QUETZAL–A LIVE DVD/CD SYSTEM BASED ON OPENBSD

MICHO ĐURĐEVICH

Abstract. This is the companion article to Quetzal–which is a live DVD/CD based on OpenBSD operating system. We explain basic features of the system, a DHCP/PXE remote-booting setup, and provide more detailed instructions for a manual hard disk install. We also discuss simple modifications to the OpenBSD kernel used in Quetzal, and some tricks, that might be interesting on their own.

Contents

The word 'Quetzal' is intertwined with roots of ancient Mexican and Central American civilizations. In Náhuatl *Quetzalli* means a 'rich green feather'. Quetzal is a beautiful and extraordinary bird, with green and red feathers. The male has long green tail coverts, and the female is more 'elliptic'. A fundamental god of ancient Mexican and Central American cultures is *Quetzalcoatl* a creature unifying the bird and Serpent (*Coatl*). A quantum superposition of Heaven and Earth...

1. Introduction

Here we shall deal with another Quetzal–a live DVD/CD system, based on OpenBSD. This article* can be viewed as a user manual for Quetzal. The system contains a full OpenBSD installation, and a collection of selected applications. Our window manager of choice is WindowMaker. Its geometrical simplicity, modularity, performance, stability and internal architecture, match nicely with the philosophy of OpenBSD and UNIX in general. Desktop images, themes and icons come from various sources [1]. A fine live CD called OliveBSD [2] was an initial inspiration for this work. We refer to [4], for a general introduction to OpenBSD operating system.

Here is a list of principal packages included with Quetzal:

- Symbolic Mathematics Package Maxima + wxMaxima Frontend;
- GNU Image Manipulation Program–GIMP + Full Help/Docs;
- Scripting Language and Graphical Toolkit TCL/TK;
- Object Oriented Scripting Language Python;
- Graphical Toolkit wxWidgets + wxPython;
- Common Unix Printing System–CUPS;
- Internet Messaging System Gaim;
- Complete TEX distribution;
- Text Editor SciTE;
- Remote Desktop;
- PDF Viewer Xpdf;
- File Manager Rox;
- OpenOffice.org Suite;
- Thunderbird Mail Client;
- Firefox Internet Navigator;
- Audio Player XMMS + MP3 Plugin;
- GhostScript & GV Postscript Viewer;
- Dockable Clock, Moon Clock, & Audio Mixer.

This document focuses on DVD releases of Quetzal. Modulo simple modifications, almost everything applies to CD releases as well.

2. First Steps

The first step is to download the corresponding bz2-compressed image. The Quetzal DVD can be created by burning the image, with the help of virtually any DVD burning software. For this to work, of course, we need a good DVD-ROM burner :)

On OpenBSD, the whole procedure would look something like this:

```
bzip2 -d quetzal-dd.mm.yyyy.iso.bz2
```

At this point it would be wise to verify the checksum for the iso-file. For example

```
md5/sha1 quetzal-dd.mm.yyyy.iso
```

To actually burn the image to a DVD-ROM,

```
growisofs -dvd-compat -Z /dev/rcd0c=quetzal-dd.mm.yyyy.iso
```

*Version 3.6.2006

The system boots to text-mode interface. The only real user defined is 'root' and the password is 'quetzal'. The system requires some configuration of virtually all OpenBSD components, including network and X windows. This is done intentionally. Quetzal is primarily intended as an educational tool, so all the major configuration options are left open. In such a way we also stay compatible with the spirit of UNIX.

Quetzal is easily installable on a local hard disk. We do not provide any installation wizard—the installation is a collection of simple manually executable steps.

In order to start the graphical interface using the X autoconfiguration, we simply execute `xinit`. This should work fine in most of the cases. An alternative is to manually configure the X server, by invoking a standard sequence of commands:

```
X -configure
cp /root/xorg.conf.new /etc/X11/xorg.conf
vi /etc/X11/xorg.conf
```

In this case, after editing `xorg.conf` and before initializing X via `xinit`, it is necessary to *delete* the file `/tmp/.X11-unix/X0`.

