Miscellaneous Linux Tricks

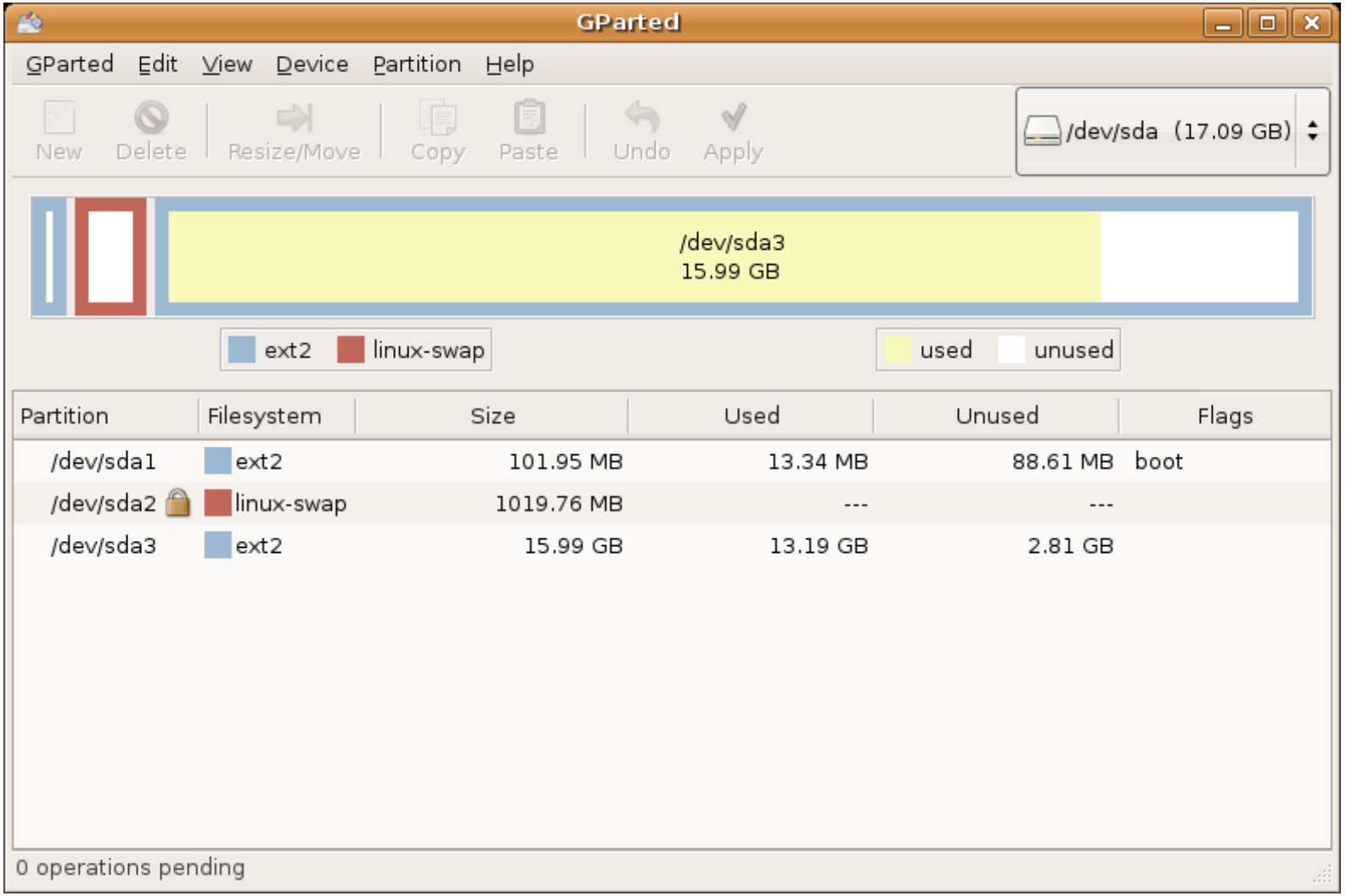
Paul Elliott
pelliott@io.com

http://www.FreeBlackPatchPanel.com/pme/linux/MiscLinux.pdf

Miscellaneous Linux Tricks

- Gnome Partition Editor (gparted) CD for partitions
- Mount lost partitions
- Linux boot floppy (with filesystem)
- Linux boot CD
- fetchmail with openssl

Gparted: Gnome Partition Editor



Gparted: Table of functionality

Are you sure you need to pay money for Partition Magic?

	Detect	Read	Create	Grow	Shrink	Move	Copy	Check	Required software
ext2	✓	✓	✓	✓	✓	✗	✓ [1]	✓	e2fsprogs
ext3	✓	✓	✓	✓	✓	✗	✓ [1]	✓	e2fsprogs
fat16	✓	✓	✓	✓ [4]	✓ [4]	✓ [4]	✓ [1]	✓	dosfstools
fat32	✓	✓	✓	✓ [4]	✓ [4]	✓ [4]	✓ [1]	✓	dosfstools
hfs	✓	✓	✓	✗	✓ [4]	✗	✓ [1]	✗	hfsutils
hfs+	✓	✓	✗	✗	✓ [4]	✗	✓ [1]	✗	
jfs	✓	✓	✓	✓ [2]	✗	✗	✓ [1]	✓	jfsutils
linux-swap	✓	✗	✓	✓	✓	✓	✓ [1]	✗	mkswap (part of util-linux)
ntfs	✓	✓	✓	✓	✓	✗	✓	✓	ntfsprogs
reiserfs	✓	✓	✓	✓	✓	✗	✓ [1]	✓	reiserfsprogs
reiser4	✓	✓	✓	✗	✗	✗	✓ [1]	✓	reiser4progs
ufs	✓	✗	✗	✗	✗	✗	✗	✗	
xfs	✓	✓	✓	✓ [2]	✗ [3]	✗	✓	✓	xfsprogs

[1] The actual copy is performed by 'dd'.
[2] You need kernelsupport for this filesystem if you want to grow it (module needs to be loaded).
[3] Although it's not possible to shrink an xfs filesystem directly, you can shrink it using GParted's copy functionality.
[4] Native available through libparted.

Use the gparted LiveCD

gparted comes with Ubuntu LiveCD and perhaps others.
The Gparted LiveCD does not lock any existing partitions.
Find at: http://gparted.sourceforge.net/livecd.php
download the iso and burn with K3B!
Boot the CD to use gparted!

Hey, save your partition setup!

Save your partition table:
sfdisk -d /dev/hda > hda.pt
To restore:
sfdisk /dev/hda < hda.pt

Save your MBR!

Save you MBR (Master boot record)
dd if=/dev/hda of=hda.mbr bs=512 count=1
to restore:
dd if=hda.mbr of=/dev/hda
back these files up in case of disk crash.
To restore the primary partition table without overwriting the MBR type:
dd if=hda.mbr of=/dev/hda bs=1 count=64 \
skip=446 seek=446

Mount lost partitions!

- Suppose your partition table is wiped out.
- But the partitions themselves are intact!
- And you know where the partitions are!
- Then you can recover!

fdisk -l -u /dev/sdb

- Disk /dev/sdb: 73.4 GB, 73407865856 bytes
- 255 heads, 63 sectors/track, 8924 cylinders, total 143374738 sectors
- Units = sectors of 1 * 512 = 512 bytes
-

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		59504760	143364059	41929650	83	Linux
/dev/sdb2		53207280	59504759	3148740	83	Linux
/dev/sdb3		63	16064	8001	83	Linux
/dev/sdb4		16065	53207279	26595607+	83	Linux

-
- 16065*512=8225280 so offset of /dev/sdb4 is 8225280 bytes.

