

Creating a Debian 5.0 Lenny USB boot stick for the Vortex86DX (RoBoard RB-100)

Version 1.1 – By Jurrie Overgoor <jurrie.overgoor@2go-mobile.nl>

The latest version of this document is available at http://jurr.org/files/Debian_RB-100.pdf

Changelog

V1.0 First version of this document.

V1.1 Typo in cpio command (noticed by Matt Denton), updated Debian businesscard iso link.

Introduction

This document describes how to create a USB boot stick for the RoBoard RB-100. We will install Debian 5.0 Lenny, which is as of this time of writing the stable branch.

Why don't we use the standard USB boot stick method?

The problem with the RB-100 is that we can not use a Debian standard kernel. We have to use the kernel supplied by DMP which is specially compiled for the Vortex86DX chip.

Before we start

Formats in this document

Normal text is typed like this.

Output of a command is typed like this.

Commands are typed like this.

Also, we use Ubuntu as host operating system. Ubuntu works without a root user, but instead uses “sudo” for every command that needs to be run as root. If you use Debian, you'll have a root account, and should execute the sudo commands as root (leaving off the “sudo” part of the command of course).

What do we need?

Of course, you'll need the RoBoard RB-100. Other than that, you'll need an USB stick. The data on the stick will be erased, so backup your stick.

Working directory

For this document we work in the directory ~/RB-100. If you choose to work in an other directory, please adjust the commands.

USB stick device

We need to know what device is assigned to your USB stick. Plug in your USB stick, and give the command **dmesg**. My output is:

```

[ 3302.332012] usb 1-5: new high speed USB device using ehci_hcd and address 5
[ 3302.465179] usb 1-5: configuration #1 chosen from 1 choice
[ 3302.465600] scsi6 : SCSI emulation for USB Mass Storage devices
[ 3302.471236] usb-storage: device found at 5
[ 3302.471239] usb-storage: waiting for device to settle before scanning
[ 3307.468139] usb-storage: device scan complete
[ 3307.468616] scsi 6:0:0:0: Direct-Access      CBM          Flash Disk      5.00
PQ: 0 ANSI: 2
[ 3307.469851] sd 6:0:0:0: [sdb] 4139520 512-byte hardware sectors: (2.11
GB/1.97 GiB)
[ 3307.470366] sd 6:0:0:0: [sdb] Write Protect is off
[ 3307.470369] sd 6:0:0:0: [sdb] Mode Sense: 0b 00 00 08
[ 3307.470371] sd 6:0:0:0: [sdb] Assuming drive cache: write through
[ 3307.471719] sd 6:0:0:0: [sdb] 4139520 512-byte hardware sectors: (2.11
GB/1.97 GiB)
[ 3307.472314] sd 6:0:0:0: [sdb] Write Protect is off
[ 3307.472317] sd 6:0:0:0: [sdb] Mode Sense: 0b 00 00 08
[ 3307.472319] sd 6:0:0:0: [sdb] Assuming drive cache: write through
[ 3307.472323]  sdb:
[ 3307.474037] sd 6:0:0:0: [sdb] Attached SCSI removable disk
[ 3307.474124] sd 6:0:0:0: Attached scsi generic sg2 type 0

```

From the lines with “[sdb]” I can see my USB device is attached to `/dev/sdb`.

Required files

We require a bunch of files.

We need a Debian install cd iso. For this, we have several choices: `businesscard` (smallest) or `netinst` (little bigger). The `businesscard` iso is the smallest, but contains less software. You'll need to download that from your local mirror. You'll find both isos at <http://cdimage.debian.org/debian-cd/current/i386/iso-cd/>. Choose the iso you like; we'll use the `businesscard` iso.

We also need the `RoBoard` Debian package for the alternative kernel. It's available at http://www.roboard.com/download_ml.htm.

We need the initial ramdisk from the Debian project. It's available at <http://http.us.debian.org/debian/dists/stable/main/installer-i386/current/images/hd-media/>.

```
wget 'http://cdimage.debian.org/debian-cd/current/i386/iso-cd/debian-503-i386-businesscard.iso'
```

```
wget 'http://www.roboard.com/Files/RB-100/linux-image-2.6.29.1-vortex86dx_1.0_i386.zip'
```

```
wget 'http://http.us.debian.org/debian/dists/stable/main/installer-i386/current/images/hd-media/initrd.gz'
```

Required programs

We need `fdisk`, `syslinux`, `mkdosfs`. So we need the following packages:

```
sudo apt-get install mtools syslinux dosfstools
```

Preparing the USB stick

Format the stick to FAT16

We need a FAT16 filesystem on the stick. This is probably already the case, but we'll do it anyway to make sure:

```
umount /dev/sdb
```

```
sudo fdisk /dev/sdb
```

Delete all partitions using the **d** command.

Create a new partition by choosing **n, p** (primary partition), **1, enter, enter**.

Switch to FAT16 by choosing **t, e**.

Make the partition bootable by choosing **a, 1**.

Write the partition table and quit fdisk by choosing **w**.

Fdisk has written the new partition table, now we need to format the new partition:

```
sudo mkdosfs /dev/sdb1
```

Install syslinux

Ok, we now have a FAT16 usb stick. We need to make it bootable:

```
sudo syslinux /dev/sdb1
```

Please note that Syslinux can only process DOS (8.3) file names.

Creating the new initrd.gz

First, we extract the current initrd.gz package in a subdirectory:

```
mkdir initrd
```

```
cd initrd
```

```
gzip -vdc ../../initrd.gz | sudo cpio -i
```

```
cd ..
```

Now we extract the files from the .deb package in a subdirectory:

```
unzip ./linux-image-2.6.29.1-vortex86dx_1.0_i386.zip
```

```
mkdir ./deb
```

```
dpkg-deb -x ./linux-image-2.6.29.1-vortex86dx_1.0_i386.deb ./deb/
```

Now we copy the files from the .deb package to the original initrd:

```
sudo cp -rv ./deb/* ./initrd/
```

And we need to correct the module dependencies list:

```
sudo chroot ./initrd /bin/sh
```

```
depmod -a 2.6.29.1-vortex86dx
exit
```

Now we need to recreate the initrd.gz file:

```
cd initrd
find . | cpio -v -o -H newc | gzip > ../../initrd.gz
cd ..
```

Finishing the USB stick

Mount the USB stick:

```
mkdir ./usb
sudo mount /dev/sdb1 ./usb
sudo cp ./deb/boot/vmlinuz-2.6.29.1-vortex86dx ./usb/vmlinuz
sudo cp ./initrd.gz ./usb/
sudo cp ./debian-502-i386-businesscard.iso ./usb/debian.iso
sudo cp ./linux-image-2.6.29.1-vortex86dx_1.0_i386.deb ./usb/
sudo touch ./usb/syslinux.cfg
echo -e "default vmlinuz\nappend initrd=initrd.gz" > ./syslinux.cfg
sudo cp ./syslinux.cfg ./usb/syslinux.cfg
sudo umount ./usb
```

When the USB stick doesn't boot

My USB stick didn't boot at first (I was left with a black screen and a blinking cursor). I did the following:

```
sudo apt-get install mbr
install-mbr /dev/sdb
```

And the stick booted perfectly.