3. About OpenOffice

The OpenOffice suite included with Quetzal is a Linux application. Therefore, we need to activate the Linux emulation:

```
sysctl -w kern.emul.linux=1
mount -o rw,linux -t procfs /proc /proc
```

Under 'normal' circumstances, this should be sufficient to run most of Linux applications.[†] However, in order to run OpenOffice from a read-only medium as the Quetzal DVD, it is necessary to apply an extra trick. If we try to run it directly from the disc, the program would crash with an error message. The origin of the problem lies in an inappropriate handling of lock file requests by the filesystem driver. Some applications require the locking functionality in order to work properly, and OpenOffice is one of them.

A solution we present here is to simply mount via NFS, the local directory `opt` containing OpenOffice distribution. The NFS does handle lock requests properly. A future version of Quetzal will include a modified cd9660 filesystem driver, and the trick will be unnecessary. At first, we start the NFS server:

```
portmap; nfsd -t -u -n 4
mountd
```

And then we mount the NFS share defined in `/etc/exports` file:

```
mount localhost:/opt /oo2
```

4. Remote Booting

In this section we shall present a simple DHCP/PXE remote-booting setup, suitable for booting diskless clients over a local area network. These clients would mount their root filesystem via NFS, and so it is necessary to compile the appropriate custom kernel:

```
config bsd root on nfs swap on nfs
```

[†]Preliminary releases of Quetzal included Linux Opera & Flash Plugin. They had to be removed from the final version due to inappropriate licensing conditions.

Such a kernel is included as `/quetzal/bsd.nfs`. On the server (and we assume it is an OpenBSD system) we have to set up various daemons. At first, it is the TFTP service. The simplest is to enable it in `/etc/inetd.conf`. The default directory `/tftpboot` should contain at least the loader `pxeboot` and the above mentioned kernel renamed to `bsd`. Second, we have to configure the DHCP server, by editing `/etc/dhcpd.conf`. Here is a simple configuration:

```
option domain-name "q-systems.mexico";
option routers 192.168.1.1;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option domain-name-servers 192.168.1.2;
server-name "DHCPserver";
server-identifier 192.168.1.2;

default-lease-time 300;
max-lease-time 300;

subnet 192.168.1.0 netmask 255.255.255.0 {
       filename "pxeboot";
       range 192.168.1.7 192.168.1.117;
   }
```

This ensures the initial client-server communication via PXE, which would load via TFTP /tt pxeboot into the client memory and execute it. The loader would, in its turn, get the kernel via TFTP into the client memory and pass control to it. At the end of kernel processing, it tries to mount the root filesystem via NFS. In order for this to work succesfully, besides properly configured NFS server, we need to configure two more daemons: `rarpd` for reverse address resolution and `rpc.bootparamd` for supplying root filesystem and swap information to the client. The reverse address resolution is controlled by `/etc/ethers`. This file translates network adapter MAC addresses into hostnames. A single client setup would look like

```
00:50:DA:1A:14:A3      qbird
```

where `qbird` is the diskless client name. The MAC address is taken from the actual client we played with (IBM Network Station 2800, equipped with a PCI 3Com 905C-TX-M adapter). The file `/etc/bootparams` controls the `rpc.bootparamd` service. It is divided in sections, corresponding to different remote-booting clients:

```
qbird   root=xochitl:/opt/quetzal \
        swap=xochitl:/opt/qbird/swap
```

Here `xochitl` is the hostname of the server. Both client names (as `qbird`) and the server name (`xochitl`) should be defined in `/etc/hosts`. The above configuration tells us where the root of the client NFS filesystem is, and the location of the virtual memory file. Assuming we already have created `/opt/quetzal` on the server, we will have to copy over the contents of the Quetzal DVD, say

```
tar cf - -C /mnt . | tar xpf - -C /opt/quetzal
```

as well as to create a nice initial collection of zeros for the swap

```
dd if=/dev/zero of=/opt/qbird/swap bs=8192 count=8192
```

The above would create a 64M swap file.