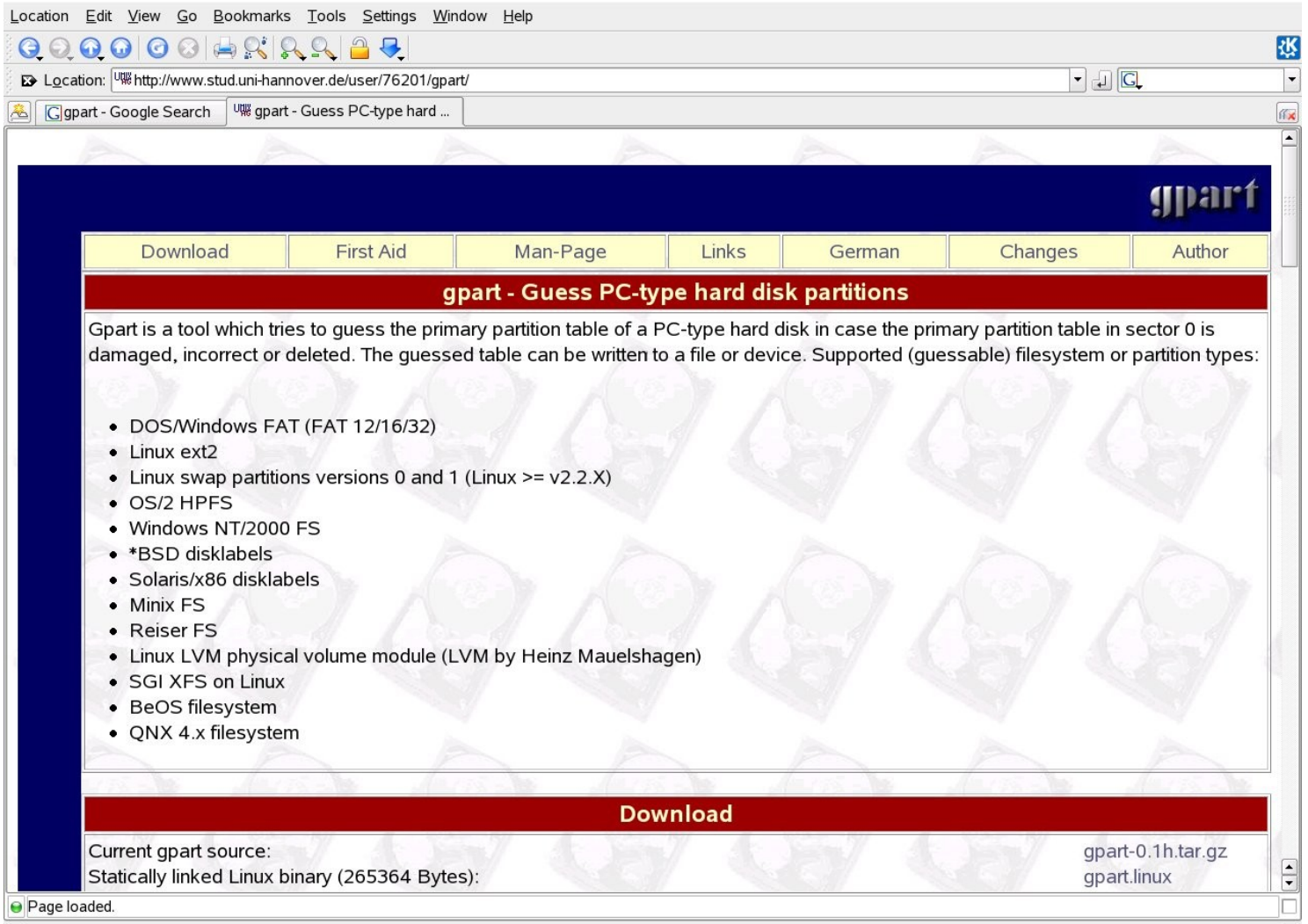
Suppose our partition table is wiped out!

- Of course, getting the offset by doing fdisk -l would be impossible, that uses the partition table.
- mounting the partition directly, would also be impossible for the same reason.
- suppose we had the offset by other means such as:
- scanning for first bock of partition. (gpart)
- we wrote it down before hand.

gpart - guess PC-type hard disk partitions

- gpart is a program that scans a hard disk looking for blocks that look like the start of a partition! It prints the offset (in 1K blocks) out. Does not use the partition table (by default).
- Read the gpart manual page!

http://www.stud.uni-hannover.de/user/76201/gpart/



losetup to the rescue!

- losetup is usually used for loopback encryption.
- But it can be used to create a device which is an offset into another device!
- losetup -o 8225280 /dev/loop0 /dev/sdb
- You can now mount the partition by mounting the loop back device.
- mount -r /dev/loop0 /mnt
- Now you can deal with the files and directories like any other mounted device.

Windblows loves to overwrite the MBR!

- windoze likes to overwrite the master boot record. (note to all you dual boot people.)
- some Distros do not like to share the MBR! (if you install multiple distros on different partitions.
- So you need a boot floppy, or boot CD!

Boot floppy with filesystem is most flexible.

- Grub manual describes a different method.
- filesystem method allows you to easily change the menu, device map.
- extra space available for additional info.

Boot floppy with filesystem: Step 1

- make filesystem on floppy.
- `# mke2fs /dev/fd0`
- eliminate filesystem checks.
- `# tune2fs -c 0 /dev/fd0`
- `# tune2fs -c 0 /dev/fd0`

Boot floppy with filesystem: Step 2

- mount the filesystem, copy the grub files.
- `# mount /dev/fd0 /mnt`
- `# mkdir –p /mnt/boot/grub`
- `# for x in   \`
- `  /usr/lib/grub/*;do`
- `    cp $x /mnt/boot/grub;`
- `done`

Boot floppy with filesystem: Step 3

- Setup the grub floppy
- `# grub`
- `grub> root (fd0)`
- `grub> setup (fd0)`
- `grub> quit`

Now would be a good time to save our work as a floppy image!

- `#umount /mnt`
- `# dd bs=18k if=/dev/fd0 \ of=grubFS.flp.img`
- `# mount /dev/fd0 /mnt`

Boot floppy with filesystem: Step 4

- Copy the menu, and device map.
- `#`
  `cp /boot/grub/menu.lst \ /mnt /boot/grub/menu.lst`
- `#`
  `cp /boot/grub/device.map \ /mnt/boot/grub/device.map`
- This step can be redone for a different grub setup!

What does a menu look like?

```
# Modified by YaST2. Last modification on Sun Jun 18 11:27:22 CDT 2006

color white/blue black/light-gray
default
timeout 8
gfxmenu (hd0,0)/message

###Don't change this comment - YaST2 identifier: Original name: linux###
title SUSE Linux 10.1
    root (hd0,0)
    kernel /vmlinuz root=/dev/sda3 vga=0x31a resume=/dev/sda2 splash=silent \
showopts
    initrd /initrd
```

Boot floppy with filesystem: Step 5: extra information.

- There is plenty of extra space on the floppy.
- Why not save the partition table, MBR in case we need it later?
- `# sfdisk -d /dev/sda \`
  `> /mnt/sda.pt`
- `# dd bs=512 count=1 \`
  `if=/dev/sda \`
  `of=/mnt/sda.mbr`

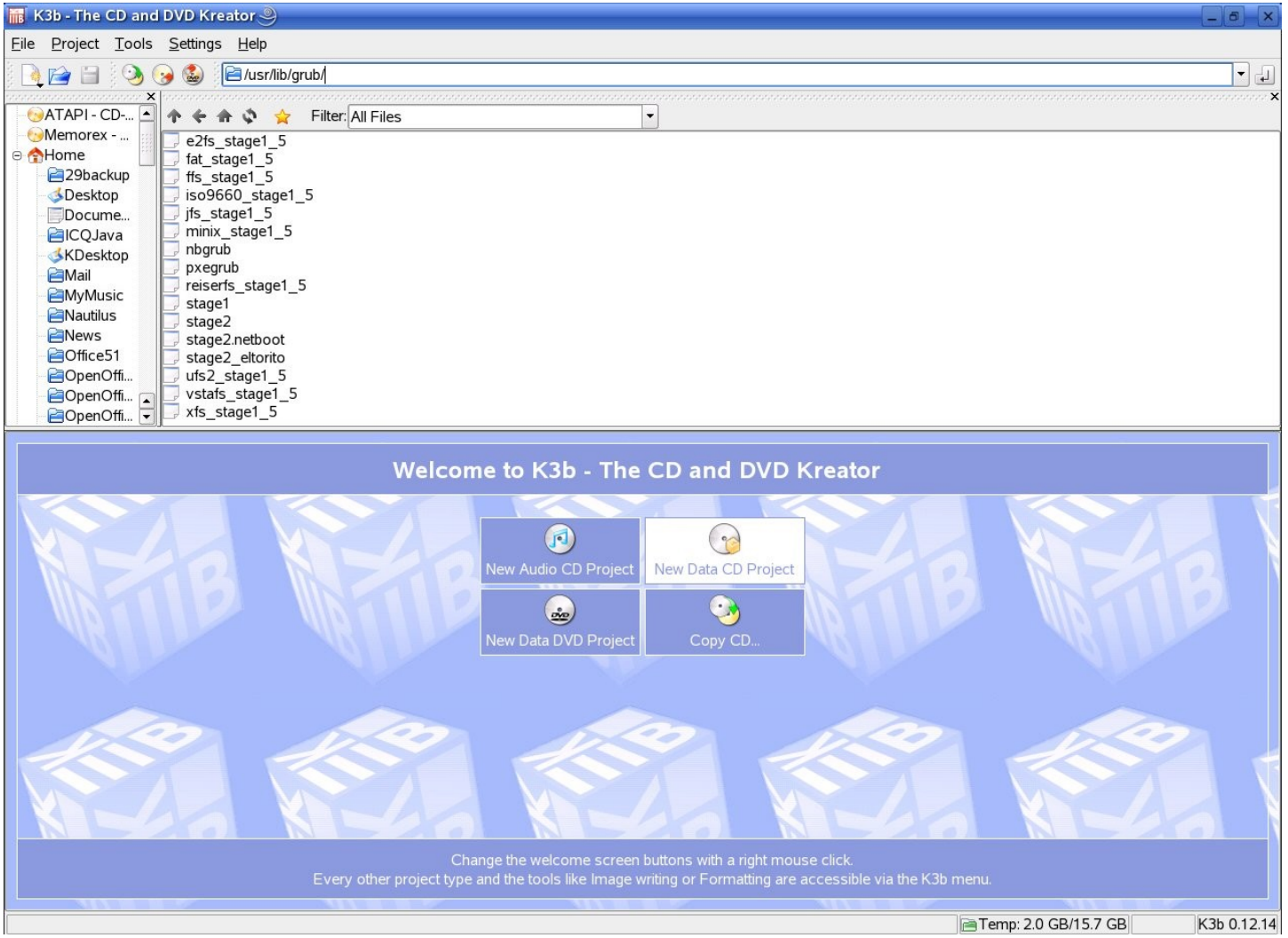
Boot floppy with filesystem: Step 6

- Don't forget to unmount the filesystem before removing the floppy.
- `# umount /mnt`

