Linux Software Installation Notes
Prada Demo 2002

Version: Mandrake Linux 8.2/Nvidia OpenGL/Firewire

Author: Andreas Schiffler
Last Updated: 22 Sept 2002

Hardware

Design Goals

- Standard PC architecture
- Linux compatible hardware
- High CPU processing power
- Good OpenGL processing power
- DV input
- Analog Video input

Design Choices

- P4 or Athlon motherboard/CPU combinations (>=1.2 GHz speed)
- Extensive RDRAM or DDR memory (>=512 MB)
- Nvidia Graphics adapter (i.e. Geforce4 Ti 4600)
- Large CPU fan attached with premium thermal paste.
- Single ATA100/133 IDE 7.2krpm harddisks for software storage.
- Bt8x8 based analog video input card (i.e. Haupauge WinTV go)
- Supported PCI 1394 input card (i.e. ATI DV-Wonder)
- Aluminum case with extra intake and exhaust fans (recommended).
- Large 450W power supply (recommended).

Software

Operating System

The operating system used is *Linux*. Linux has the benefit to be easily available (downloadable from the internet), very cost effective (zero additional cost for any number of installations), secure (peer reviewed source code and security fixes are available fast) and well suited for the Prada software.

For the installation, the *Mandrake Linux*distribution is recommended, as it is currently (Aug 2002) the most widely used distribution, feature rich and provides most of the required software components as part of the standard installation. It is also the only distribution which is compiled with full optimization for current PC based hardware.

(Note: Other Linux distributions such as RedHat or Suse can be used as well – the outlined installation details have to be adjusted for the different system layout by a system administrator with Linux knowledge.)

Typically during the installation, the distribution will be customized with the latest software, additional software packages are installed via the distribution media or the internet.

References

A. Mandrake Linux: http://www.linux-mandrake.com
B. Linux Kernel: http://www.kernel.org

Application Software

The final application is comprised of several freely available modules and software packages available from the Internet plus a custom programmed component "Prada" that comes with a large repository of custom 3D model data and textures.

Additional software that needs to be installed is for example:
- OpenGL drivers
- Artoolkit libraries
- OpenVRML library
- Performer library
- DV library

and other packages.

Linux Installation

Choose Linux Distribution

Below we describe the necessary installation steps for Mandrake Linux 8.2 (or higher) – an easy to install, freely available version of Linux that is based on RedHat Linux.

All steps described can be performed in a similar fashion using other Linux distributions such as:
- RedHat Linux 7.3 (or higher)
- Suse Linux 8.0 (or higher)

In that case, references to RPM packages or configuration file contents and locations might differ slightly but the basic installation procedure is the same.

Basic Installation of Mandrake Linux 8.2

Initially we perform a basic Mandrake-Linux 8.2 installation (3 CDs for download edition, 5 CDs for commercial edition) with the following options:
- partition disk with one large / (root) partition and one large swap partition
- use a journaling filesystem (ReiserFS or Ext3) for the partitions
- choose to install all packages (approx 1.5Gbytes)
- choose medium security or the default (can be changed later)
- do not modify services at this time (the final services setup will be done later)
- create recovery bootdisk if desired

Please refer to the Mandrake installation manual for information on configuration and settings that can be done during install.

The installation takes approximately 15-45min depending on the speed of the hardware.

Customization of Mandrake Linux

After the installation, a whole range of custom steps and updates are performed to configure the operating system and software packages for use as a PRADA display engine.

(Note: Perform installation/update steps while logged in as "root".)

Local Package Archive (Recommended)

Mandrake Linux software is organized in packages with the file extension .rpm for easy management and updates.

Copy all RPMs from the CDs into a directory onto the harddisk. This allows for easy updates, faster system installation or remote recovery without the need for CD-ROMs.

```
cd; mkdir Mandrake82
```

Repeat the following with all CD-ROMs that you have:

```
mount /mnt/cdrom
cp -v /mnt/cdrom/Mandrake/RPMS*/* Mandrake82
umount /mnt/cdrom
eject
```

One can now install RPMs quickly from this directory using the command:

```
rpm -Uvh filename
```

Should the depedency check require additional packages, one can include as many package filenames as required to satisfy the installation. Additional useful command line parameters are `--nodeps` and `--force` which can be used together to force an installation.

