

# Linux on Compaq EVO N115 laptop

Paolo Subiaco - [psubiaco@creasol.it](mailto:psubiaco@creasol.it)  
<http://www.creasol.it/>

13 luglio 2002

## Sommario

This document illustrates the Linux installation procedure on the Compaq laptop EVO N115 (the same information is right for other Compaq laptops with AMD Duron cpu and VIA chipset). I've test the information below on a Mandrake Linux 8.2, but i suppose that other distributions are similar (just change some program path in the scripts below).

Knowledge about kernel compilation and bash script development is required!

## 1 Introduction

Linux installation on notebooks is not so easy as desktop computers because the relevant of certain aspects as like as power management and software suspend.

Using this document you can install Linux on your Compaq laptop enabling power management (through ACPI) and software suspend/resume; also your sound card will work, but you need to make several patch to your 2.4.18 linux kernel. Please consider that this information may be out-of-date soon, due to the fast patch available for the linux kernel!

Finally, i would thank *Juliusz Chroboczek* for his very useful document about linux installing on Compaq Presario 711EA, that you will find at the URL <http://www.pps.jussieu.fr/~jch/software/presario/>

## 2 Kernel patching

- Download a clean (vanilla) 2.4.18 kernel from <http://www.kernel.org> and uncompress it to /usr/src: the /usr/src/linux directory will be created.
- Download the *acpi-20020709-2.4.18.diff.gz* patch from <http://sf.net/projects/acpi/>
- Download the *kacpid.patch* from <http://www.pps.jussieu.fr/~jch/software/presario/kacpid.patch>
- Download *patch-vanilla-2.4.18-swsusp9.gz* from [http://sourceforge.net/project/showfiles.php?group\\_id=25964](http://sourceforge.net/project/showfiles.php?group_id=25964)
- Download *ad1886-presario.patch* from <http://www.pps.jussieu.fr/~jch/software/presario/ad1886-presario.patch>

- Apply all these patches:  
cd linux  
zcat ../acpi-20020709-2.4.18.diff.gz | patch -p1  
patch -p1 <../kacpid.patch  
zcat ../patch-vanilla-2.4.18-swsusp9.gz | patch -p1  
patch -p1 ../ad1889-presario.patch  
Don't worry if swsusp9 patch return one error (a file is missed due to the previous acpi patch, which delete some files used by swsusp9).
- Configure kernel appropriately: you must chose Athlon CPU, disable SMP and Local APIC, disable Advance Power Management and select all feature of ACPI.  
Select in the File systems menu, Kernel automounter version4, Ext3 journalling, DOS FAT, MSDOS, VFAT, /dev file system support and Network File Systems -> SMB  
Select Sound card support (M), VIA 82c686 Audio Codec (M), VIA 82C686 MIDI, OSS sound modules (M).  
In the Kernel hacking menu select Software suspend.
- Now is time to compile kernel, with the *make dep && make bzImage && make modules && make modules\_install* command, than *cp arch/i386/boot/bzImage /boot/bzImage-2.4* : if you got an error in kernel\_stat.h, modify that file adding the line  
#define smp\_num\_cpus 1
- Add the new kernel two times to the lilo.conf file; note that my root partition is /dev/hda6 and my swap partition /dev/hda5 : the latter is the place where swsusp will save computer configuration before suspend. The first lilo instance, linux-resume, will load configuration from previous suspend, instead of linux-noresume which will load linux normally. When lilo.conf will be modified, call *lilo* to update boot sectors with the new lilo configuration, and then restart computer.

```
default=linux-resume
image=/boot/bzImage-2.4
    label=linux-resume
    root=/dev/hda6
    append="devfs=mount resume=/dev/hda5"
    read-only
image=/boot/bzImage-2.4
    label=linux-noresume
    root=/dev/hda6
    append="devfs=mount resume=/dev/hda5 noresume"
    read-only
```

### 3 Software suspend setup

These are little problem involving software suspend:

- the ethernet interface must be down, as like as sound module: these module must be reload at the resume

- the system clock must be synced with the hardware clock, at the resume
- the hard disk write buffer must be disable during suspend and re-enable at the resume

I have write a little shell script to to these functions, which must be called to suspend notebook with root privileges: due to this reason, you must install *sudo* program and setup */etc/sudoers* file adding the following line:

```
ALL                ALL=NOPASSWD: /usr/local/bin/suspend.sh
```