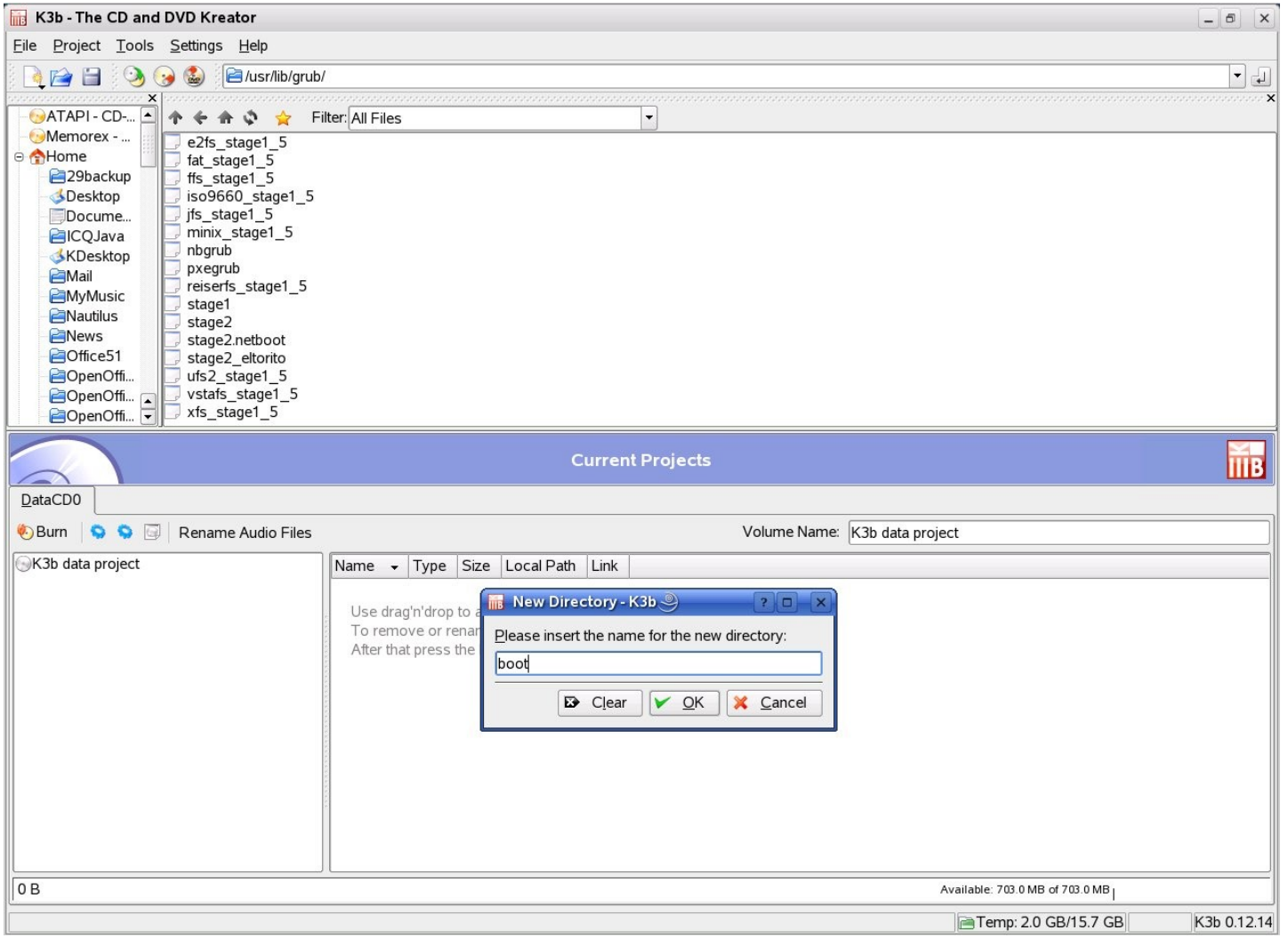
Build a boot CD with K3B.

- Many newer computers have CD player but no floppy!
- Grub manual describes how to build boot CD from the command line.
- http://www.gnu.org/software/grub/manual/
- We describe the point and clicky way: K3b