The whole remote-boot system is orchestrated by the following sequence of commands:

```
dhcpd; rarpd <ethn>
portmap
rpc.bootparamd
nfsd -t -u -n 4
mountd
```

A principal use of remote-booting is to create a simple yet efficient and highly manageable LAN consisting of thin clients (diskless machines). Another interesting use of remote-booting, is to facilitate local operating system installations. We can, for example, remote-boot Quetzal on a machine with a local hard disk, and install everything locally. Such a procedure is discussed in the next section. It is the same for DVD and remote-booted versions of Quetzal.

5. Installing Quetzal on Hard Disk

5.1. **Assumptions**

We shall assume here that Quetzal is being installed on the first IDE hard disk, corresponding to `wd0` in OpenBSD nomenclature. We shall also assume that the whole hard disk is used for Quetzal. Other configurations are treatable in a similar spirit.

5.2. **Partitioning the Disk**

At first we create the appropriate MBR, using `fdisk` in the interactive mode:

```
fdisk -e wd0
```

Within the editor, we execute the following 4 instructions:

```
reinit,update,write,quit
```

This would create a single `0xA6` partition covering the whole disk. The next step is to create the disk label. Fire up the OpenBSD label editor.

```
disklabel -E wd0
```

The OpenBSD partitions created will be {`wd0a,wd0b,wd0d,wd0e,wd0f`} and correspond to {`/root,swap,/usr,/var,/opt`} respectively.

5.3. **Creating File Systems**

After creating the partitions, we need to create a fresh FFS on them:

```
newfs /dev/rwd0a; newfs /dev/rwd0d
newfs /dev/rwd0e; newfs /dev/rwd0f
```

The disk is now ready to hold Quetzal files. In order to be able to copy the optical disc contents, we first need to mount the relevant partitions:

```
mount /dev/wd0a /mnt
mkdir /mnt/{usr,var,opt,dev,etc,proc,emul}
mount /dev/wd0d /mnt/usr
mount /dev/wd0e /mnt/var
mount /dev/wd0f /mnt/opt
```

5.4. **Copying Files**

This is a time-consuming step and runs a higher risk of failure, if the optical disc or DVD unit are low-quality. Here we shall use `tar` to copy across filesystems. For example,

```
tar cf - -C / usr opt sbin bin | tar xpf - -C /mnt
cd /etcq; tar cf - * | tar xpf - -C /mnt/etc
cd /varq; tar cf - * | tar xpf - -C /mnt/var
```

While the hdd-install `/root` folder will be simply unpacked:

```
tar xpf /quetzal/root.tar -C /mnt
```

Let us mention here that the `/quetzal` directory contains archived forms of `/root,/etc,/var` folders, the files `root.tar etc.tar` and `var.tar` respectively. These archives are unpacked during the system boot, along the memory filesystems creations (from the `rc` script). However directories `/varq` and `/etcq` are aimed for the hard disk installations. In order to save memory, diverse files from `/etc` and `/var` are simply symbolic links to the originals in `/etcq` and `/varq` respectively. A handful of important files are different between Quetzal and the hdd-install versions of these two directories. And many files need to be writable—hence the difference.

```
cp /quetzal/{bsd,boot}          /mnt
```

The above step is necessary because the kernel for Quetzal is compiled with the root filesystem pointing to `/dev/cd0a`. Such a kernel is not suitable for booting from a hard disk. The correct kernel is supplied in `/quetzal` directory. Also, it is convenient to have the loader at the root of the filesystem.