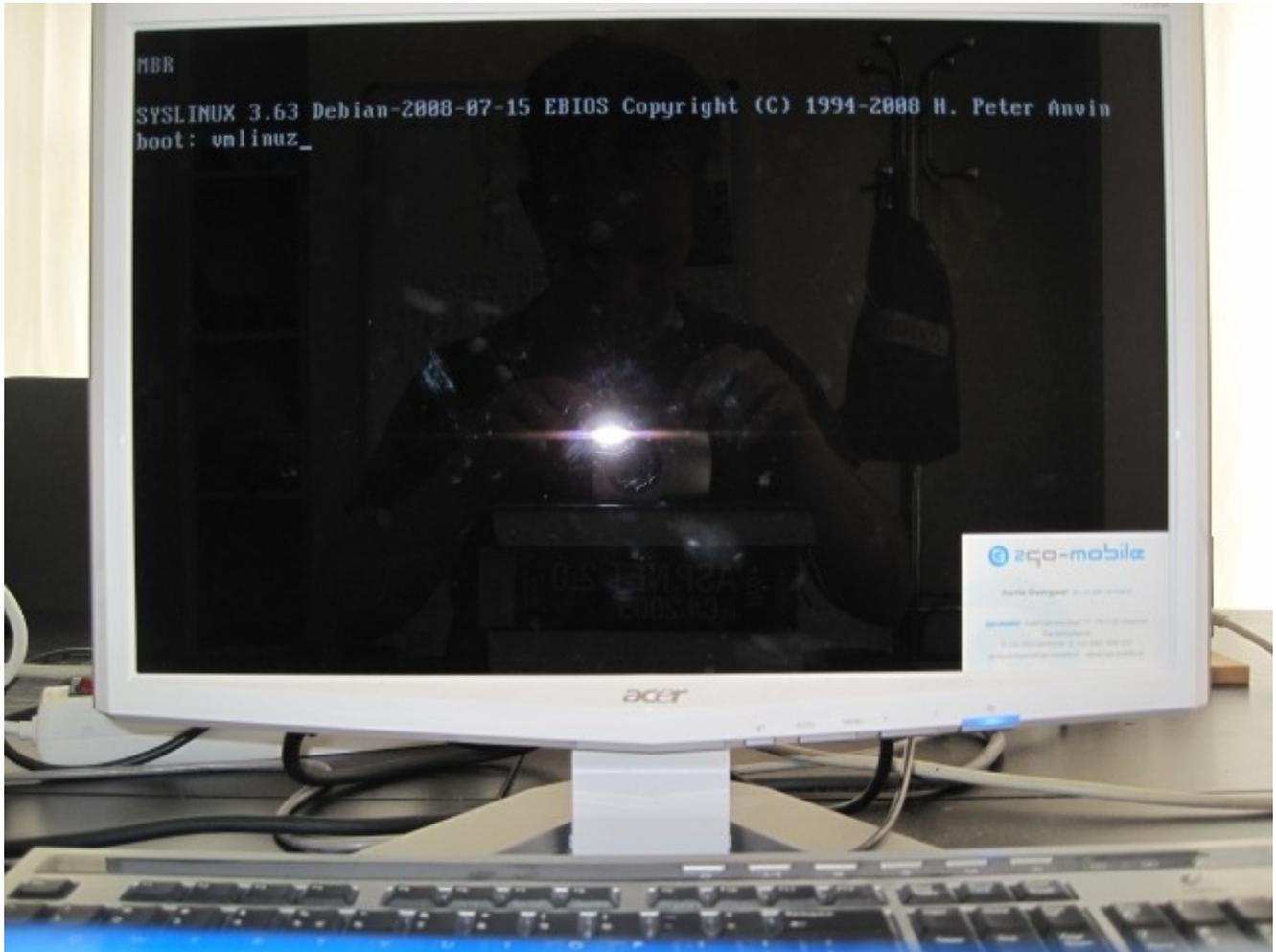
Installing Debian on your RB-100

Insert your newly created USB stick. Make sure you also have a SD card in the slot – this is where we are going to install Debian on.

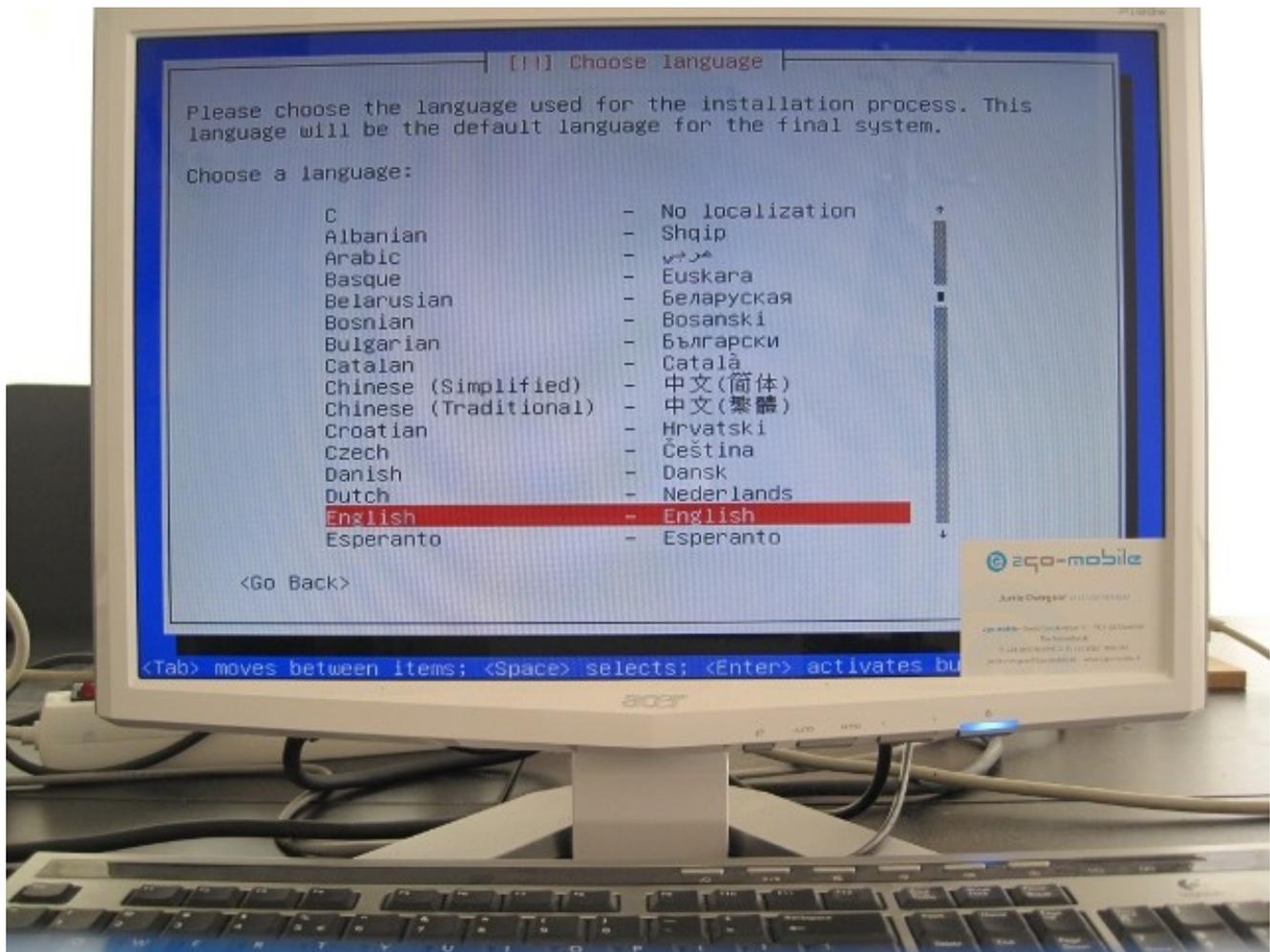
Press F11 when booting to get the boot device menu, choose the USB stick and press ENTER:



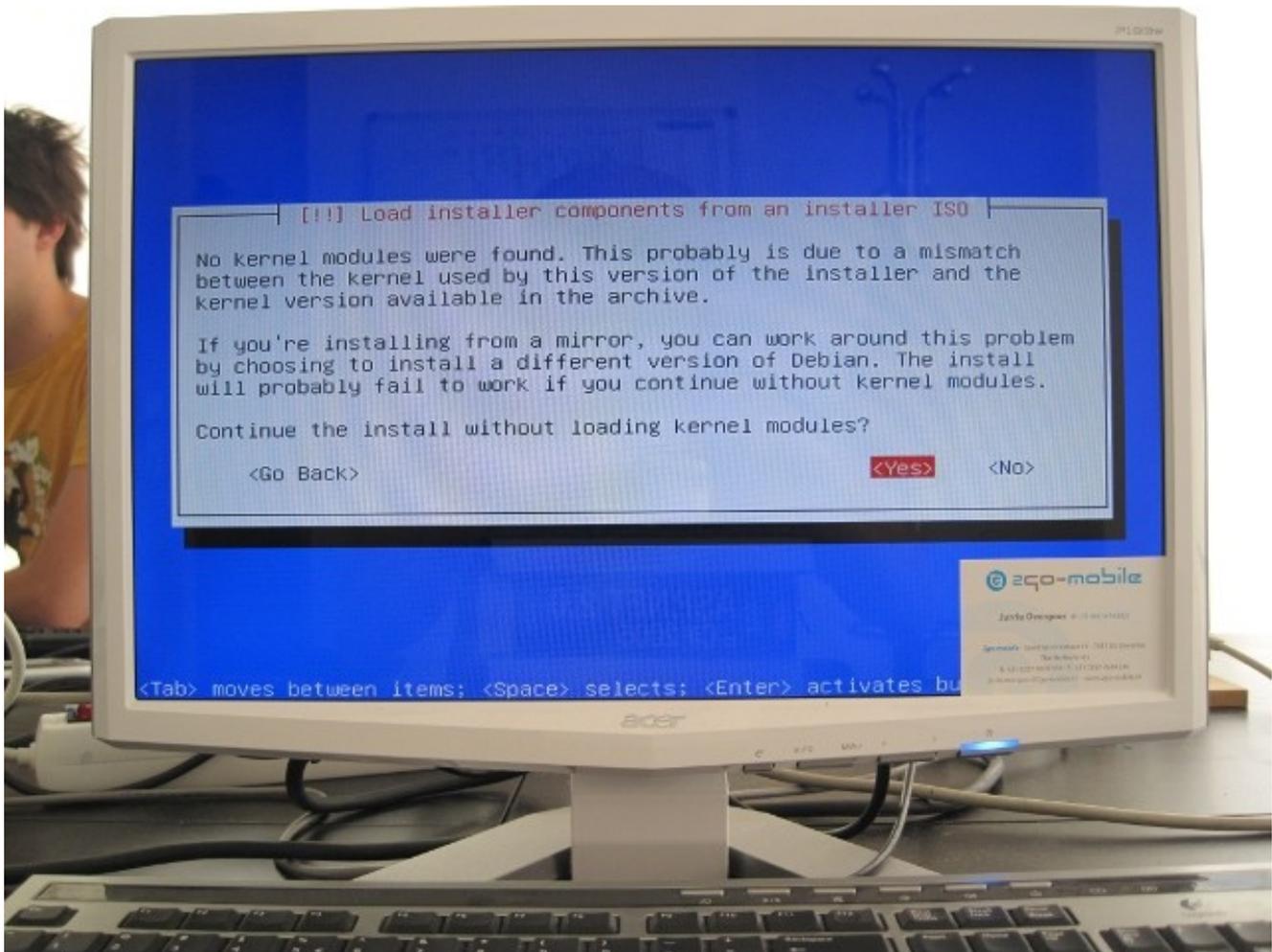
When you get the “boot:” prompt, enter vmlinuz.