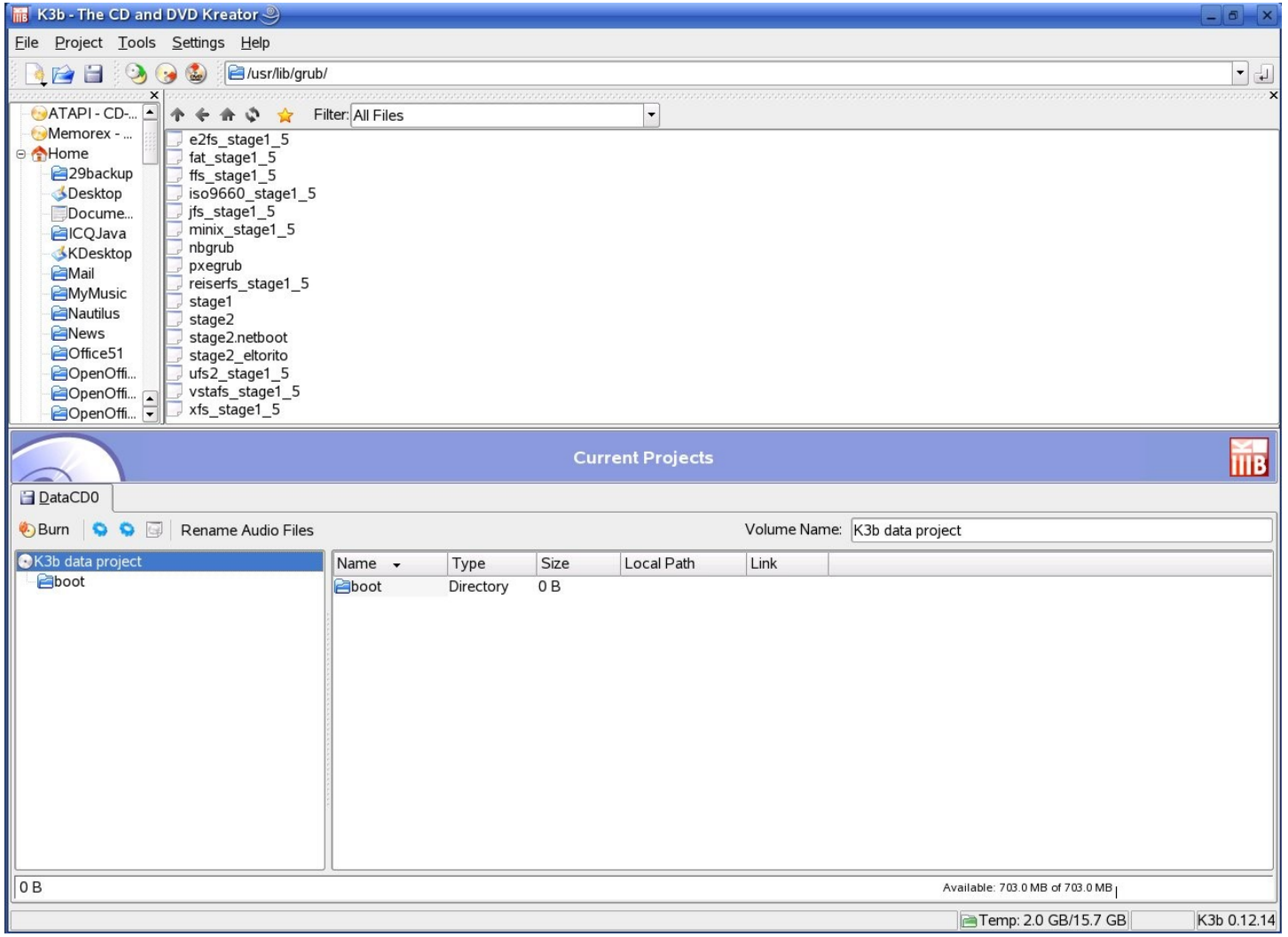
K3b: New Data CD project



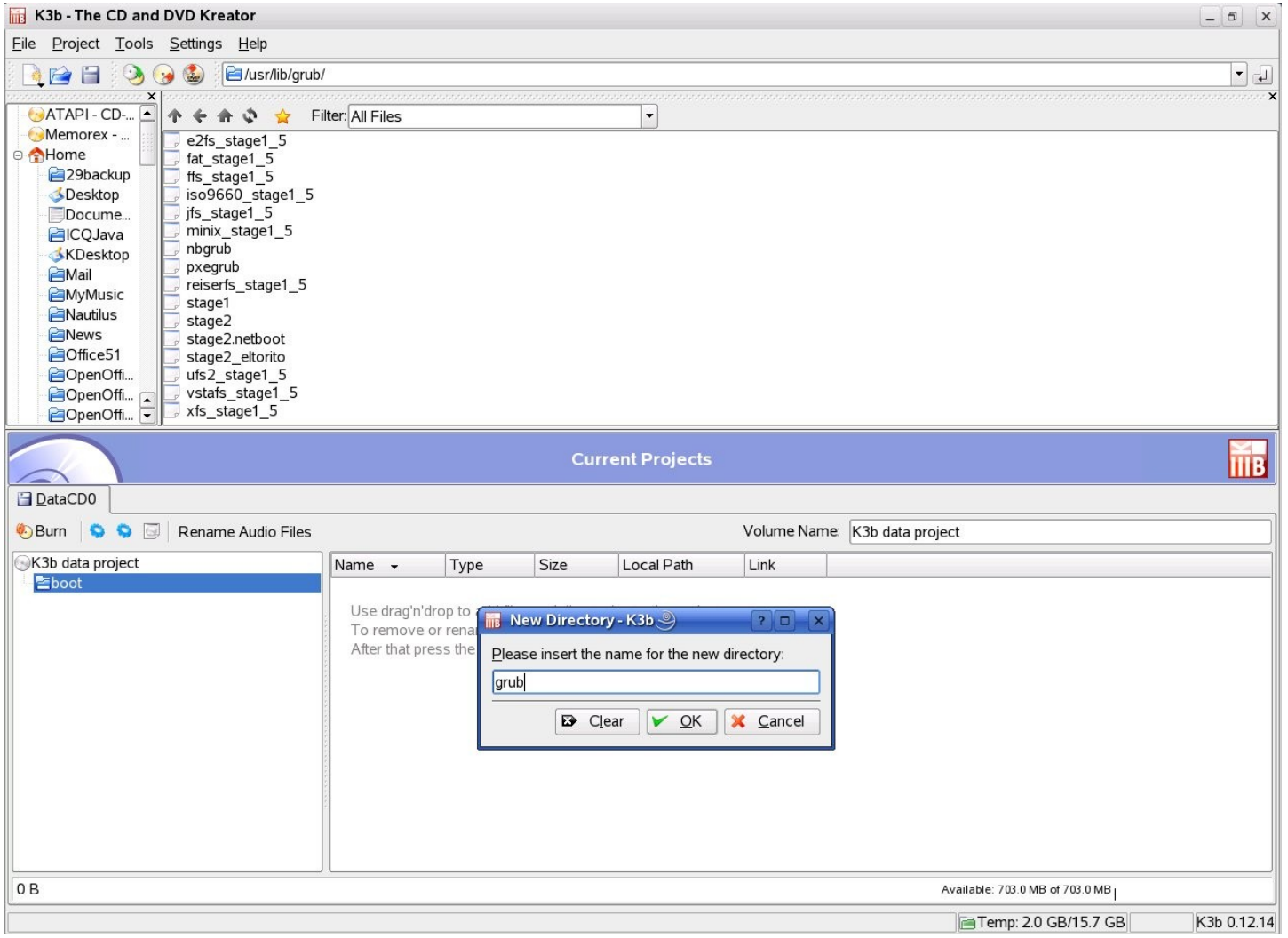
K3b: Right click, New directory to create /boot directory on the CD.



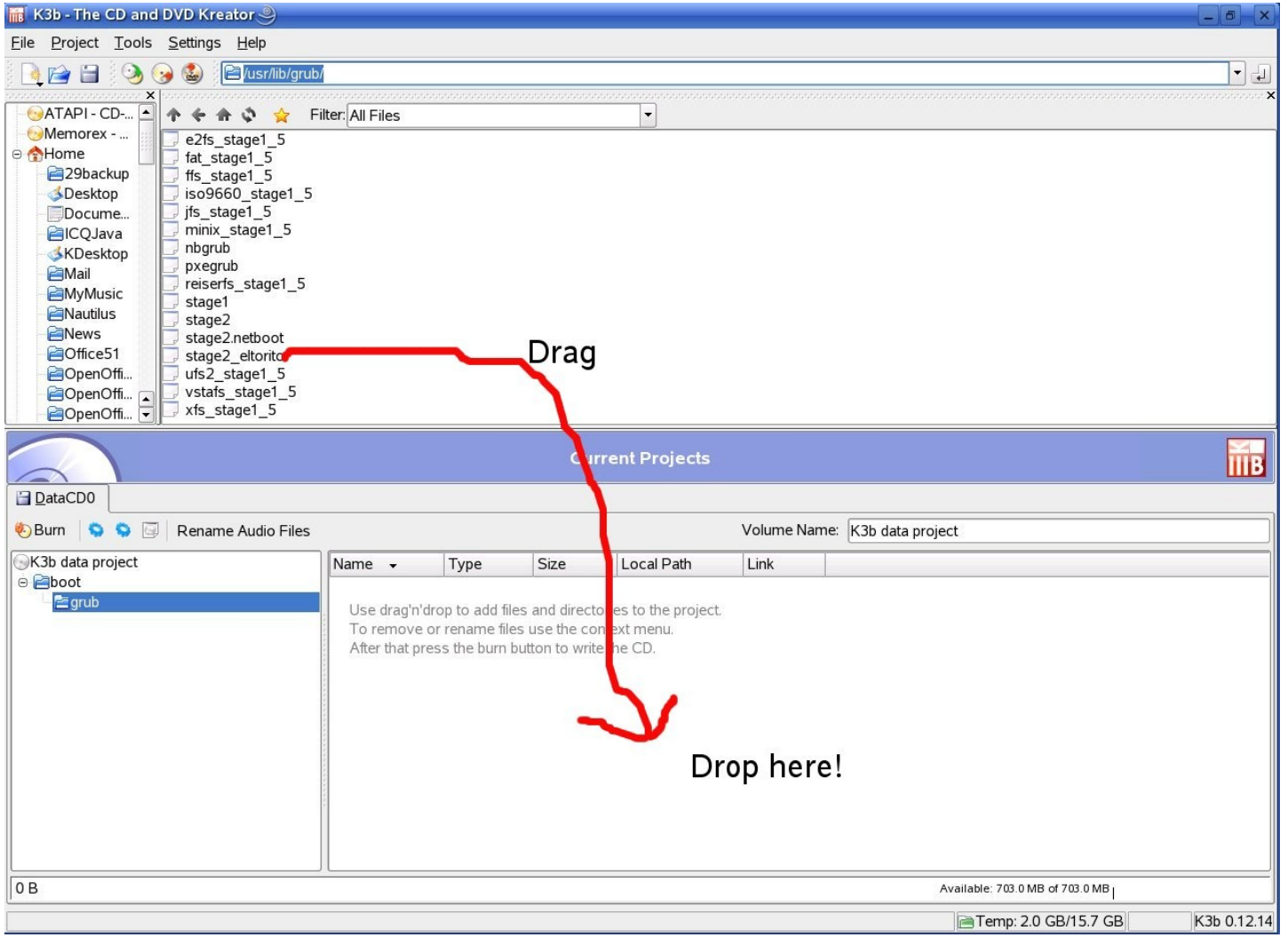
K3b: click on boot directory to enter it.



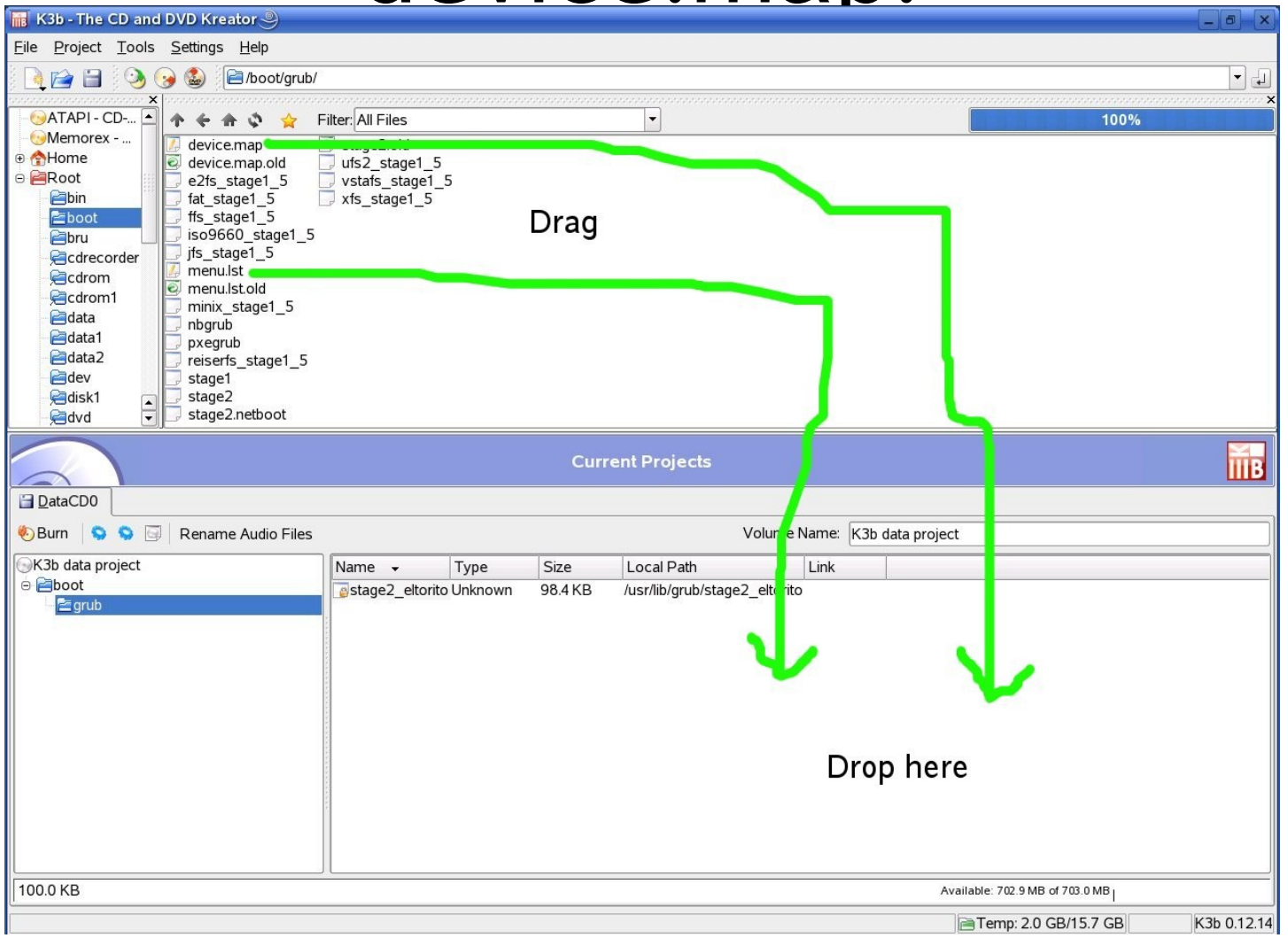
as before, create grub subdirectory within /boot and enter it.



