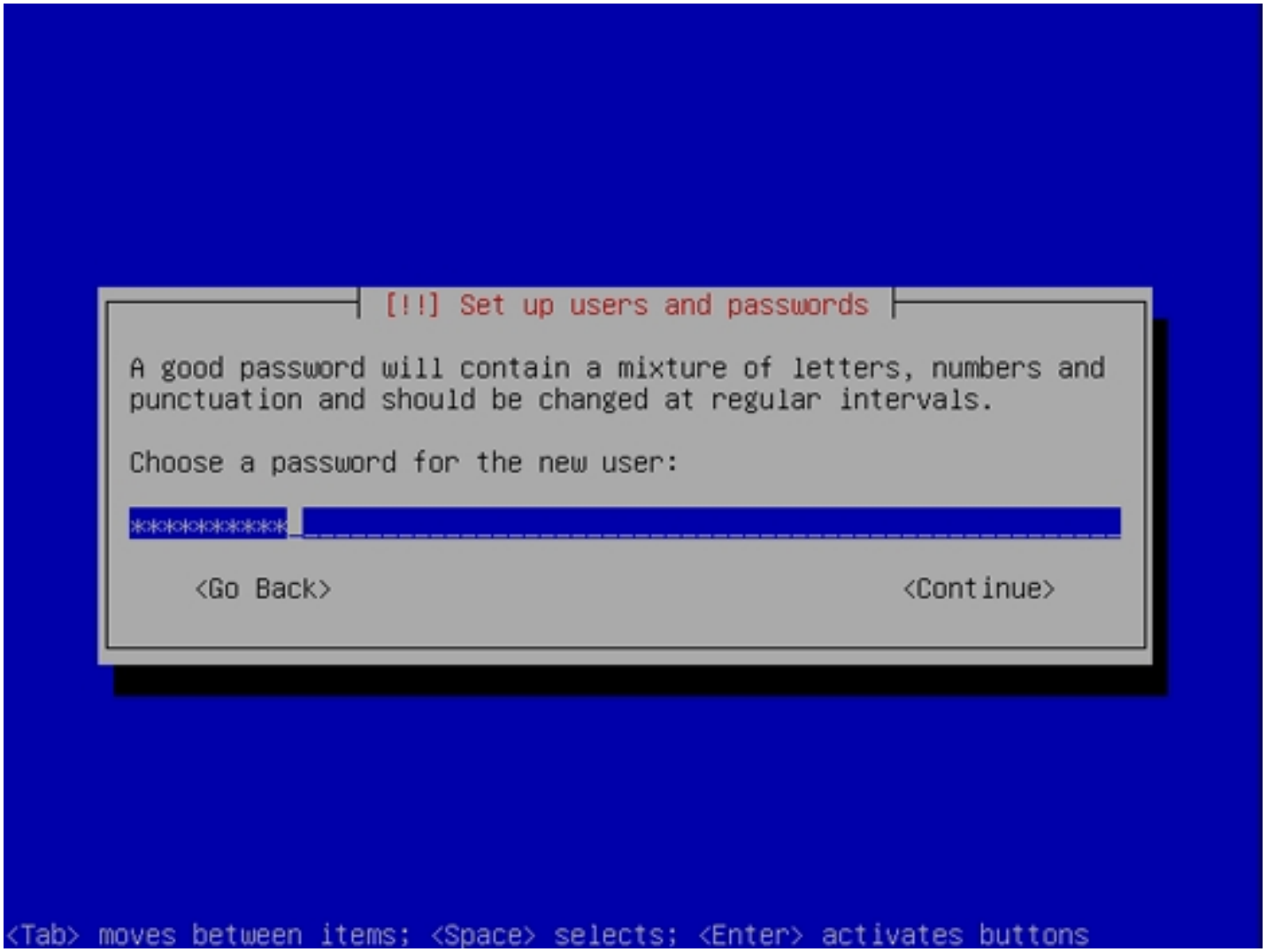


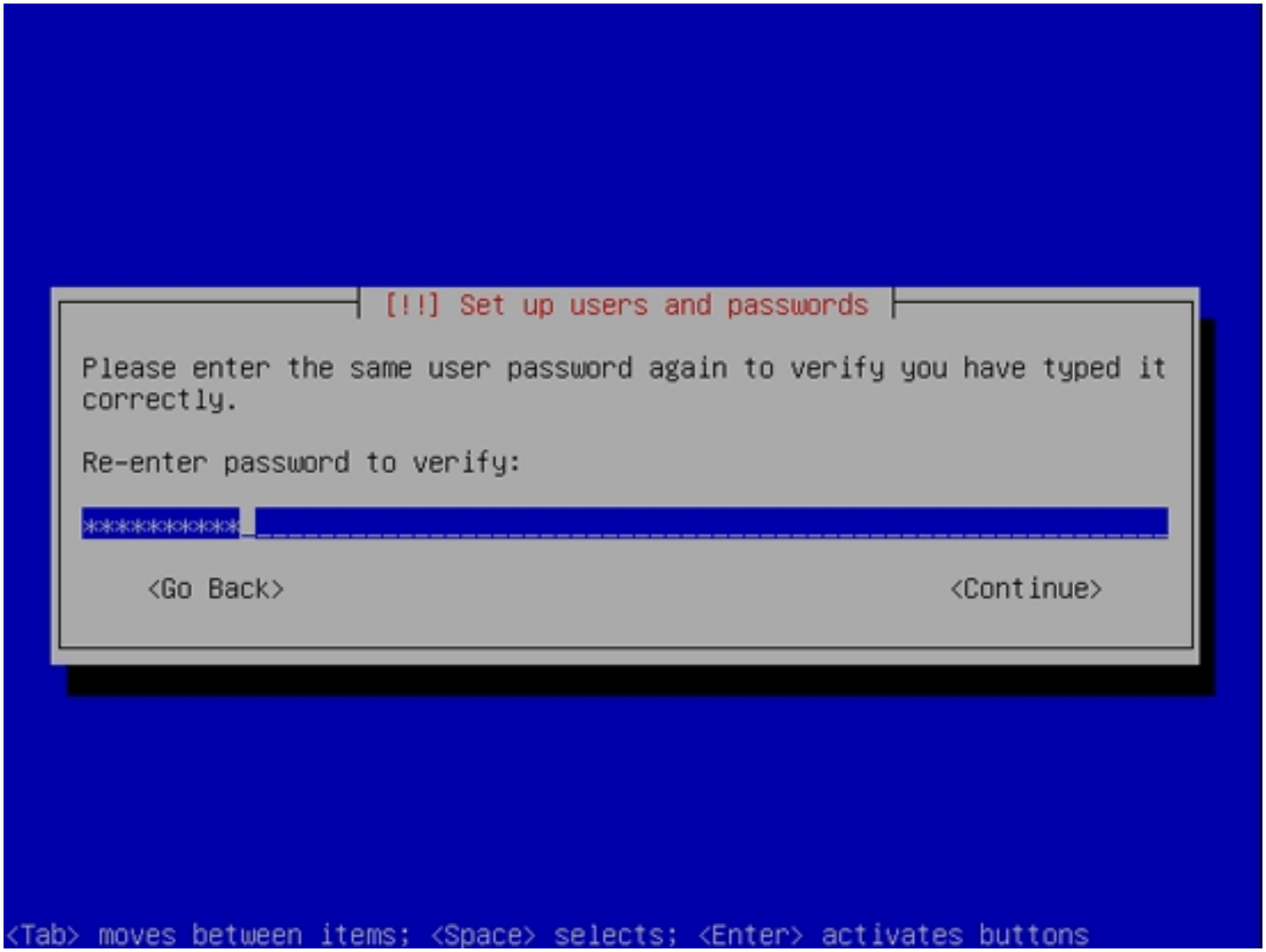



Create a user, for example the user *Administrator* with the user name *administrator* (don't use the user name *admin* as it is a reserved name on Ubuntu 8.04):