A lot of loading is going on. Eventually the Debian installer should start, and should present you with the following screen:

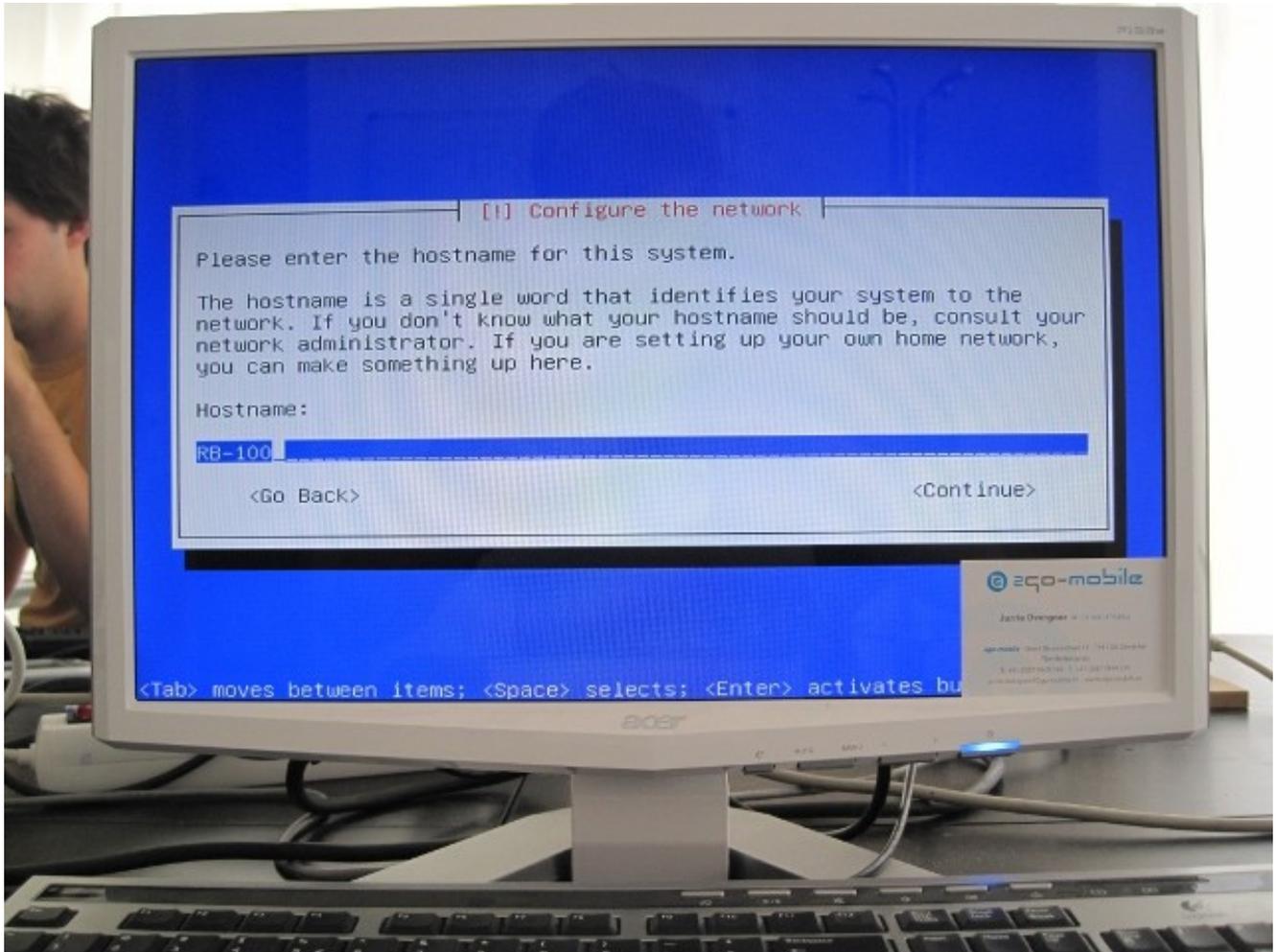


Step through the installation. You will get a warning that the installer can not find any kernel modules. You can ignore this warning (choose Yes). It is actually correct: we are running the special Vortex86DX kernel, but are installing the vanilla i386 kernel.

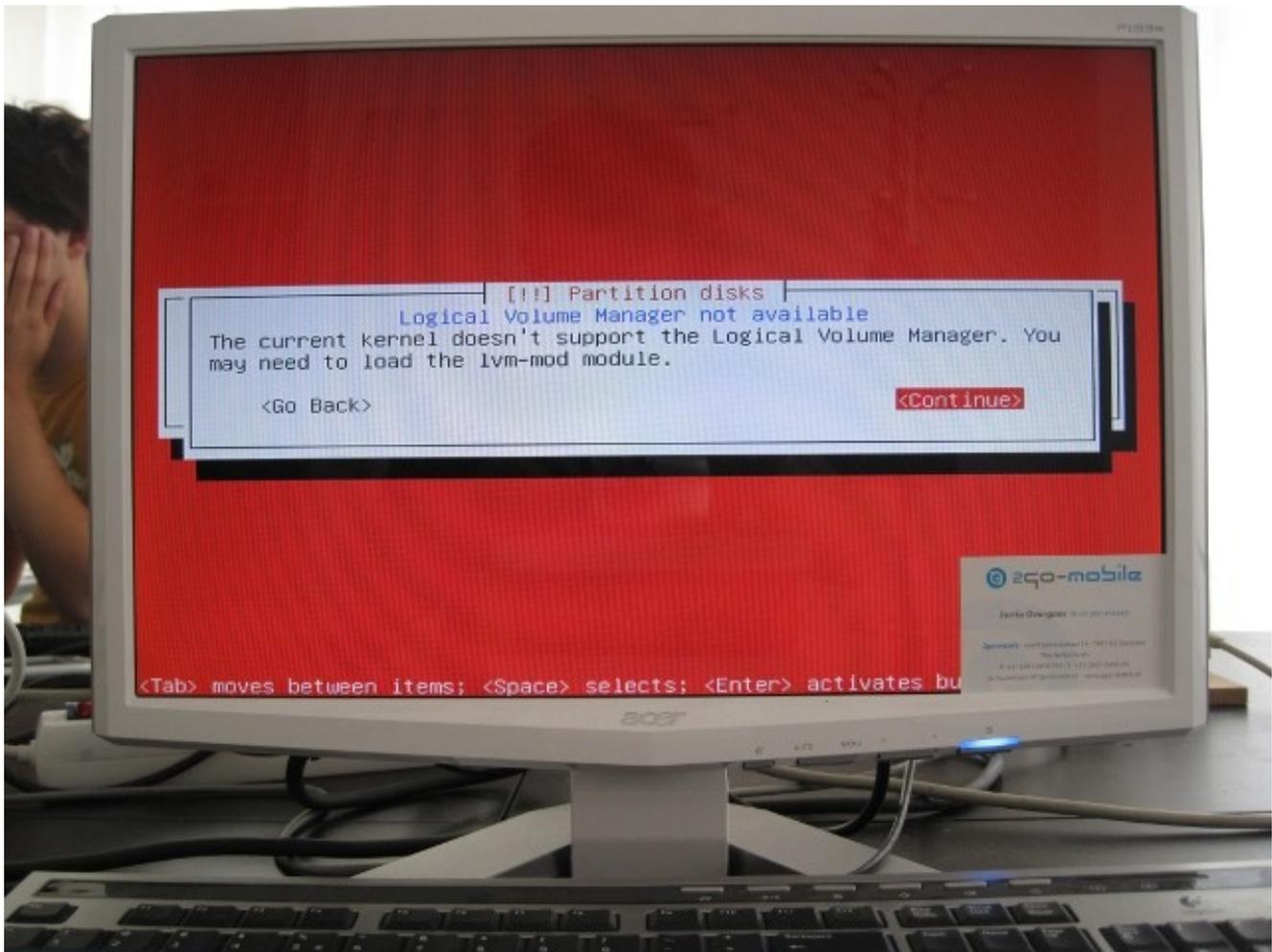


From now on, make sure you plug in your network cable. (Yes, this will work! Because we are running the correct kernel.) We need this to download additional packages and security updates. You can probably run through the installation without network, but I did not test this.