It is also recommended to update the Mandrake Package Manager available via the Mandrake Control Center for this local RPM archive. Remove the CD1 to CDx from the source list. Add the local `/root/Mandrake82` directory to the source list.

Kernel Upgrade (Optional)

To get the latest performance improvements and hardware support for the operating system, a kernel upgrade to the latest stable version can be performed as deemed necessary.

The latest kernel image is available on http://www.kernel.org - follow standard installation manuals or the steps outlined below to recompile and install a new kernel (i.e. here for kernel version 2.4.18 upgrading from kernel version 2.4.8).

```
cd /usr/src; rm linux
tar xvyf linux-2.4.18.tar.bz2
mv linux linux-2.4.18; ln -s linux-2.4.18 linux
cd /usr/include
rm -rf linux asm asm-generic
ln -s /usr/src/linux/include/linux linux
ln -s /usr/src/linux/include/asm asm
ln -s /usr/src/linux/include/asm-generic asm-generic
cd /usr/src/linux
cp ../linux-2.4.8/.config .
make oldconfig
make dep clean bzImage modules modules_install
cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.18
cp System.map /boot/System.map-2.4.18
cd /boot
mkinitrd initrd-2.4.18.img 2.4.18
```

Edit `/etc/lilo.conf` to include the new kernel by copying existing entry and editing with the new kernel image and a new name.Then install a new bootloader configuration

```
lilo
```

If this step works without error and lists the new kernel entry name, reboot with the new kernel.

Desktop Upgrade and Red-Carpet Installation (Recommended)

To get acces to the Ximian Red-Carpet software update services the Ximian gnome desktop is installed. Red-Carpet allows for easy installation and updates of RPM packages with a graphical interface.

Launch the installer program while running Xwindows using this command line:

```
lynx -source http://go-gnome.com/ | sh
```

and follow directions.

After the installation is complete, restart Xwindows, login as user with a “Gnome” session selected. Choose to install Ximian’s desktop preferences.

Please refer the following URL for a subscription based, premium update service based on red-carpet:
http://www.ximian.com/products/ximian_red_carpet/express/

Security Updates (Recommended)

Update all RPM packages with latest security fixes.

When using using Helix Gnome, use the `red-carpet` application (commandline or Menu→System→Get Software …). Subscribe to the Mandrake 8.2 channel.

Otherwise use the Mandrake package manager (Desktop→Mandrake Control Center) to upgrade all packages. Select a server to check for security updates or use the wizard.

Repeat once a week on a running, internet connected system to stay up-to-date and apply security fixes.

Installation of Additional Software Packages

Xwindows Server Update (Required)

Upgrade the Xwindows server with drivers for Nvidia cards for optimal 2D and 3D performance.

Obtain driver RPMs from http://www.nvidia.com and compile kernel driver from source using:

```
rpm --rebuild NVIDIA_kernel*.rpm
```

This has to be done after the kernel has been updates and the system has been rebooted since the module is kernel specific. Switch to text mode to continue the Xwindows server software installation and configuration.

```
telinit 3
```

Install the Mesa, NVIDIA_kernel and NVIDIA_GLX packages.

```
rpm -Uvh -force Mesa-* libMesa*
rpm -Uvh NVIDIA_kernel*
rpm -Uvh NVIDIA_GLX*
```

Now configure the file */etc/X11/XF86config-4* as described in the *README* document available on the NVIDIA site.

"nv" → "nvidia"
add `Load "glx"` to modules section
choose default color depth of 24bpp

Test the X server installation by starting the server manually (can be exited using Ctrl+Alt+Backspace).

```
X
```

If this works fine, return to graphics mode.

```
telinit 5
```

Server Services Configuration (Recommended)

Configure services that are running on the server using the *Mandrake Control Center*:

```
drakxservices
```

and configure to start the following services 'On boot':

anacron, atd, bastille-firewall, crond, harddrake, internet, iptables, keytable, kheader, kudzu, network, numlock, postfix, random, sshd, syslog, usb, xfs

and any other services as required.

IDE Disk Optimization (Optional)

Install the **hdparm** package from the local RPM archive or using *red-carpet*.

```
rpm -Uvh hdparm*
```

Add the following lines to */etc/rc.d/rc.local* to optimize IDE disk access (adjust device names for your disk setup):

```
echo "Tuning harddisks ..."
/sbin/hdparm -c1 -A1 -m16 -d1 /dev/hda
```

Editor (Optional)

For a quick and easy to use editor from the commandline, install *joe*.

```
rpm -Uvh joe*
```

Edit */etc/joe/joerc* to taste.