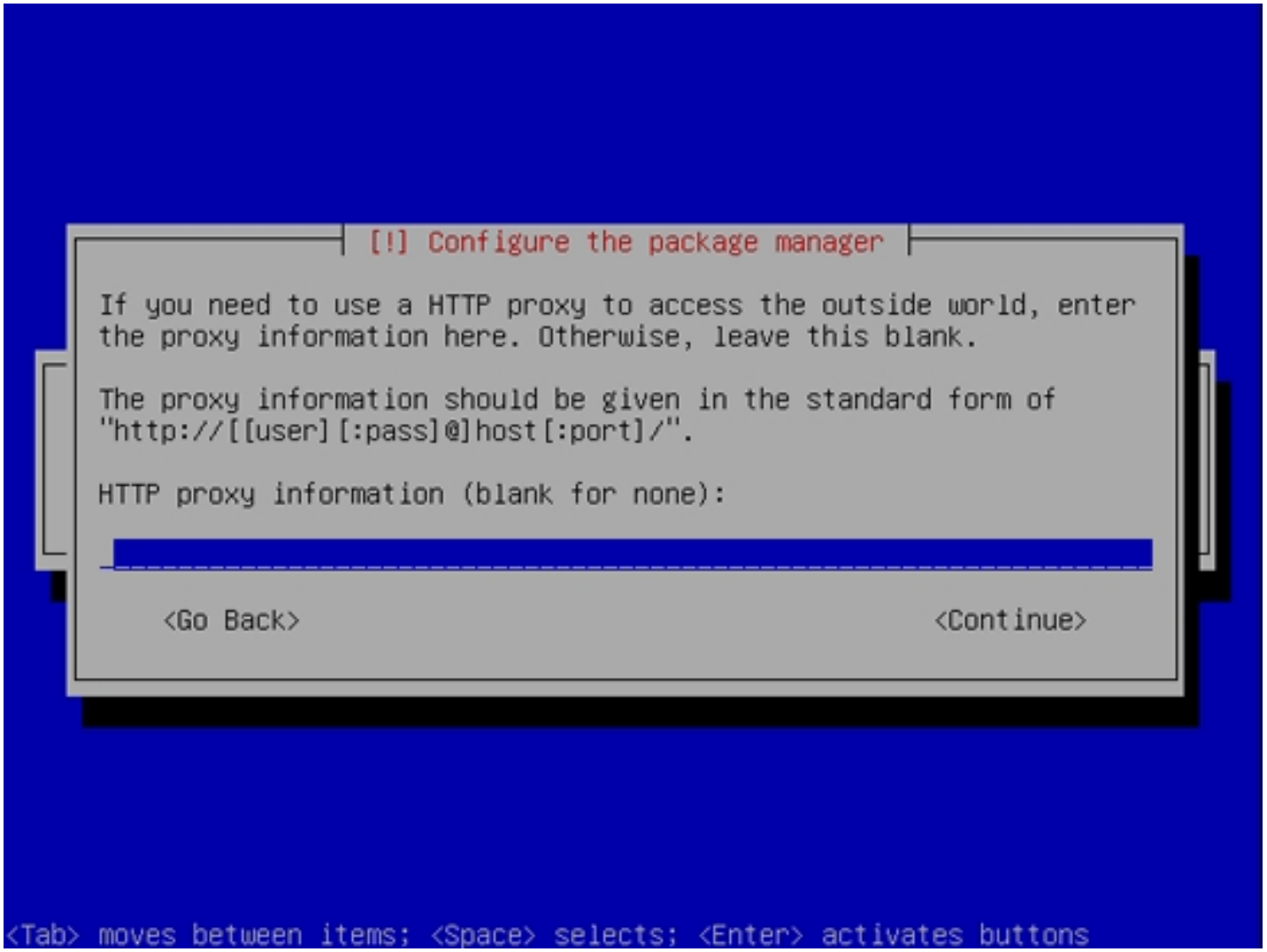


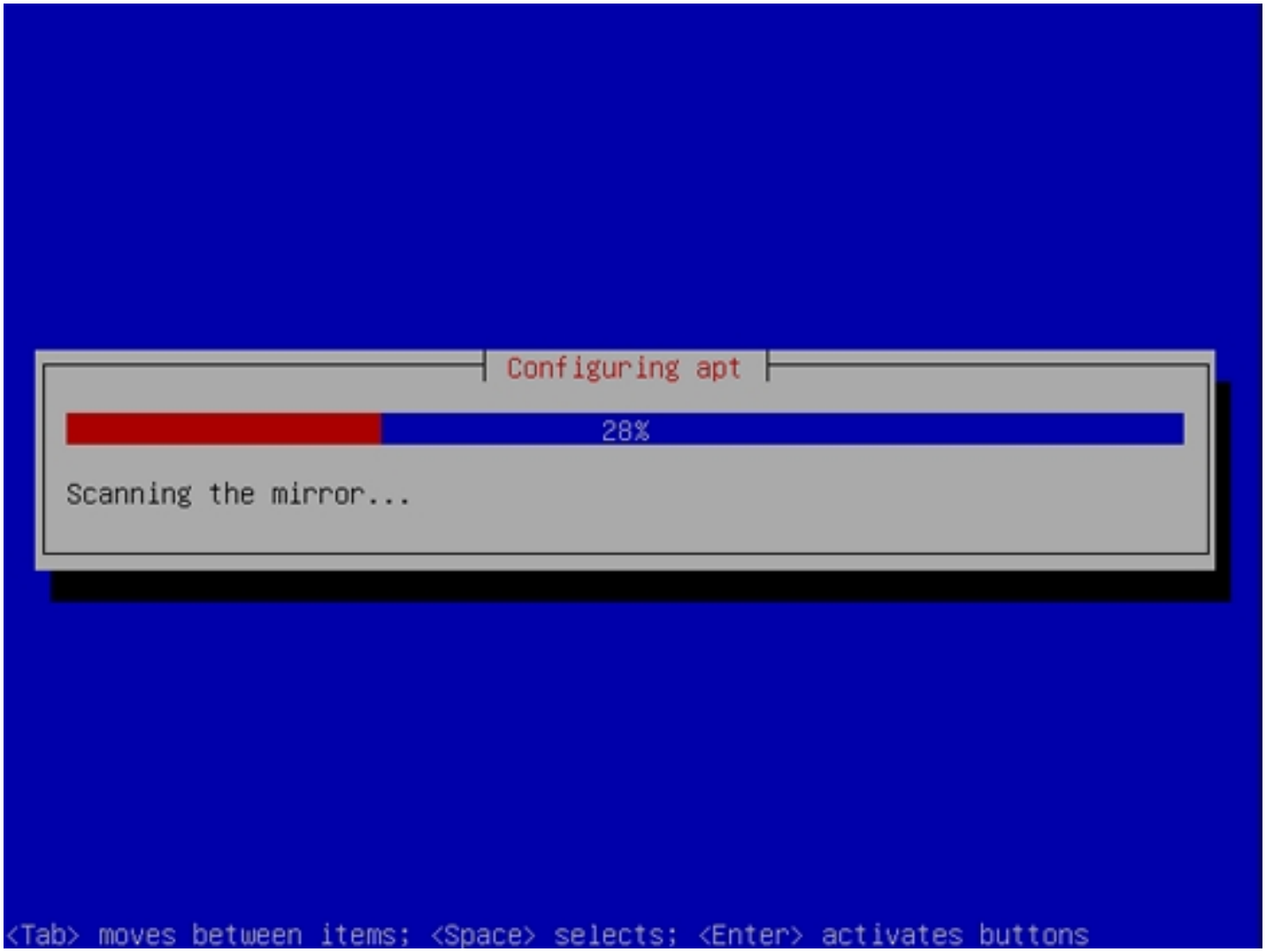




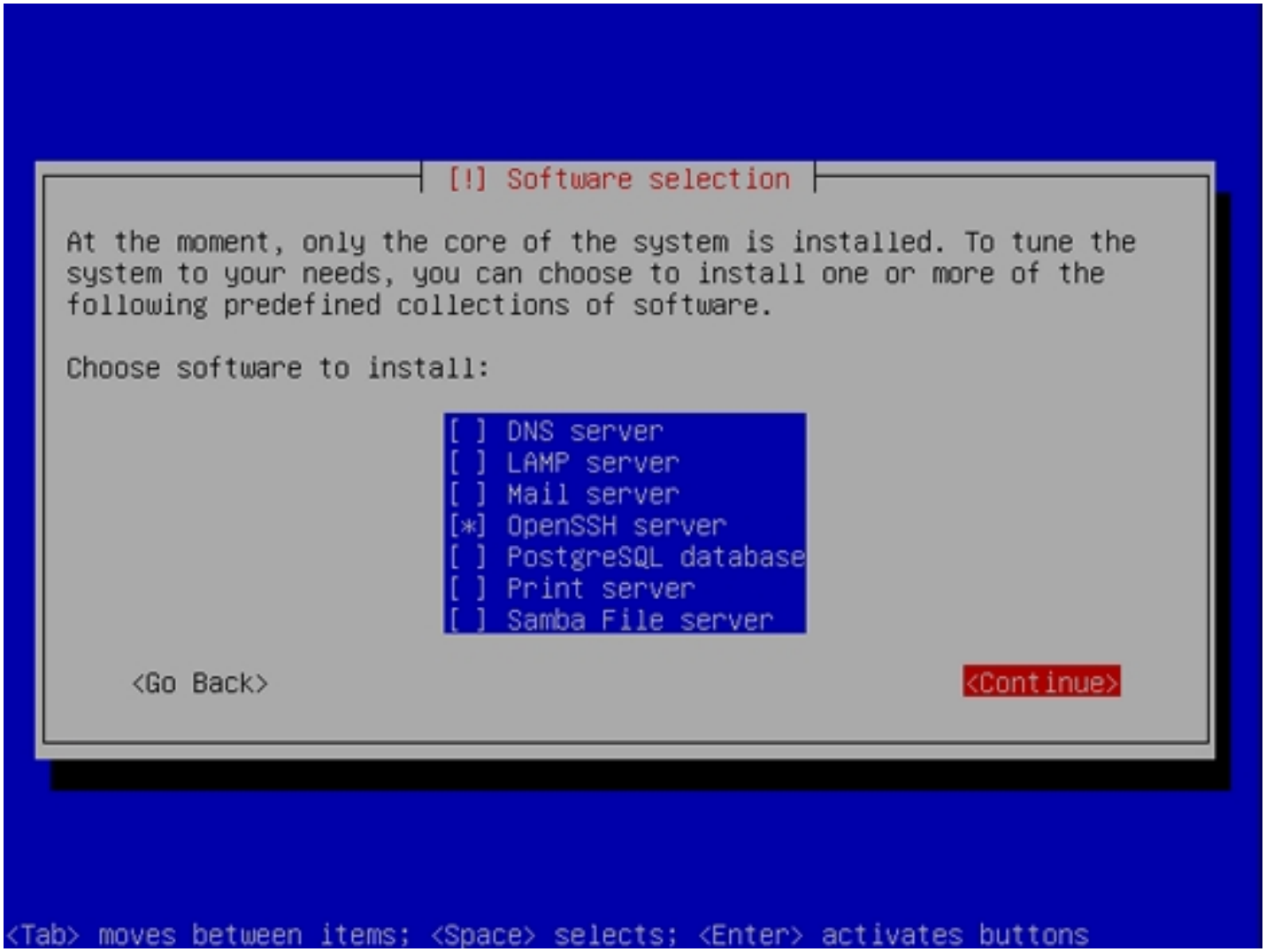



Next the package manager apt gets configured. Leave the HTTPproxyline empty unless you're using a proxy server to connect to theInternet:

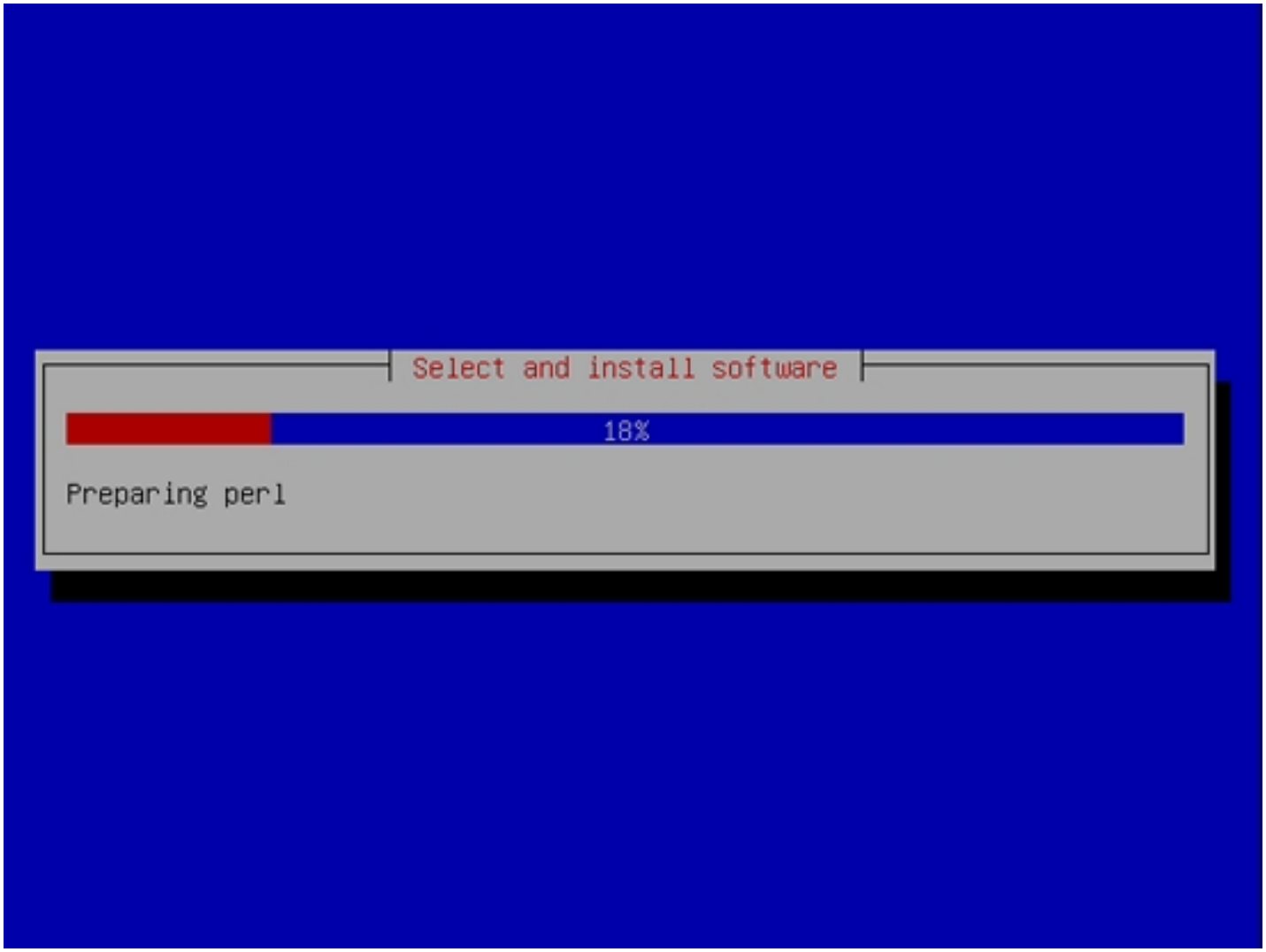




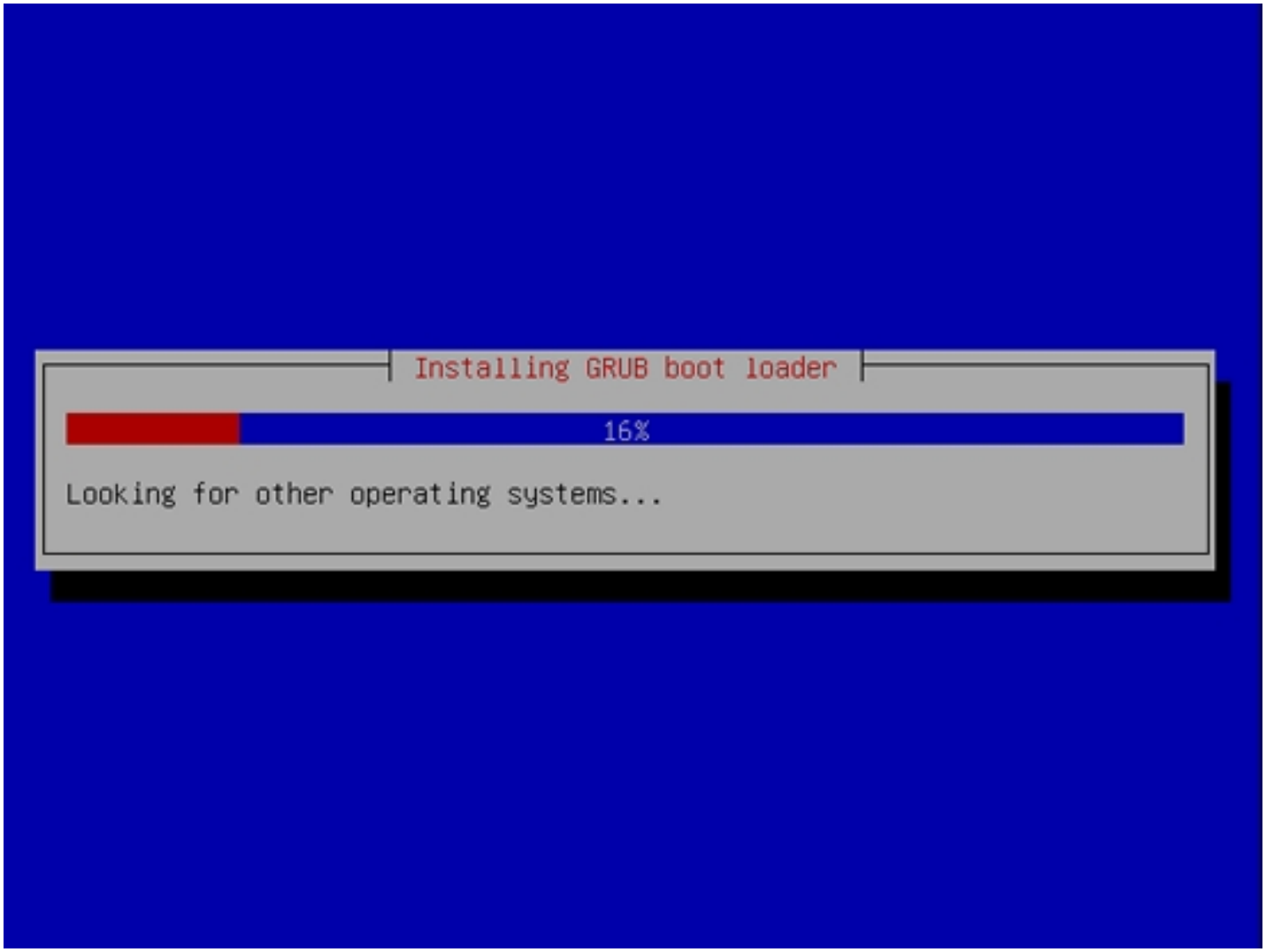

We need a DNS server, but nevertheless I don'tselect any of them now because I like to have full control over whatgets installed on my system. We will install the needed packagesmanually later on. The only item I select here is `OpenSSH server` so that I canimmediately connect to the system with an SSH client such as ***PuTTY*** after the installation hasfinished:



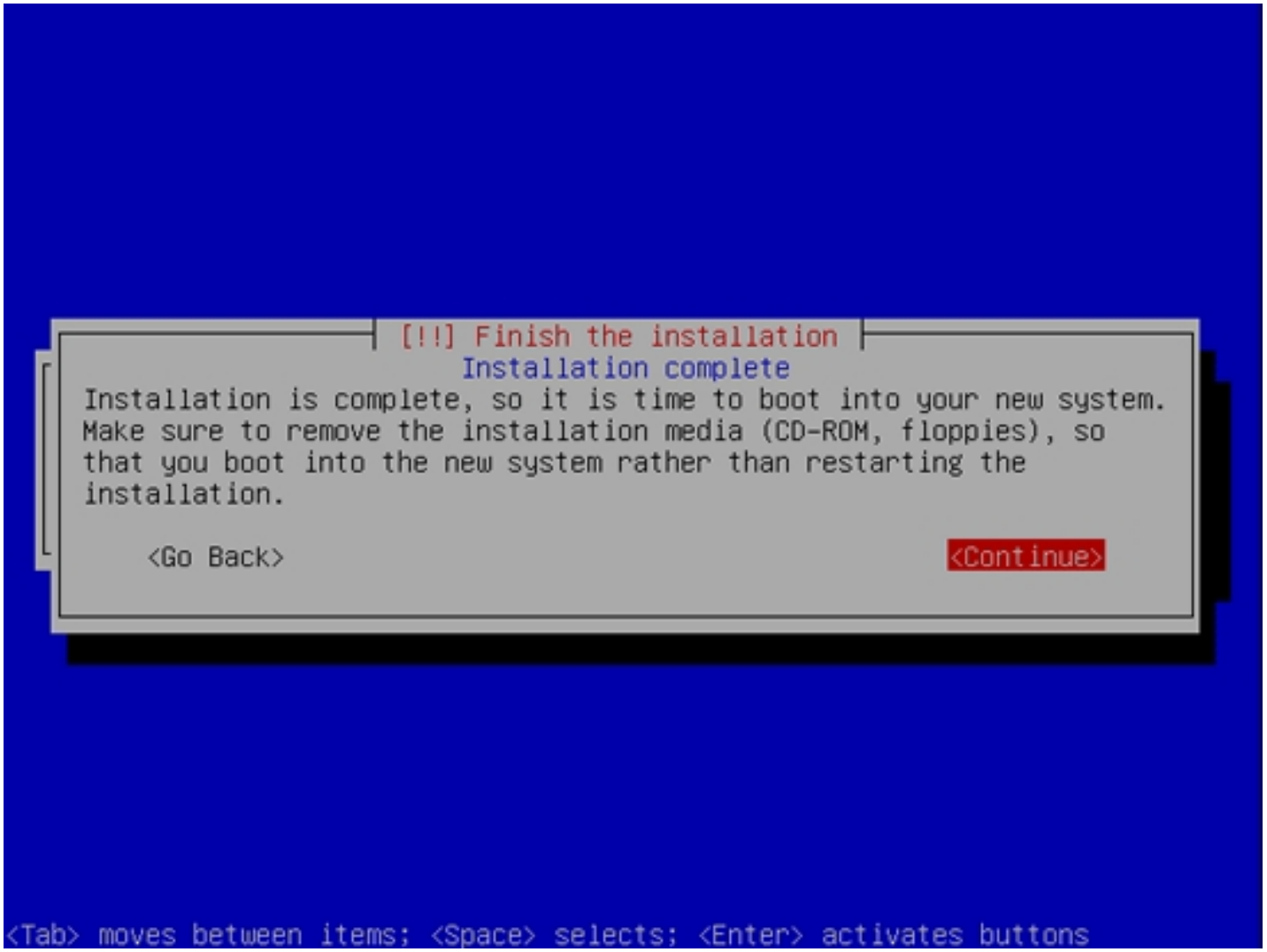

The installation continues:




The GRUB boot loader gets installed:




The base system installation is now finished. Remove theinstallation CD from the CD drive and hit *Continue*to reboot the system:




On to the next step...

## *4 Enable The root Account*

After the reboot you can login with your previously createdusername (e.g. *administrator*).Because we have to run all the steps from this tutorial as root user, so we will enable the root account.

Run

```
sudo passwd root
```

and give root a password. Afterwards we can switch root byrunning

```
su
```

## 5 Install The SSH Server (Optional)

If you did not install the OpenSSH server during the systeminstallation, you can do it now:

```
apt-get install ssh openssh-server
```

From now on you can use an SSH client such as ***PuTTY*** and connect from yourworkstation to your Ubuntu 8.04 LTS server and follow the remainingsteps from this tutorial.

## 6 Install vim-full (Optional)

I'll use *vi*as my text editor in this tutorial. The default *vi*program has some strange behavior on Ubuntu; to fix this,we install *vim-full*:

```
apt-get install vim-full
```

(You don't have to do this if you use a different texteditor such as joe or nano.)

## 7 Configure The Network

Because the Ubuntu installer has configured our system toget itsnetwork settings via DHCP, we have to change that now because a servershould have a static IP address. Edit ***/etc/network/interfaces*** and adjust it to your needs (in this example setup I willuse the IP address 192.168.0.100):

```
vi /etc/network/interfaces
```

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
 address 192.168.0.100
 netmask 255.255.255.0
 network 192.168.0.0
 broadcast 192.168.0.255
 gateway 192.168.0.1
```

Please make sure your network configurationare set correctly, feel free to change that based on your networkconfiguration.

Then restart your network:

```
/etc/init.d/networking restart
```

Then edit `/etc/hosts`.Make it look like this:

```
vi /etc/hosts
```

```
127.0.0.1     localhost.localdomain localhost
192.168.0.100 server1.tm.local      server1

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Now run

```
echo server1.tm.local >
/etc/hostname

/etc/init.d/hostname.sh start
```

Afterwards, run

```
hostname

hostname -f
```

Both should show `server1.tm.local` now.

## 8 Edit /etc/apt/sources.list And Update Your LinuxInstallation

Edit `/etc/apt/sources.list`.Comment out or remove the installation CD from the file and make surethat the `universe` and `multiverse` repositories are enabled. It should look like this:

```
vi /etc/apt/sources.list
```

```
#
# deb cdrom:[Ubuntu-Server 8.04 _Hardy Heron_ - Release i386 (20080423.2)]/ hardy main restricted
#deb cdrom:[Ubuntu-Server 8.04 _Hardy Heron_ - Release i386 (20080423.2)]/ hardy main restricted
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://de.archive.ubuntu.com/ubuntu/ hardy main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy main restricted
## Major bug fix updates produced after the final release of the
## distribution.
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu security
## team.
deb http://de.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy universe
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
```

```
## security team.
deb http://de.archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy multiverse
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
# deb http://de.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
# deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
## Uncomment the following two lines to add software from Canonical's
## 'partner' repository. This software is not part of Ubuntu, but is
## offered by Canonical and the respective vendors as a service to Ubuntu
## users.
# deb http://archive.canonical.com/ubuntu hardy partner
# deb-src http://archive.canonical.com/ubuntu hardy partner
deb http://security.ubuntu.com/ubuntu hardy-security main restricted
deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://security.ubuntu.com/ubuntu hardy-security universe
deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://security.ubuntu.com/ubuntu hardy-security multiverse
deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
```

## Then run

```
apt-get update
```

to update the apt package database and

```
apt-get upgrade
```

to install the latest updates (if there are any).

## 9 Disable AppArmor (This is a must for things to go well here)

AppArmor is a security extension (similar to SELinux) that shouldprovide extended security, which usually causes more problems than advantages. Therefore Idisable it.