The installer will continue with DHCP setup and asks you for your hostname.

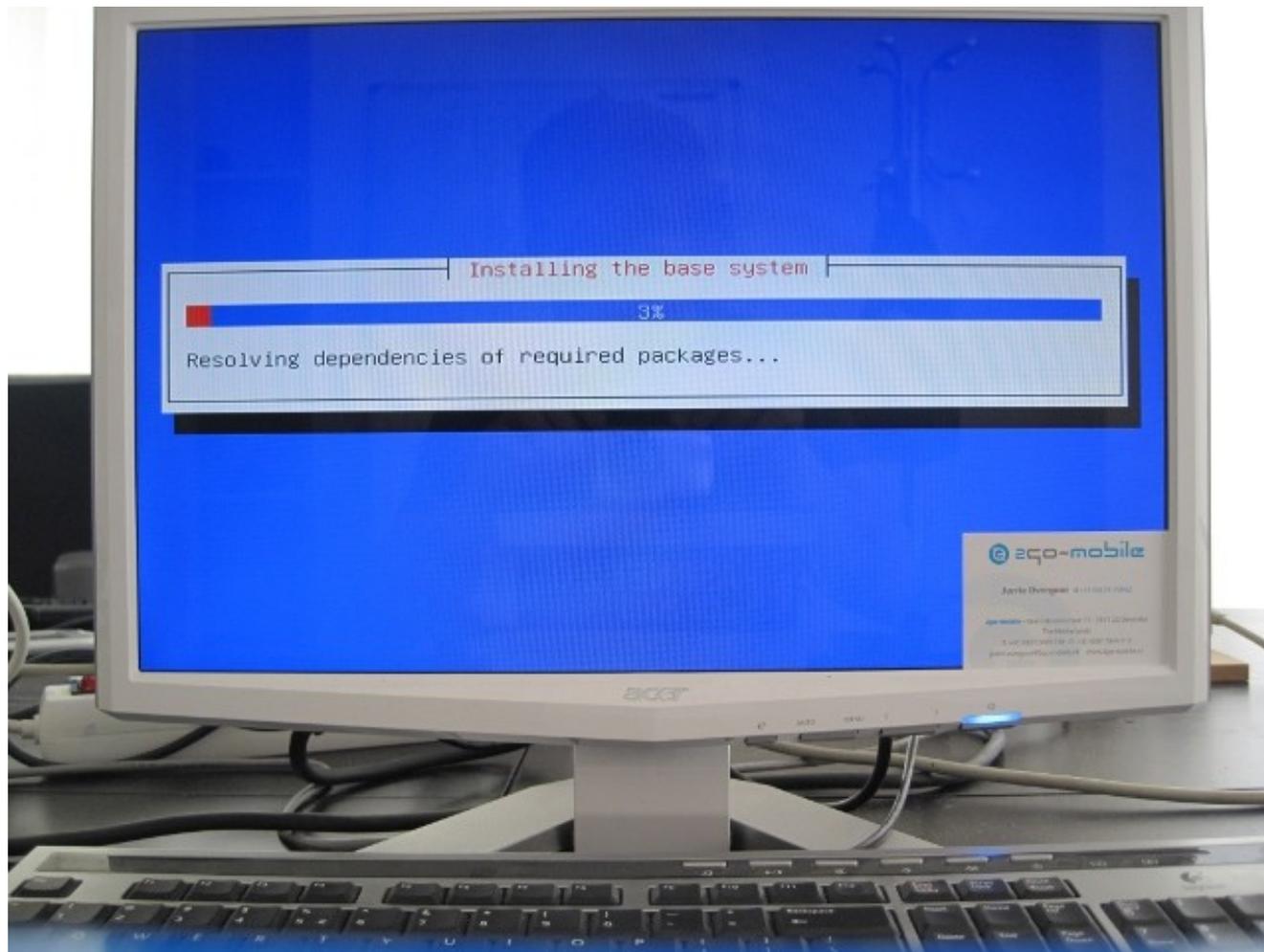


When you come up to the part where you partition your disk, you get the following error about no Logical Volume Manager support. I don't know what causes this (maybe just the fact that we are using the usinss card iso), but you can ignore it. I don't use LVM anyway...

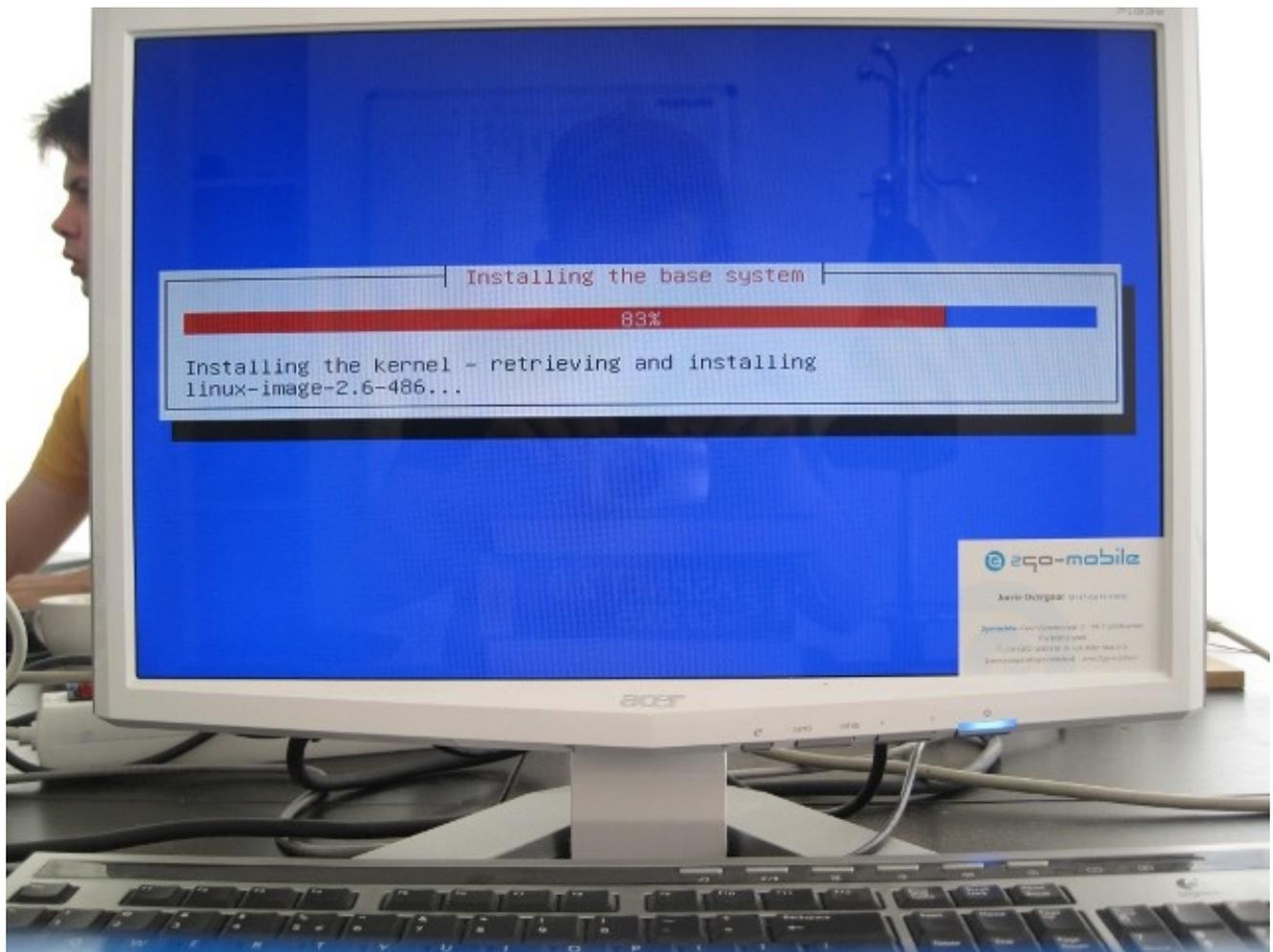


Partition your SD card the way you like it. I use a 2Gb SD card and made only 1 partition (with standard values) to fill up the whole. It's ext3, mount point /, bootable. I don't use a swap partition because SD cards can't handle that much read/write cycles, and I don't plan to run memory hungry applications anyway.