Drag /usr/lib/grub/stage2_eltorito into /boot/grub on the CD.



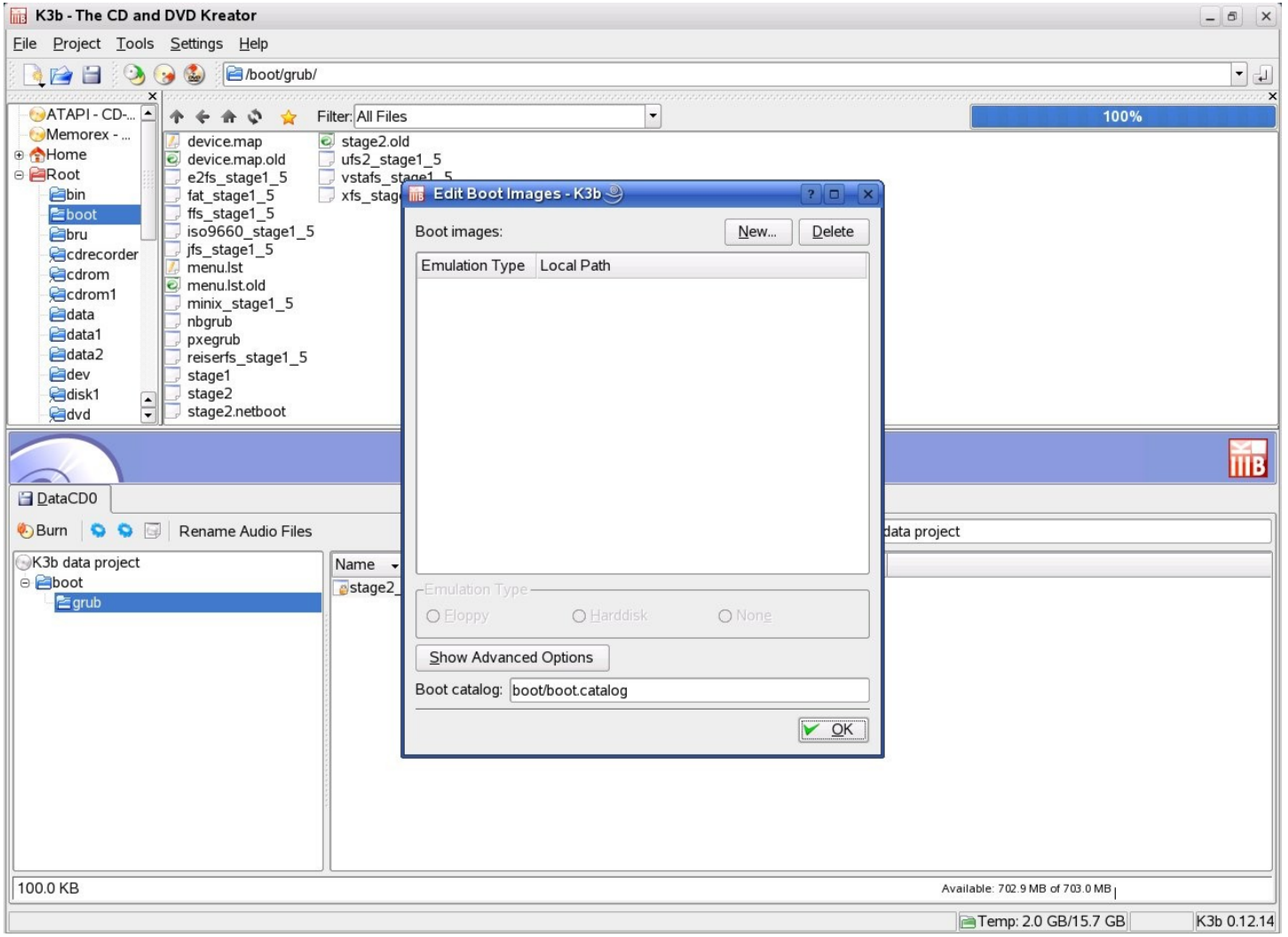
In the same way drag /boot/grub/menu.lst and device.map.



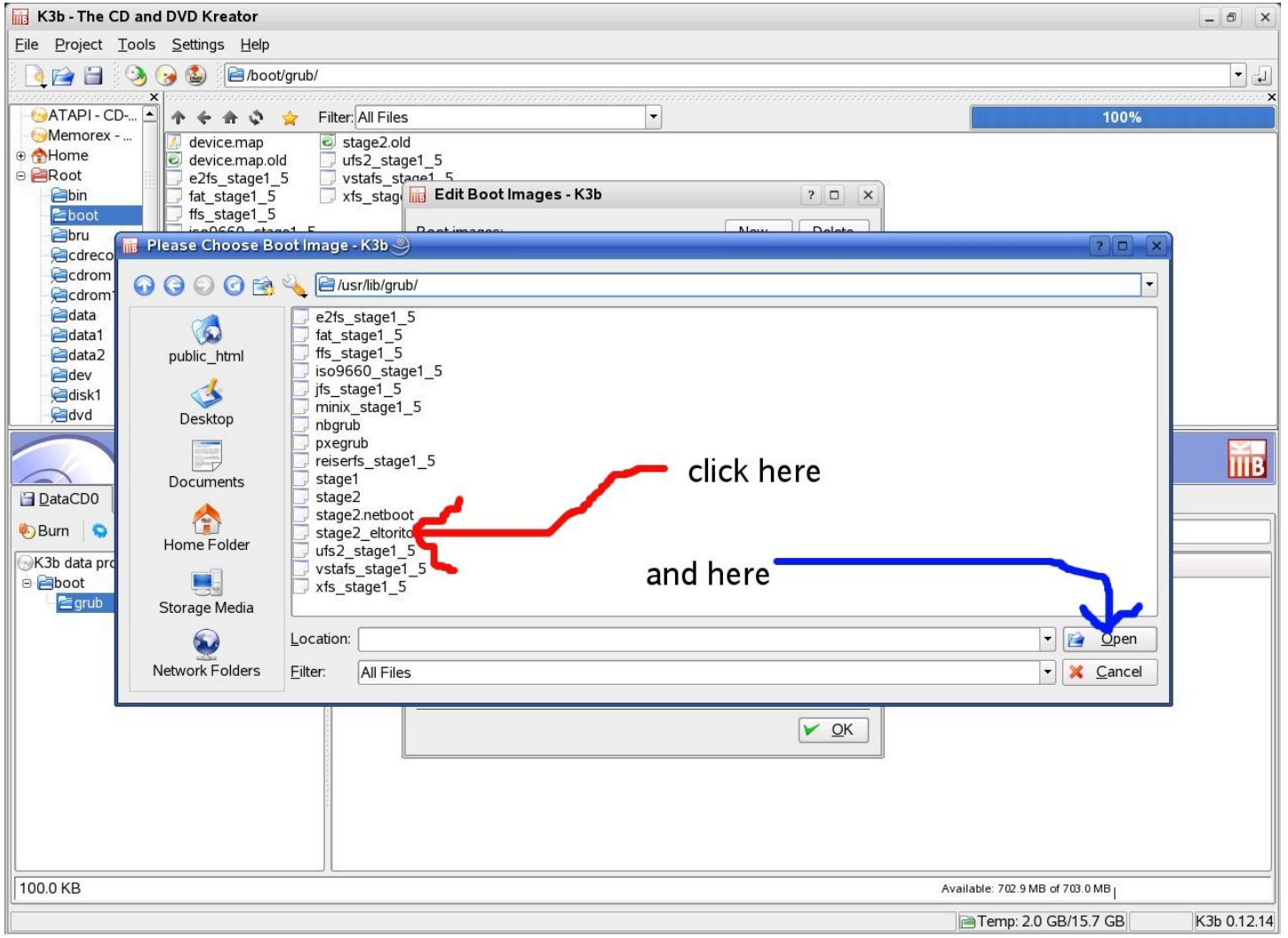
At this point we could add any additional info to the CD.

- like the partition table info and the MBR info, like we did for the floppy.
- In the interests of brevity, we will skip this step.