We can disable it like this:

```
/etc/init.d/apparmor stop

update-rc.d -f apparmor remove
```

## 10 Install the DNS Server

Run

```
apt-get install bind9
```

For security reasons we want to run BIND chrooted so we have to do the following steps:

```
/etc/init.d/bind9 stop
```

Edit the file */etc/default/bind9* so that the daemon will run as the unprivileged user *bind*, chrooted to */var/lib/named*. Modify the line: *OPTIONS="-u bind"* so that it reads *OPTIONS="-u bind -t /var/lib/named"*:

```
vi /etc/default/bind9
```

```
OPTIONS="-u bind -t /var/lib/named"
# Set RESOLVCONF=no to not run resolvconf
RESOLVCONF=yes
```

Create the necessary directories under */var/lib*:

```
mkdir -p /var/lib/named/etc
mkdir /var/lib/named/dev
mkdir -p /var/lib/named/var/cache/bind
mkdir -p /var/lib/named/var/run/bind/run
```

Then move the config directory from */etc* to */var/lib/named/etc*:

```
mv /etc/bind /var/lib/named/etc
```

Create a symlink to the new config directory from the old location (to avoid problems when bind gets updated in the future):

```
ln -s /var/lib/named/etc/bind /etc/bind
```

Make null and random devices, and fix permissions of the directories:

```
mknod /var/lib/named/dev/null c 1 3
mknod /var/lib/named/dev/random c 1 8
chmod 666 /var/lib/named/dev/null /var/lib/named/dev/random
chown -R bind:bind /var/lib/named/var/*
```

```
chown -R bind:bind /var/lib/named/etc/bind
```

We need to modify */etc/default/syslogd* so that we can still get important messages logged to the system logs. Modify the line: *SYSLOGD=""* so that it reads: *SYSLOGD="-a /var/lib/named/dev/log"*:

```
vi /etc/default/syslogd
```

```
#
# Top configuration file for syslogd
#

#
# Full documentation of possible arguments are found in the manpage
# syslogd(8).
#

#
# For remote UDP logging use SYSLOGD="-r"
#
SYSLOGD="-a /var/lib/named/dev/log"
```

Restart the logging daemon:

```
/etc/init.d/sysklogd restart
```

Start up BIND, and check */var/log/syslog* for errors:

```
/etc/init.d/bind9 start
```

## 11 Configure BIND

Now the main configuration file in BIND is named.conf, however named.conf.local is already included in named.conf and its there for customized configuration, so we will edit named.conf.local and we will add our zones, here I added a zone camed tm.local as well as a reverse zone for 192.168.0.0:

```
vi /etc/bind/named.conf.local
```

```
zone "tm.local" {
        type master;
        file "/etc/bind/zones/tm.local.db";
        };



zone "3.13.10.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/rev.0.168.192.in-addr.arpa";
};
```

Please note that if you want to add a comment in *named.conf* or *named.conf.local* use //, also you can see above the zone file for *tm.local* is called *tm.local.db* and is located in */etc/bind/zone*, the most important thing that the zone file uses ; as the prefix for a comment and not //, as I saw confusions in a lot of forums so I thought to add it here - (same for the reverse zone).

## 12 Configure the Zones

We will start with the zone tm.local

```
mkdir /etc/bind/zones
```

```
vi /etc/bind/zones/tm.local.db
```

```
$TTL 1500
@  IN SOA server1.tm.local. root (
                        2007062703      ;serial
                        28800           ;refresh
                        3600            ;retry
                        604800          ;expire
                        38400 )         ;minimum 25 minutes
tm.local.     IN     NS     server1.tm.local.
server1       IN     A      192.168.0.100
webserver1    IN     A      192.168.0.103
webserver2    IN     A      192.168.0.104
loadb1        IN     A      192.168.0.101
loadb2        IN     A      192.168.0.102
tm.local.     IN     MX     10    server1.tm.local.
```

Feel free to replace the above zone name (tm.local) or your dns server name (server1) as needed,  just note the . DOT after the zone name.

Now let's go ahead with the reverse zone.

```
vi /etc/bind/zones/rev.3.13.10.in-addr.arpa
```

```
$TTL 1500
@  IN SOA server1.tm.local. root (
                        2007062703      ;serial
                        28800           ;refresh
                        3600            ;retry
                        604800          ;expire
                        38400 )         ;minimum 25 minutes

               IN   NS     server1.tm.local.
```

```
100             IN  PTR  server1.tm.local.
103             IN  PTR  webserver1.tm.local.
104             IN  PTR  webserver2.tm.local.
101             IN  PTR  load1.tm.local.
102             IN  PTR  load2.tm.local.
```

Now configure the server to forward any requests to your ISP server so it case resolve external IPs.

```
vi /etc/bind/named.conf.options
```

Uncomment the forwarder section to look like this:

```
forwarders {
    # Replace the address below with the address of your ISP DNS server
    123.123.123.123;
};
```

## 13 Configure the server to use itself as DNS

```
vi /etc/resolv.conf
```

```
search tm.local
nameserver 192.168.0.100
```

## 14 References and Sources

- Falko's Article "***The Perfect Server - Ubuntu Hardy Heron (Ubuntu 8.04 LTS Server)***"
- BIND on Ubuntu "***Howto: Setup a DNS server with bind***"