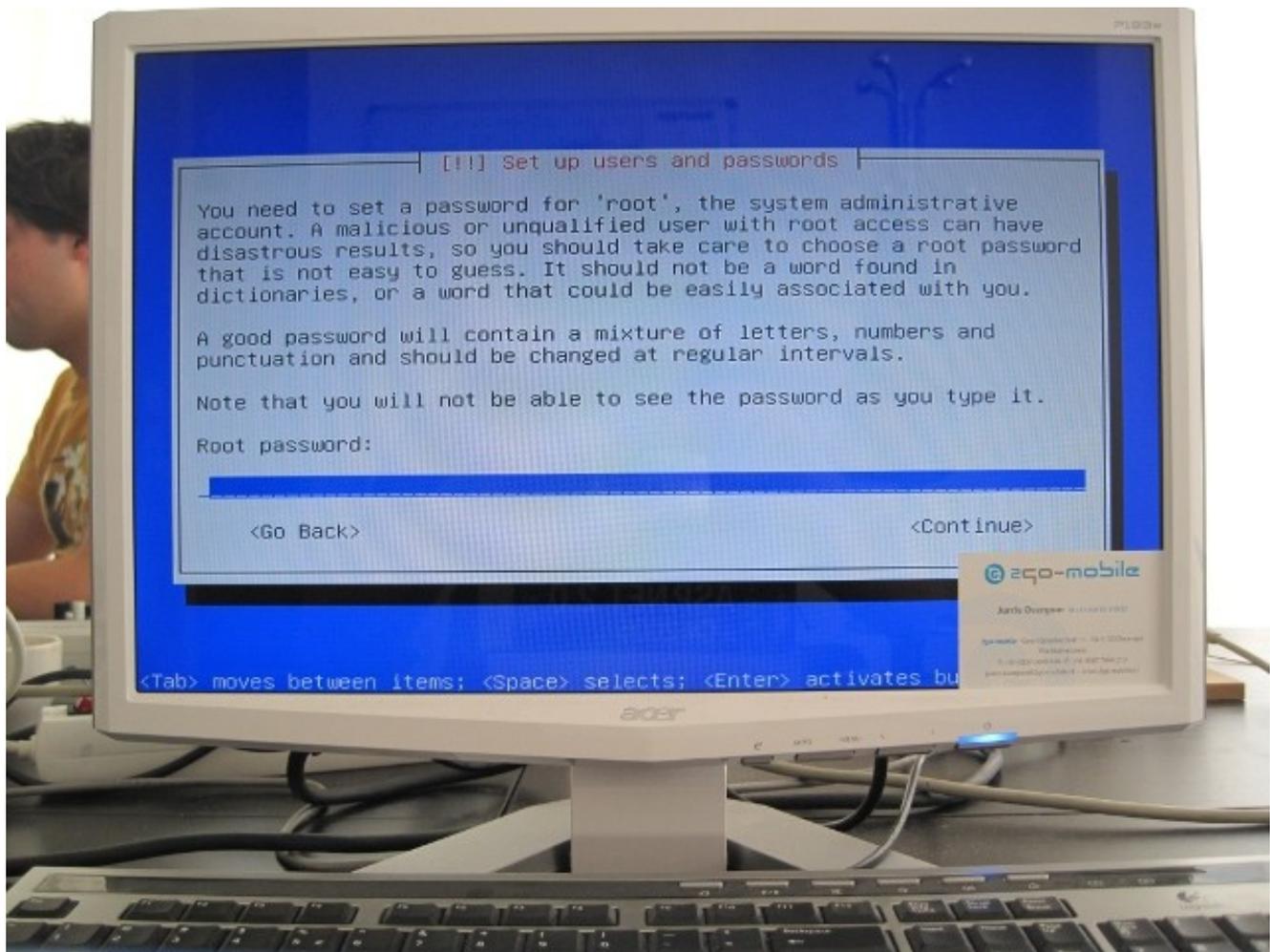
Step through, and the installer is finally installing the base system! Woohoo!



In this screenshot you see the installer installing the vanilla kernel... Uh oh, this is not correct. Before we close the installation and reboot, we need to remember to install our own kernel.

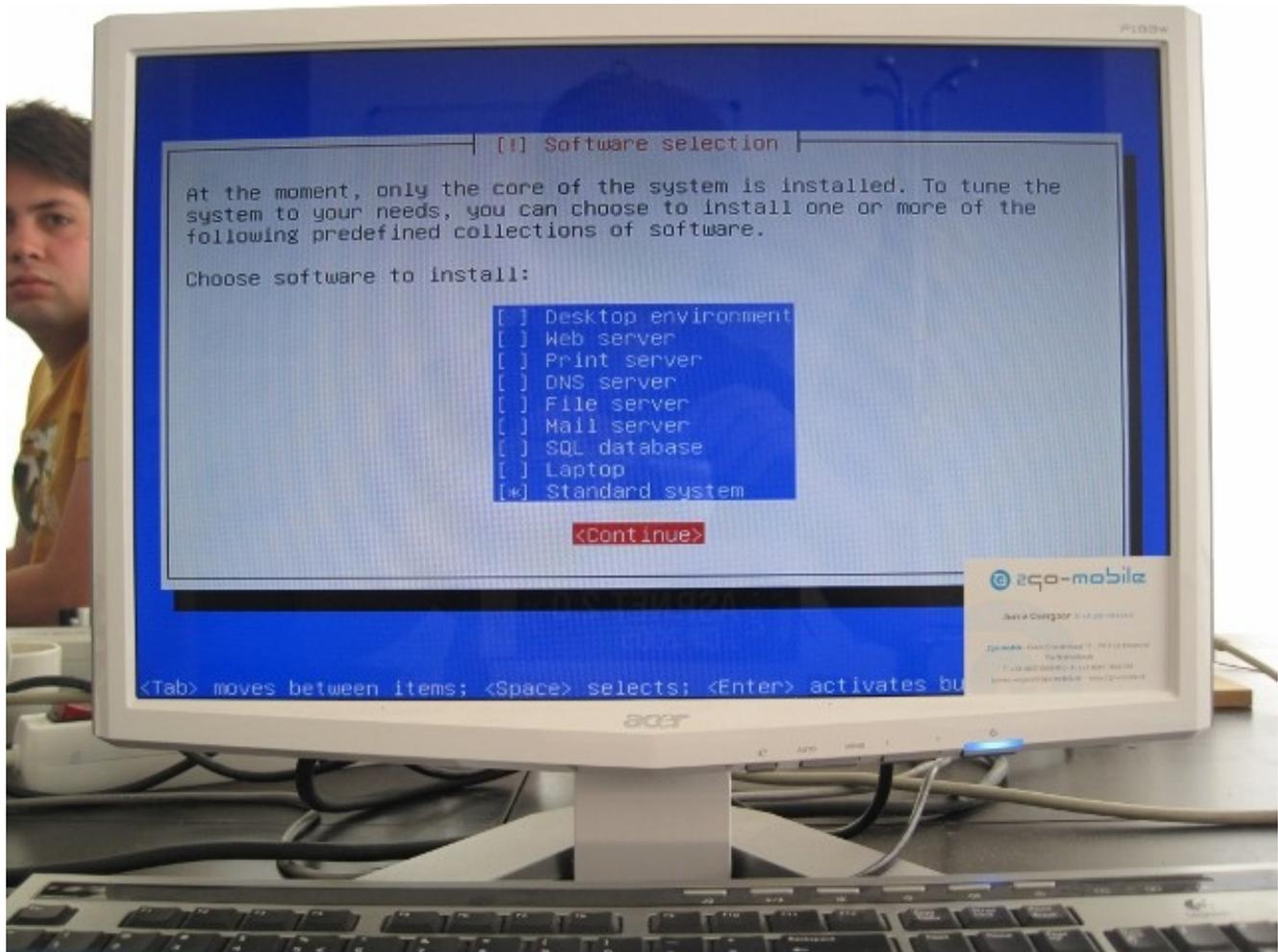


Keep following the installation. The installer will ask for your root password and for a normal user account.