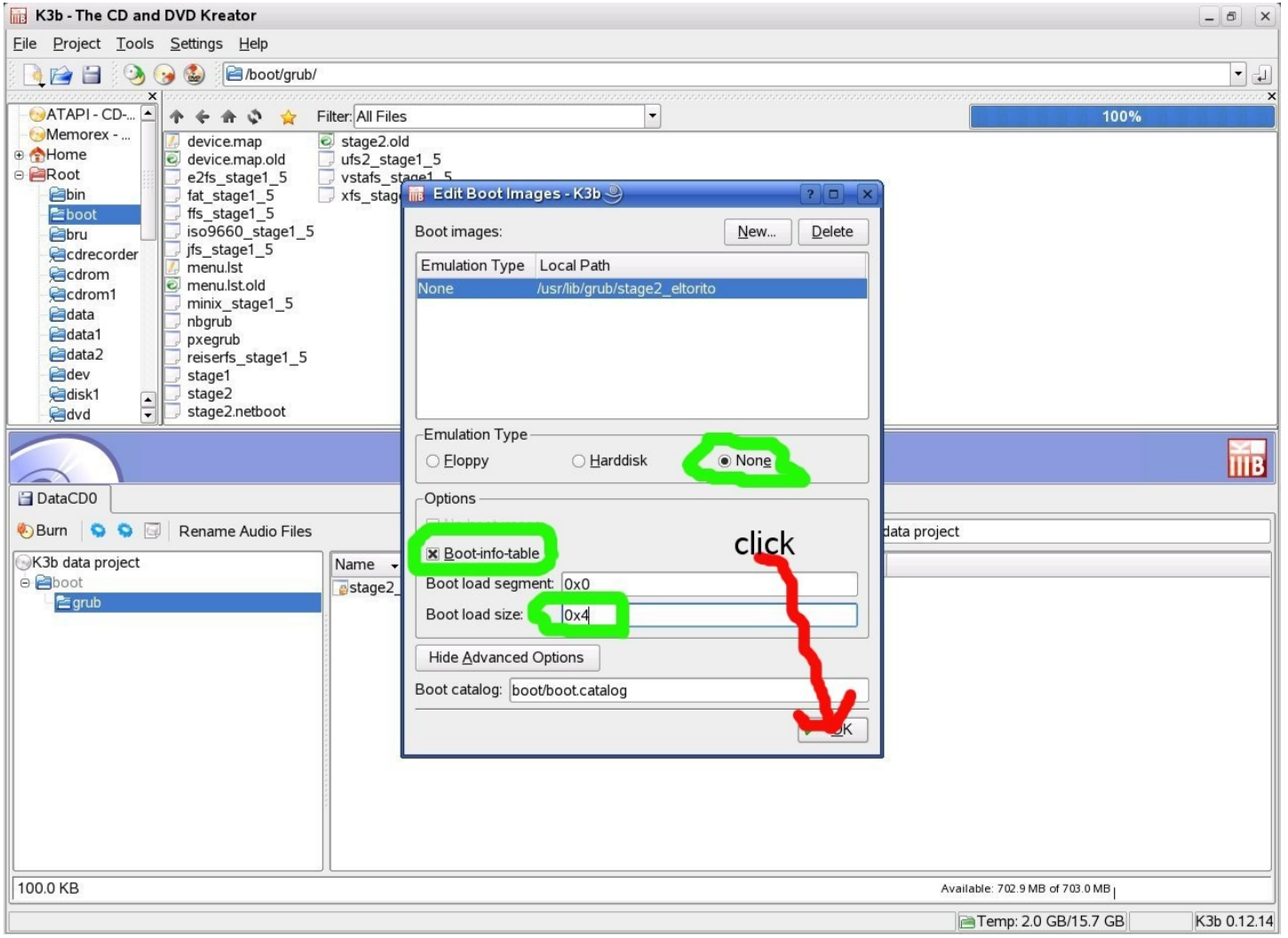
Now click on Project\Edit Boot Images...



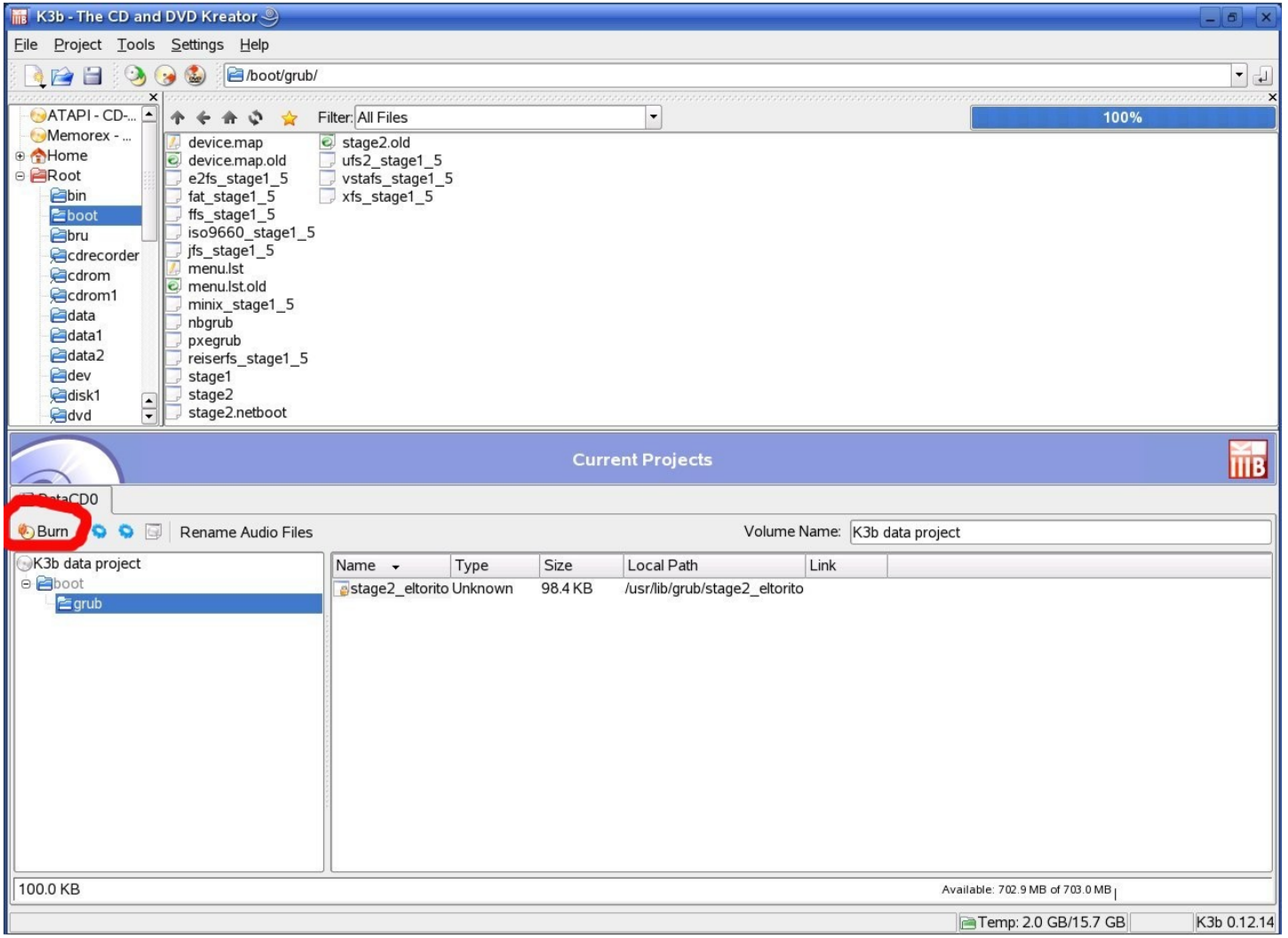
Click on Advanced options and New and open stage2_eltorito



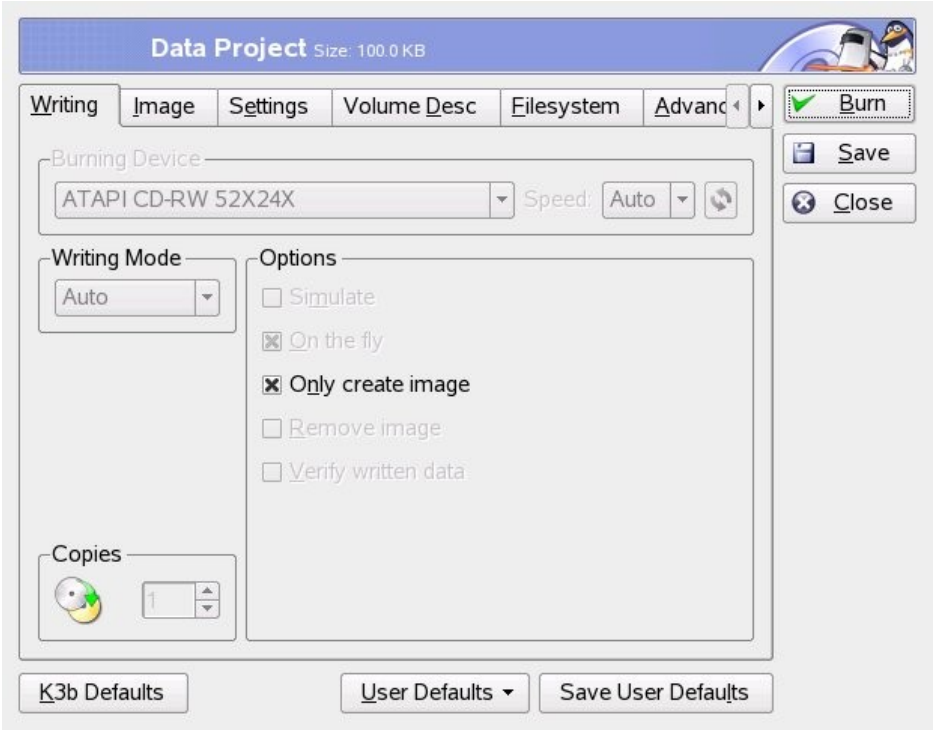
click on none, boot-info-table and set boot load size to 0x4



Almost done, click burn



Click on “Only Create Image”



- For some reason, direct burning does not work.
- If the local computer does not have a CD burner, the iso can be transferred to one that does.
- Use tools\Burn CD Image... to burn iso.