Then create the file */usr/local/bin/suspend.sh* with the following lines:

```
#!/bin/bash

warn()
{
    echo "$@" >&2
}

die()
{
    echo "$@" >&2
    exit 1;
}

oldconsole=`fgconsole`
/etc/init.d/sound stop
/etc/init.d/network stop
ifconfig eth0 down

/sbin/hdparm -q -W 0 /dev/hda || die "Couldn't disable write cache"
#chvt 1 || die "Couldn't switch VTs"
#Change line to '1 0 2' for debugging purpose!
#echo "1 0 2" > /proc/sys/kernel/swsusp
echo "1 0 0" > /proc/sys/kernel/swsusp

sleep 10
echo "Resuming previous configuration..."
chvt $oldconsole
/sbin/hdparm -q -W 1 /dev/hda || warn "Couldn't reenable write cache"

/etc/init.d/network start
/etc/init.d/sound start
/sbin/hwclock --hctosys
```

Remember to add execution attribute to */usr/local/bin/suspend.sh* ! Also control the path of each command/script called by the *suspend.sh* script!

Now you can invoke *sudo /usr/local/bin/suspend.sh* to try notebook suspend: you need about 10 seconds to suspend notebook!

To resume, just press the power button: lilo will load the kernel, then it will resume previous configuration (with all previous open applications) in about 40 seconds.

## 4 Tips&tricks for WindowMaker users

What is WindowMaker? Is a X11 window manager. Other window managers are KDE, GNOME, FVWM,TWM, SawFish, ....

Why this chapter? Because i love WindowManager, which permit me to do any configuration i want. I know that the most of you using KDE or GNOME, which are nicier and more integrated with their applications.

Although, i prefer Window Maker because it require very few memory, and i can setup for each application where the application must be opened (which desktop), how the window must be designed, which icons display in each desktop, and so on....

Using WindowMaker with a laptop, you need an application which display the current battery state, so you must install an application which monitor battery status. Which kind of application? In other window manager there exists applications and applet applied to the panel; in WindowMaker there is a solution similar to the latter, its name is "dock".

Docks are icons which display something and can implement several controls: for the battery monitoring purpose i have installed *wmacpi*, but you can find lot of useful and nice docks at the URL <http://www.bensinclair.com/dockapp/showstyle.php3?show=name>

## 5 Automounter configuration

Very often the notebook is used to access SMB shares (i.e. windows shares) in a LAN: for this purpose i think that the best solution consist to setup linux automounter to mount/umount shares when required: this way you can access remote shares just selecting file `/mnt/servername/sharename` and the corresponding share will be mounted: after 60 seconds (or similar timeout) which the mounted share is not used, the share will be automatically unmount; this way you reach a good reliability accessing remote filesystem.

How to do this?

Fist, verify to have compile `automount_4` and `smbfs` in the kernel, then you must install the `autofs` package: `urpmi autofs` and configure several files about automounter daemon. Here is an example of configuration files, supposing to have a lan like this:

- server1
  - share1
  - share2
  - share3
  
- server2
  - share1
  - share2

Create the directories corresponding with the two server names:

```
mkdir /mnt/server1 /mnt/server2
```

Then edit `/etc/auto.master` file leaving uncommented the two lines below:

## 5 AUTOMOUNTER CONFIGURATION

---

```
/mnt/server1 /etc/auto.server1 --timeout=60
/mnt/server2 /etc/auto.server2 --timeout=60
```

Now create the file */etc/auto.server1* within the following 3 lines (one of each share):

```
share1 -fstype=smbfs,username=paolo,password=AAA ://server1/share1
share2 -fstype=smbfs,username=paolo,password=AAA ://server1/share2
share3 -fstype=smbfs,username=sysadm,password=secretpass ://server1/share3
```

Finally, create the file */etc/auto.server2* with the following 2 lines:

```
share1 -fstype=smbfs,username=pippo,password=pluto ://server1/share1
share1 -fstype=smbfs,username=paperino ://server1/share1
```

Now is time to try the automounter daemon: type */etc/init.d/autofs restart* then *cd /mnt/server1*

Opening this directory, if you type *ls -l* you will find nothing, because there are no shares mounted. Just type *cd share1* and then *ls -l* to see the content of *//server1/share1* ! If you leave this directory, after 60 seconds it will be unmounted automatically. Now enjoy the unix operating system, which permit you to see all shares/partions in one filesystem!