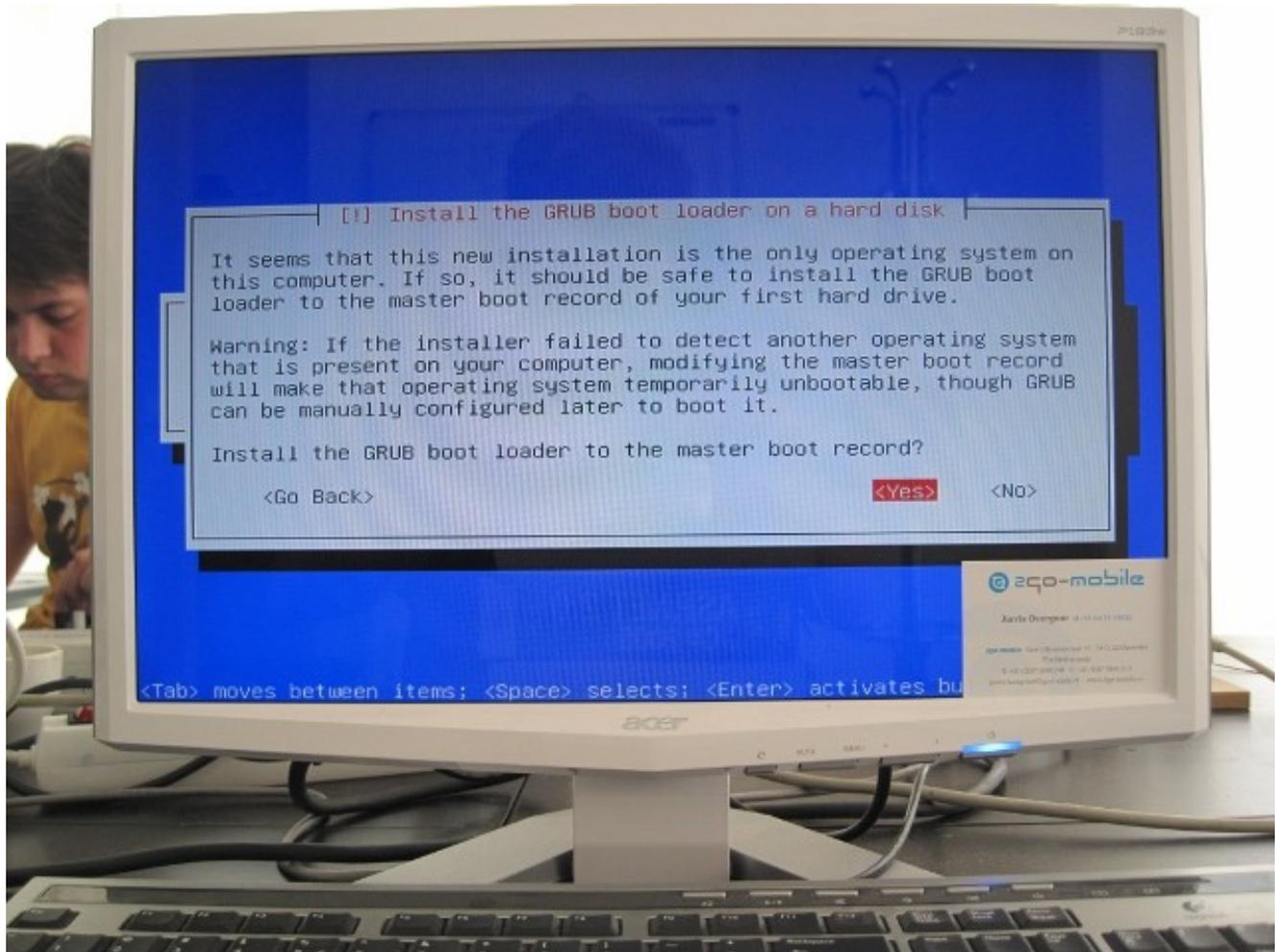


When the installer asks you which additional software you want to install, choose “standard system”. Don't be tempted to leave it at the core system – the “standard system” packages are needed to correctly install the Vortex86DX kernel package. If anyone can tell me what packages are needed, please tell me. I'd love to leave out “Standard system” and just install the bare minimum packages.

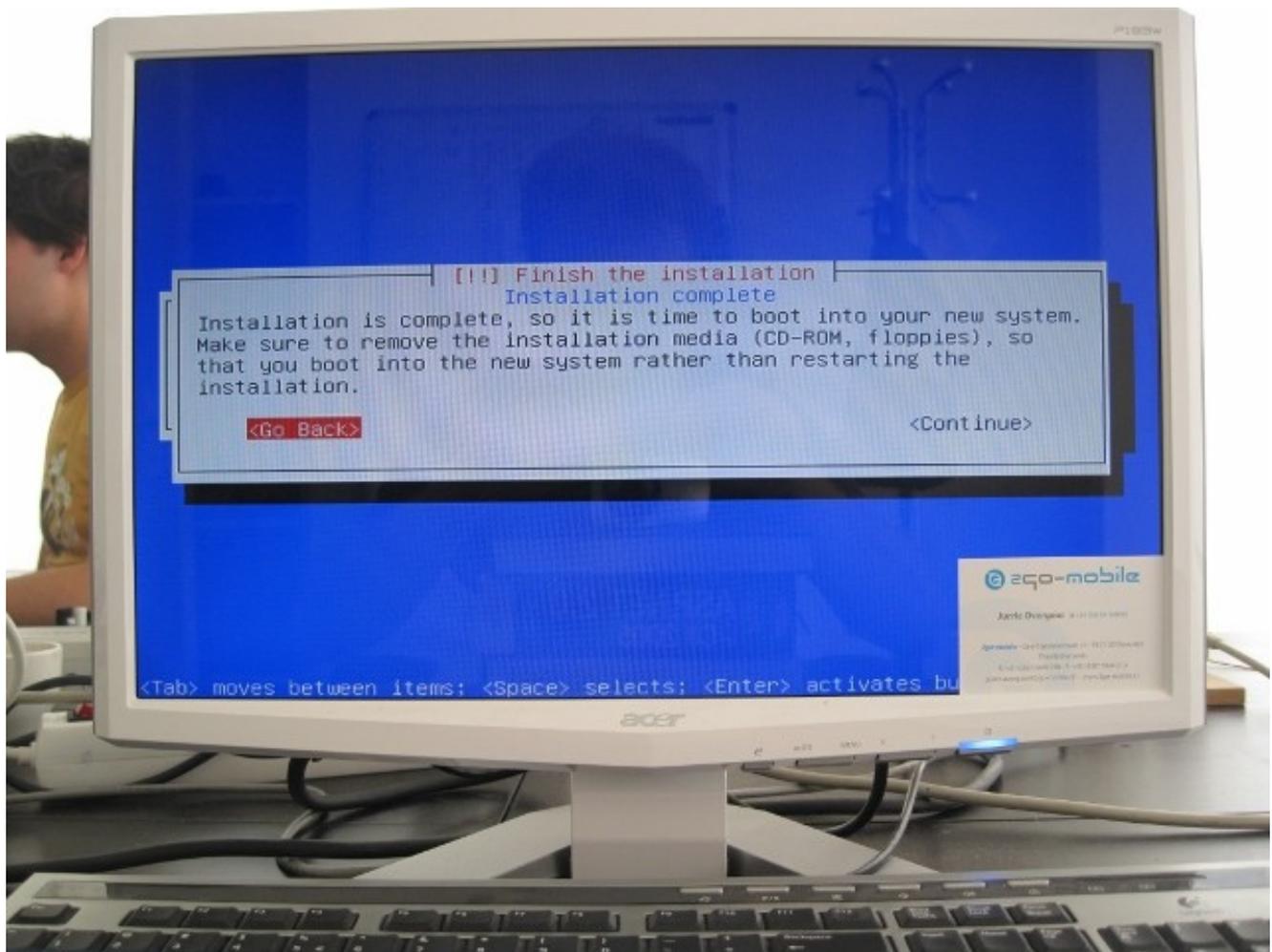
For now, select “Standard system” and choose “continue”.



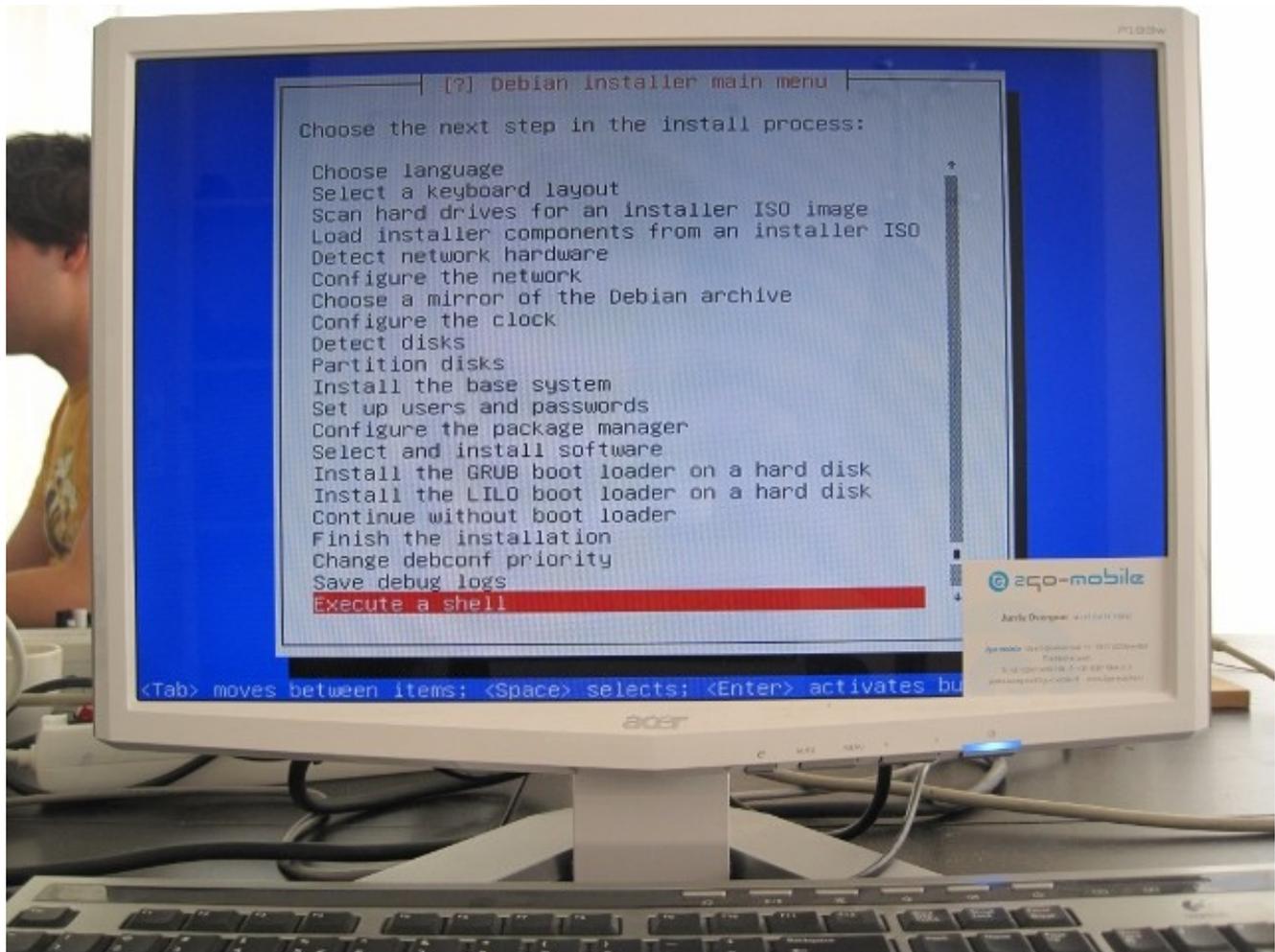
Continue the installation. The installer will ask you to install Grub (the boot loader) to the master boot record. Choose “yes”.