Note to all distro authors.

- The grub manual describes command line procedure, to create Boot CDs.
- Many new computers do not have floppy.
- All distros should have a “point and clicky” option to create boot a CD, just like the one to create a boot floppy.
- This code should be fairly easy to write.

use fetchmail with openssl!

- If you connect to your ISP's imap via modem or the ISP's cable line you probably don't need any encryption.
- Modem conversations are almost impossible to decypher.
- If cable line is insecure, then ISP's host probably is insecure too.

But if sometimes you connect to your ISP via the open Internet....

- then you can worry.
- You don't want to have your password moving in the clear over the open Internet!
- Solution: use ssl with fetchmail!
- ssl encrypts passwords and prevents man-in-the-middle attacks!

Step1: run fetchmail -v

- check that you do not already have an ssl command in .fetchmailrc
- fetchmail -v
- look for the first line that starts like this:
- fetchmail: IMAP< * OK [CAPABILITY
- check for the capability “STARTTLS” in this line.
- If not there, you are out of luck, your imap server does not support ssl. Otherwise, goto next step.

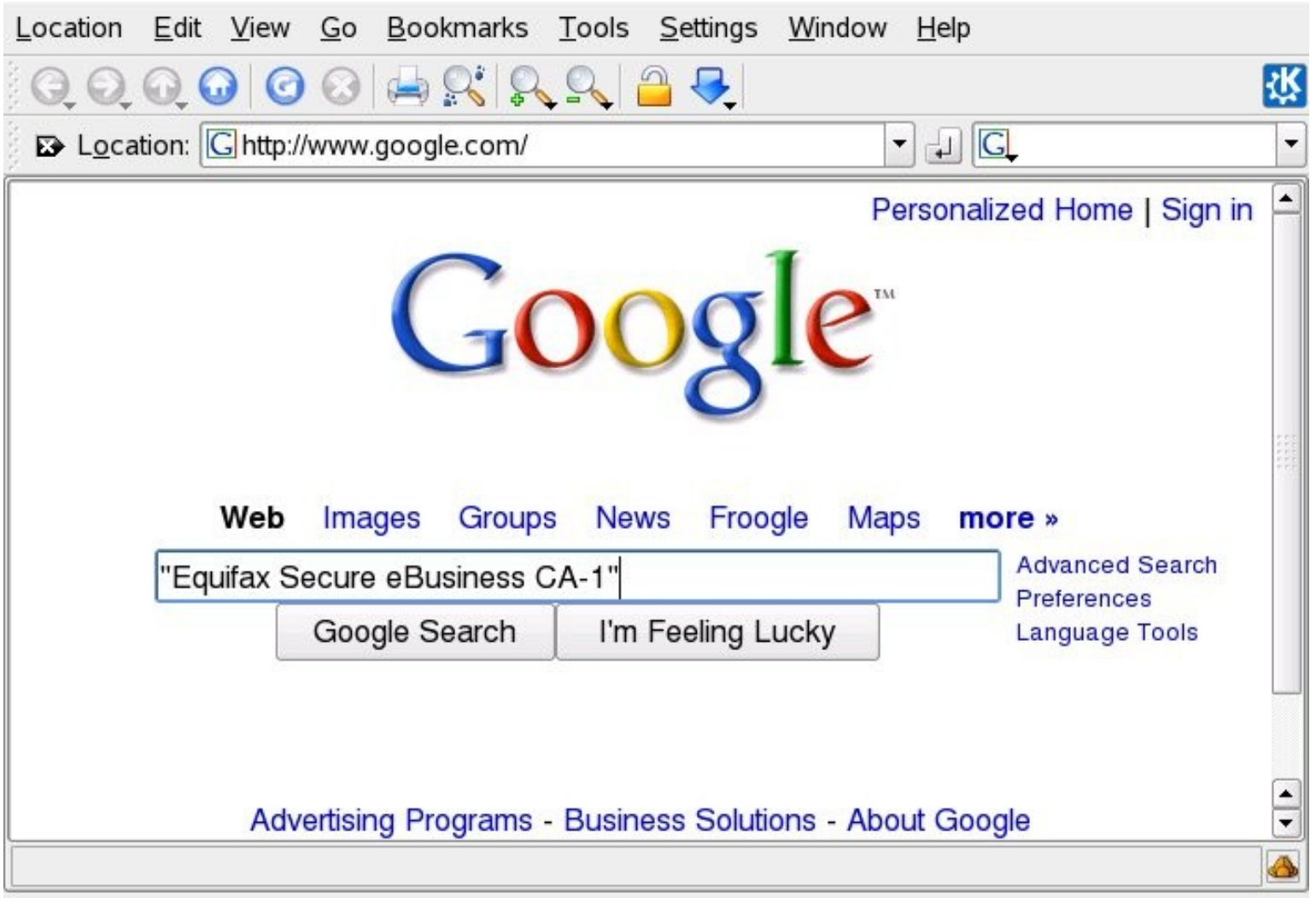
Step2: run fetchmail -v --ssl

- look for errors that look like this:
- `fetchmail: Server certificate verification error: unable to get local issuer certificate`
- `fetchmail: Server certificate verification error: certificate not trusted`
- `fetchmail: Server certificate verification error: unable to verify the first certificate`
- no errors: done, add ssl to .fetchmailrc
- otherwise continue to step 3

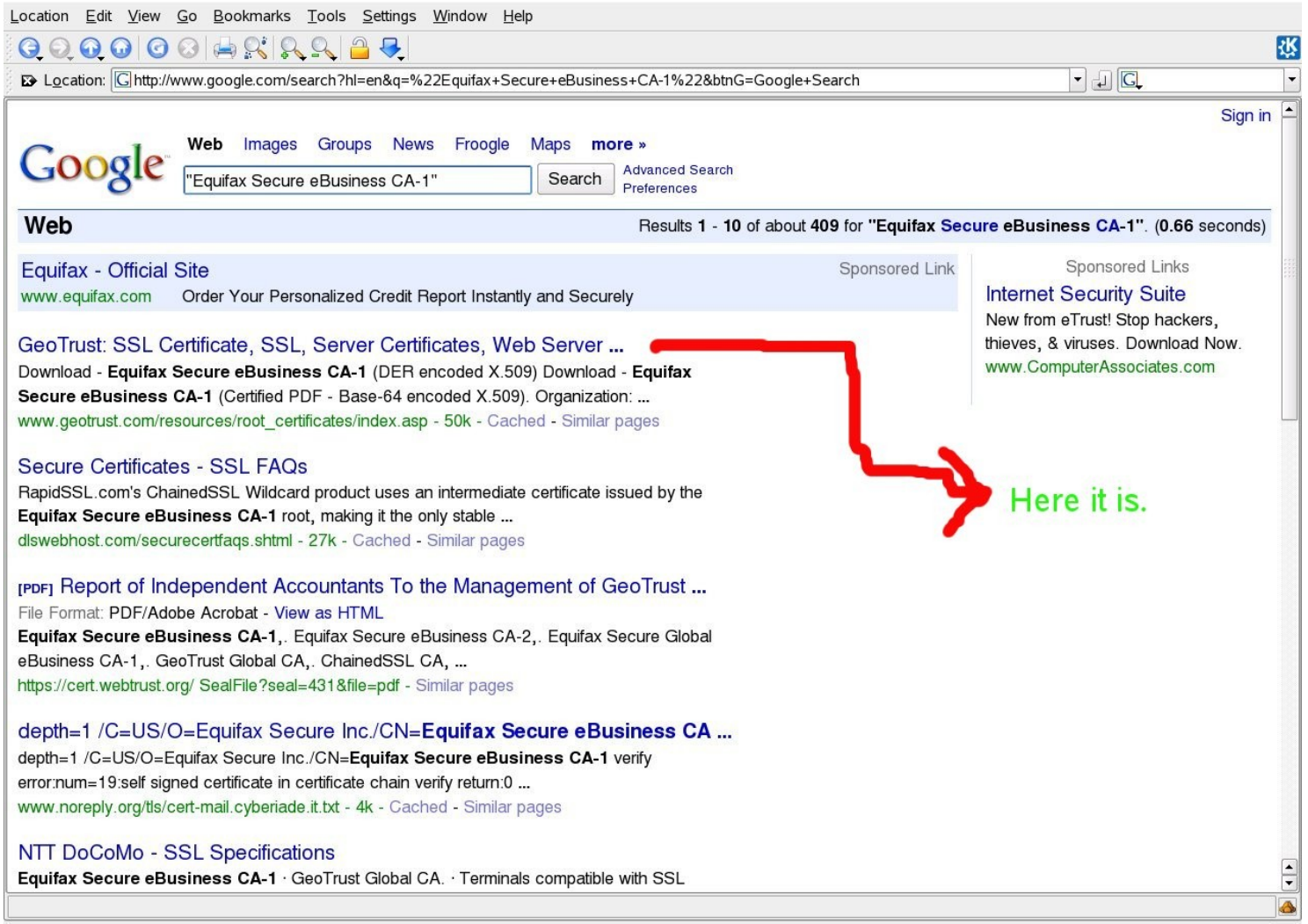
Step3: run fetchmail -v --ssl

- look for a line that looks like this:
- `fetchmail: Issuer CommonName: Equifax Secure eBusiness CA-1`
- Search the net for the CommonName.