OpenVRML (Required)

To install openVRML from the current source package (0.12.3) several links have to be manually adjusted.

```
cd /usr/X11R6
rm libGL.la libGL.so.1
ln -s /usr/lib/libGL.so libGL.so.1
ln -s /usr/lib/libGL.so libGL.so.1.0
ln -s /usr/lib/libGL.so libGL.so.1.2
cd /usr/lib
ln -s /usr/X11R6/lib/libGLU.so libGLU.so
ldconfig
```

And edit /usr/X11R6/lib/libGLU.la to remove reference to libGL.la.

Now unpack the openVRML source archive and configure with

```
./configure -without-spidermonkey -without-jdk -without-doxygen
```

then compile and install

```
make
make install
```

Performer (Required)

Install the performer RPM packages using:

```
rpm -Uvh performer*
```

Firewire (Required)

To get firewire device support for development several components need to be installed:

```
rpm  -Uvh libraw*1394* dvgrab*
```

Now run the firewire.sh script:

```
./firewire.sh -refresh
```

Get and compile libdc1394:

```
./configure
   make
make install
```

and add to the /etc/ld.so.conf file the path /usr/local/lib. The run

```
ldconfig
```

Check firewire availability and devices with the tool *gscanbus* (separate installation).

firewire.sh (Sample)

```
#!/bin/sh
if [ "$*" == "-install" ]; then
 echo "Installing ..."
 UNINSTALL=0
 INSTALL=1
 LIST=1
fi
if [ "$*" == "-uninstall" ]; then
 echo "Uninstalling ..."
 UNINSTALL=1
 INSTALL=0
 LIST=1
fi
if [ "$*" == "-refresh" ]; then
 echo "Refreshing ..."
 UNINSTALL=1
 INSTALL=1
 LIST=1
fi
if [ "$*" == "-list" ]; then
 echo "Listing ..."
 UNINSTALL=0
 INSTALL=0
 LIST=1
fi
if [ "$LIST" == "" ]; then
 echo "Commands available:"
 echo " -install"
 echo " -uninstall"
 echo " -refresh"
 echo " -list"
 exit
fi
if [ $UNINSTALL == 1 ]; then
 echo "Uninstalling drivers ..."
 /sbin/rmmod video1394 2>/dev/null
 /sbin/rmmod ohci1394 2>/dev/null
 /sbin/rmmod raw1394 2>/dev/null
 /sbin/rmmod ieee1394 2>/dev/null
```

```
 echo "Done."
 echo
fi
if [ $INSTALL == 1 ]; then
 echo "Installing drivers ..."
 /sbin/insmod ieee1394
 /sbin/insmod raw1394
 /sbin/insmod ohci1394 attempt_root=1
 /sbin/insmod video1394
 echo
fi
if [ $LIST == 1 ]; then
 echo "Kernel modules loaded:"
 DRIVERS=`/sbin/lsmod | grep 1394`
 if [ "$DRIVERS" == "" ]; then
  echo "None."
 else
  /sbin/lsmod | grep 1394
 fi
 echo
fi
```

Coriander (Recommended)

To view firewire camera feeds and check performance and quality of DV input, the coriander tool program is useful. To compile and install it follow these steps:

```
rpm -Uvh libSDL*-devel*
rpm -Uvh libalsa*-devel*
```

In the coriander source directory:

```
./autogen.sh
```

Edit line 89 of src/thread_iso.c and add ",1" to function parameters before video_device.

```
make
make install
```

Test by running

```
coriander
```

DMA setup and proper device setup is important. Choose "File Preferences"→ "Receive" → "Method" = *Video1394* and "Device" = */dev/video1394/0*

References

A. NVIDIA Linux Drivers - http://www.nvidia.com/view.asp?IO=linux_display_archive
B. OpenVRML - http://www.openvrml.org
C. Libdv - http://libdv.sourceforge.net
D. Coriander - http://www.tele.ucl.ac.be/PEOPLE/DOUXCHAMPS/ieee1394/coriander
E. Performer - http://www.sgi.com/software/performer