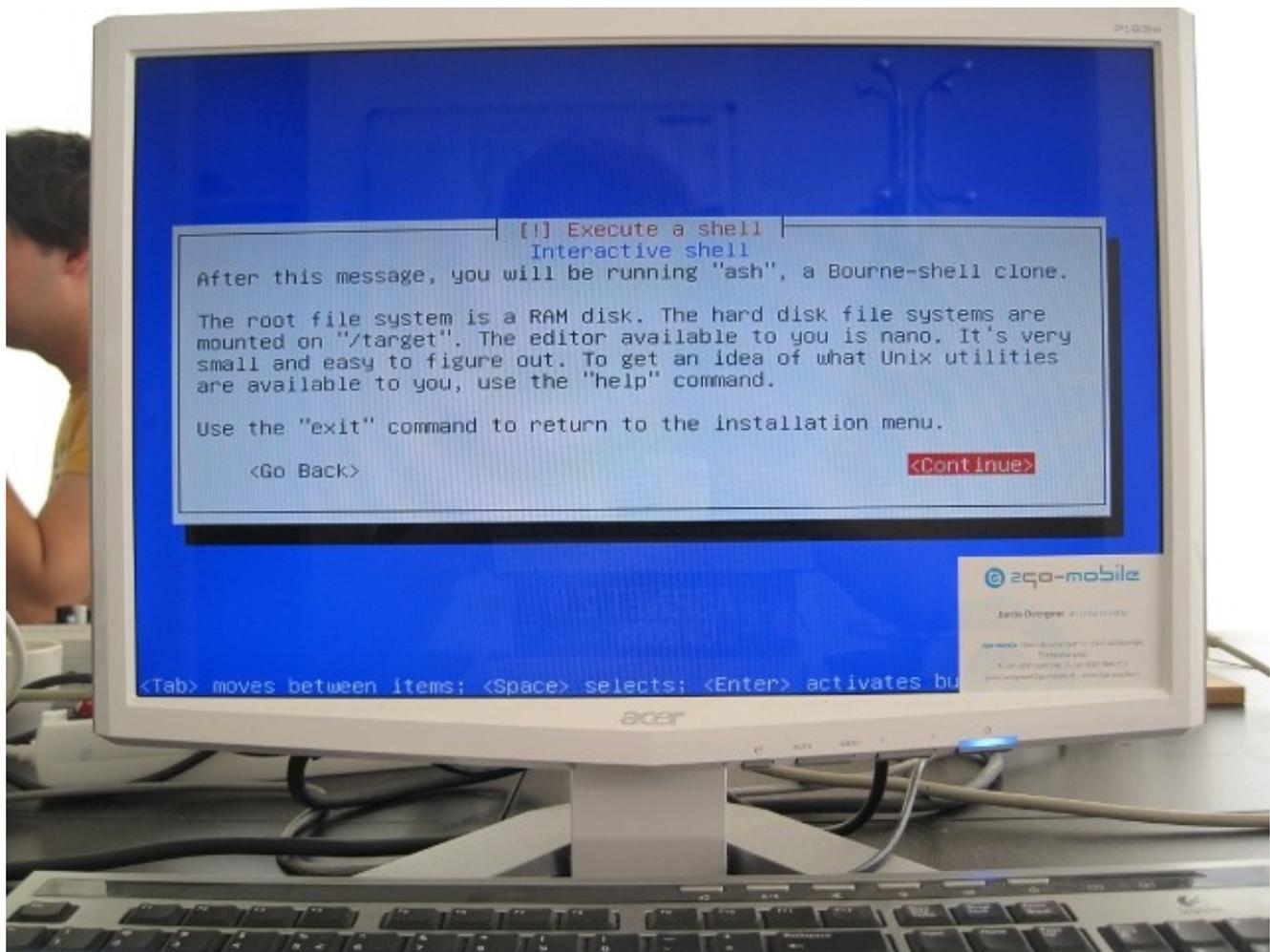
Continue the installation. Eventually you'll get a message saying that the installation is complete. We now need to install our Vortex86DX kernel, so don't choose "continue"! Instead, choose "Go Back".



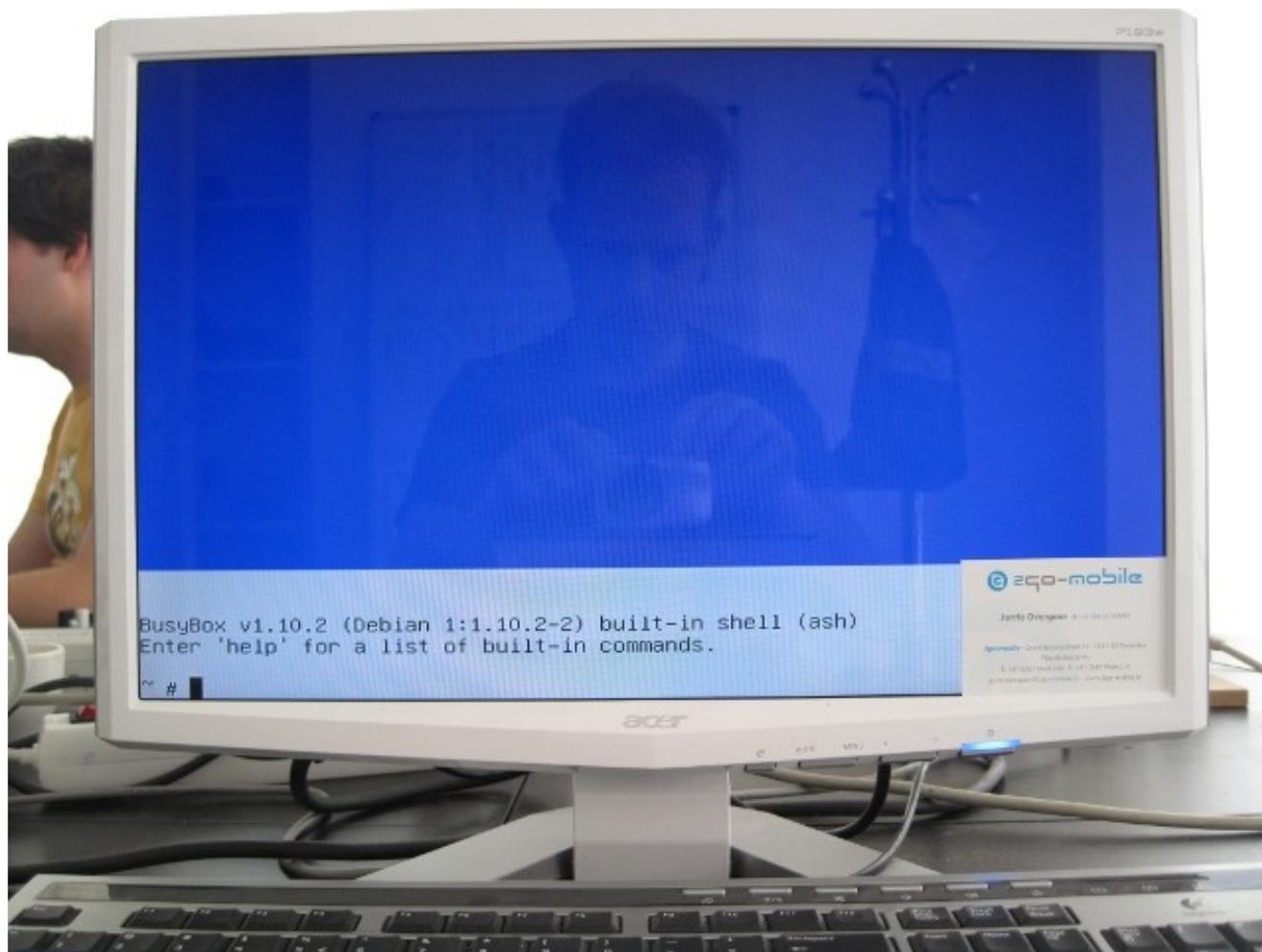
In the screen that follows, choose "Execute a shell".