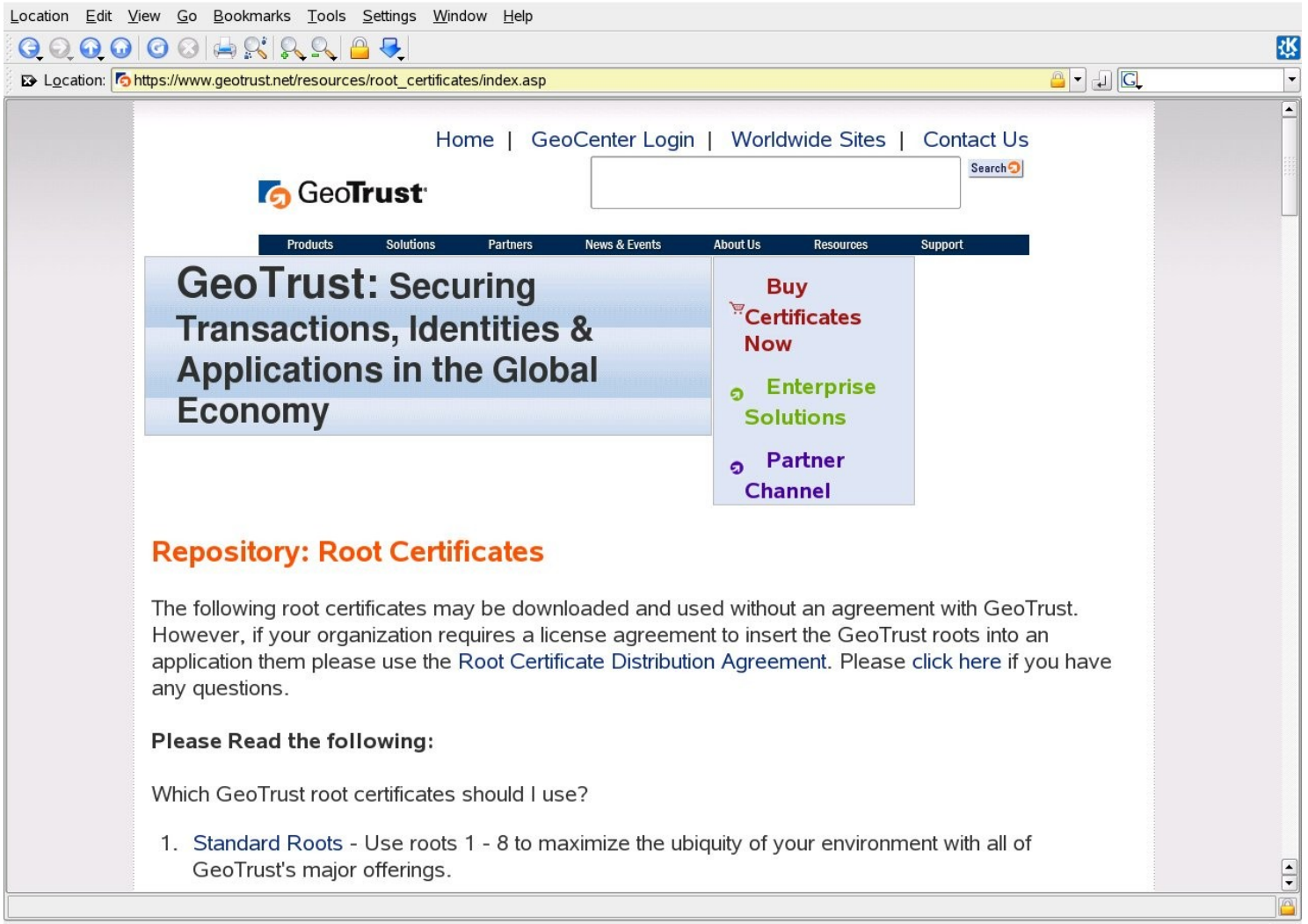
step4:Find the place to download the certificate.



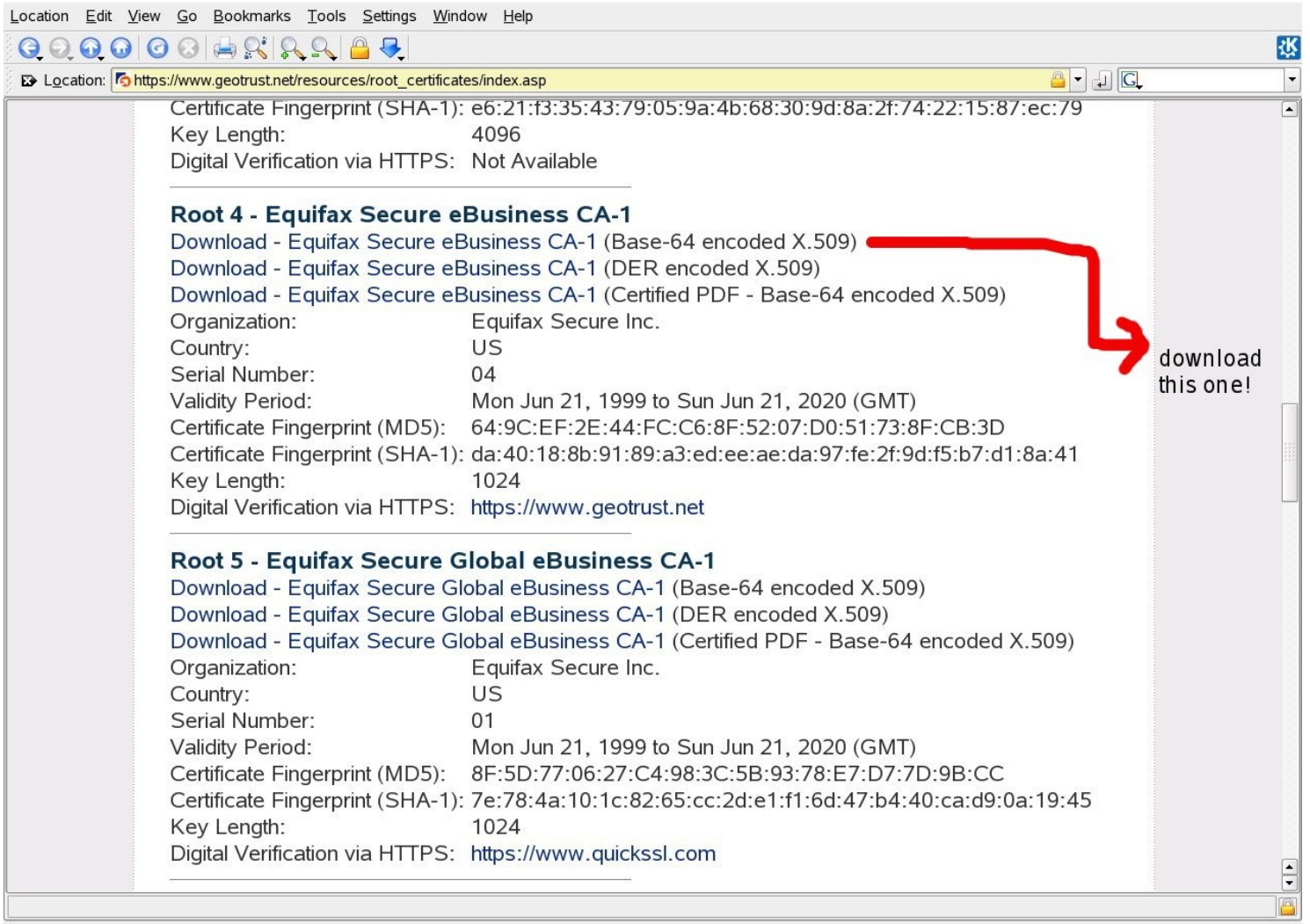
We have found it!



We have found it! Check that it is the true respectable site.



step5: Download the base64 version of the certificate.



step 6:Find the place openssl keeps all its certificates!

- There will be a lot of files of type “.pem” there.
- On my system it is: /etc/ssl/certs
- logon to root, copy the certificate you downloaded there, renaming the type to “.pem”.
- Then do a “c_rehash” on the directory. Example:
- # c_rehash /etc/ssl/certs

step7: fetchmail -v --ssl again!

- This time check that the error messages we encountered before about: `Server certificate verification error:` has gone away!
- OK, good we have verified that ssl works!

Step8: edit .fetchmailrc

- add “ssl” and “sslcertck” to the user line.
- Mine looks like this:
- `set postmaster "pelliott"`
- `set bouncemail`
- `set no spambounce`
- `set properties ""`
- `poll mail.io.com with proto IMAP`
- `user 'pelliott' there with password 'XXXXXX' is \ 'pelliott' here ssl sslcertck`

Success.

- From now on whenever we use fetchmail, it will use openssl to encrypt our password and protect us from man-in-the-middle attacks!
- Done.

Do you know 3 or 4 Miscellaneous Linux Tricks?

- Perhaps you can compile your favorite Linux Tricks into a presentation for ALG?
- ALG can use some internally generated presentations.