You will be presented with a dialog that tells you what's about to happen, and how to get back to the installer. Choose "Continue".



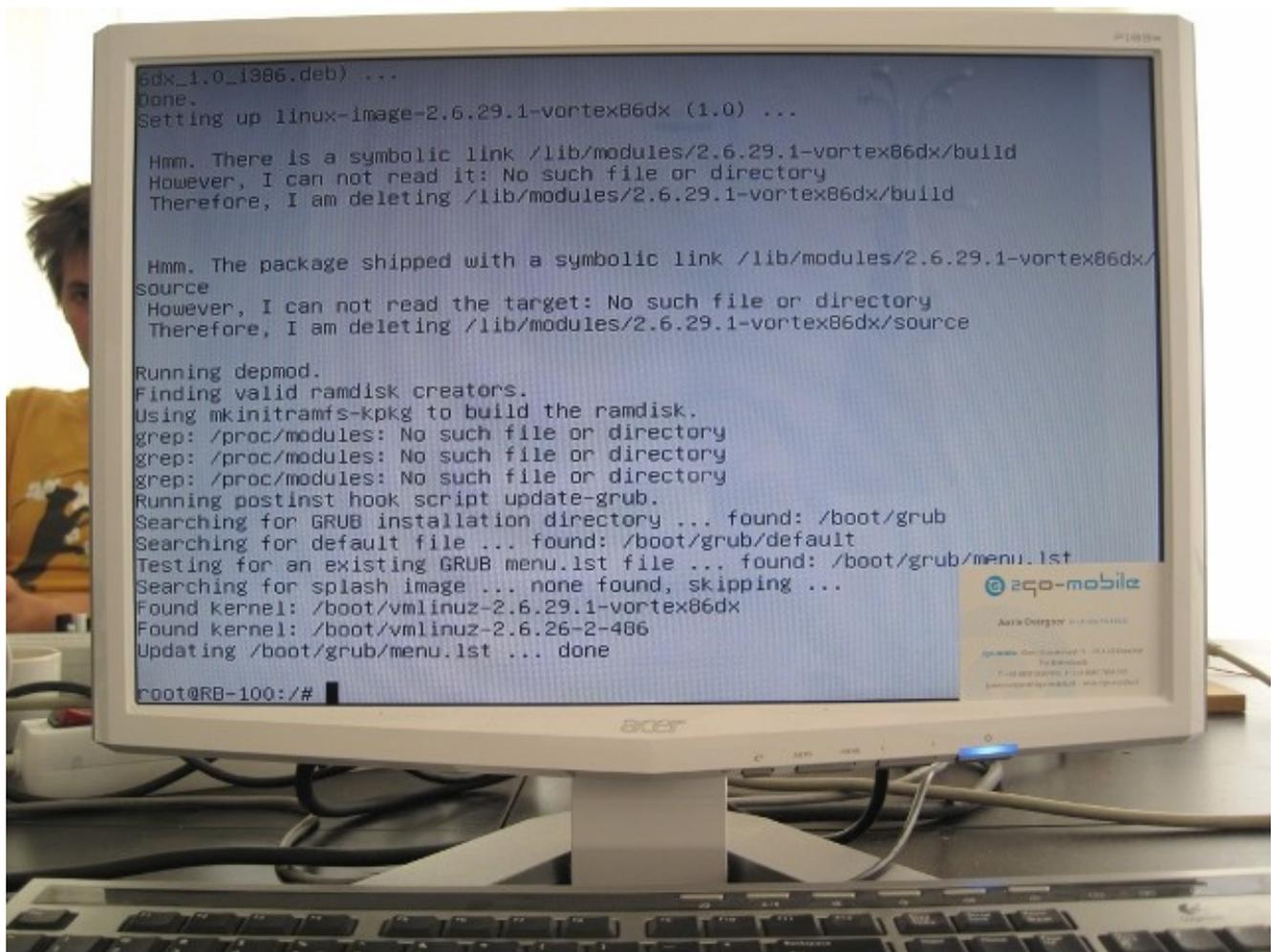
You'll be presented with a BusyBox shell.



Now it is time to insert some commands:

```
cp /hd-media/linux-image-2.6.29.1-vortex86dx_1.0_i386.deb  
/target/root/  
chroot /target /bin/bash  
dpkg -i /root/linux-image-2.6.29.1-vortex86dx_1.0_i386.deb
```

The last command will install the new kernel. The output will be something like the screen below. You can safely ignore the warnings about `/lib/modules/2.6.29.1-vortex86dx/build` and `/lib/modules/2.6.29.1-vortex86dx/source` being broken symbolic links.

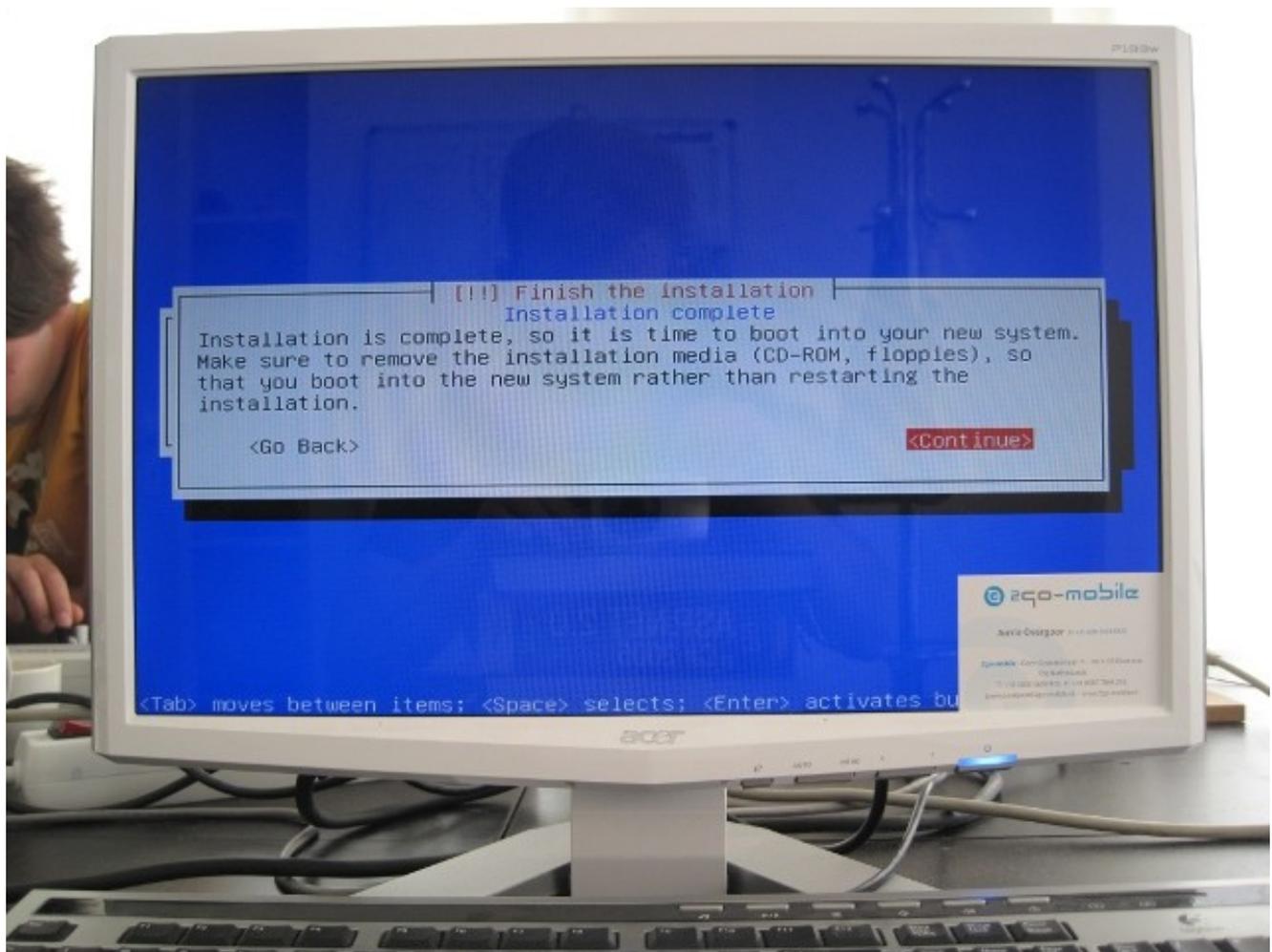


Now insert some more commands to exit the chroot jail and the BusyBox shell (we will go back to the installer).

exit

exit

When you are back in the list of options, choose “Finish the installation”. Follow the instructions, this time choose “continue” when you get the following screen:



The system will reboot. Grub will show, and will default to the Vortex86DX kernel. Debian will start, and eventually you should get your lovely login prompt:



More information

Check <http://wiki.debian.org/BootUsb> for information on booting from USB. Also, Google is your friend!

If you find errors in this document, or find some things unclear, please drop me a line at jurrie.overgoor@2go-mobile.nl.