5.5. **Final Adjustments**

Let us proceed by making all the devices:

```
cp /quetzal/MAKEDEV  /mnt/dev
cd /mnt/dev;   sh MAKEDEV all
```

We have to create a number of links at the root of the filesystem:

```
ln -s /usr/local/emul/redhat /mnt/emul/linux
ln -s /opt                   /mnt/oo2
ln -s /var/tmp               /mnt/tmp
ln -s /opt/home              /mnt/home
```

The `oo2` link is to ensure that desktop objects for OpenOffice would work correctly ‘out of the box’. To complete our setup, let us install the first and the second stage boot loaders:

```
cd /usr/mdec; ./installboot -v /mnt/boot biosboot wd0
```

We should now be ready to boot from the hard disk. Execute `halt` remove the Quetzal disk from its unit, reboot and enjoy!

6. Concluding Remarks

6.1. Some Specific Changes to Code

A kernel function `inittodr` defined in `/sys/arch/i386/isa/clock.c` was modified, in order to avoid printing of a horrifying bogus error message about filesystem time discrepancy. The modified function, when called with `base=0` skips all comparisions between `base` and the system time.

Here is an example of a modified code:

```
if (base > 0 && base < zzzz) {
       printf("WARNING: blah blah"}
       base = xxxx
}
```

The kernel of OpenBSD was compiled with the appropriate root filesystem information

```
config bsd root on cd0a
```

The kernel messages are red on black. This is achieved by putting

```
WS_KERNEL_FG=WSCOL_RED
WS_KERNEL_BG=WSCOL_BLACK
```

in the kernel configuration.

The X server was modified, in order to remove CID folder from the default font folder search path.

The filemanager ROX was modified, in order to remove annoying remainders about running as root. The default folder icon was replaced by a more 'NeXT-ish' one. When searching for mount points, the program will only look at actually mounted filesystems, without any `fstab` involvement.

The window manager was modified, in order to replace the GNUstep info dialog by a Quetzal dialog. Furthermore, WindowMaker comes with custom-prepared 48-point icons and additional artwork [1], as already mentioned.

The package database `/var/db/pkg` does not fully reflect the above modifications to the packages.

6.2. Constructing ISO

Perhaps the most interesting question here is how to construct a bootable floppy used for the emulated boot. Let us assume we are at the root of the hdd-folder that holds all other Quetzal files.

```
dd if=/dev/zero of=bfloppy.fs bs=0x10000 count=45
```

This would create the file for the 2.88M floppy type (64k*45).

```
vnconfig  -c -v svnd0 bfloppy.fs
disklabel -w -r svnd0 floppy288
newfs -m 0 -o space -i 524288 /dev/rsvnd0a
mount /dev/svnd0a /mnt
```

The above would format the image as a 2.88M floppy. The disk label is taken from `disktab` and the filesystem is optimized for space, with minimum-free set to zero, and the inode density set to a minimal value (0x80000=524288).

```
cp /usr/mdec/boot /mnt
gzip -c bsd > /mnt/bsd
```

Here `bsd` is our custom prepared Quetzal kernel. We put it on the floppy image, in the compressed form. We complete the construction by installing the 1/2 boot loaders.

```
cd /usr/mdec; ./installboot -v /mnt/boot biosboot svnd0
umount /mnt
vnconfig -u svnd0
```

The bootable ISO image can be now created by `mkisofs`. For example, from the Quetzal parent directory:

```
mkisofs -b bfloppy.fs -c boot.catalog -l -R -V Quetzal \
-o dimages/quetzal.iso quetzal
```

References

[1] See `/quetzal/ARTWORK` for a complete listing of artwork sources. In particular, the wallpapers collection contains selected works by Brad Cover (*http://americanpsycho.deviantart.com*) as well as selected photographs by Stanko Kostic (*http://www.skostic.com*). In addition we included 3 iconsets from Barbara Kaemper collection: *http://www.barbarakaemper.de*. And 4 complete themes from the same website. All this artwork is included in Quetzal with kind permitions of the above mentioned authors. For default desktop icons we used a custom-prepared set of 'Gant' icons from Window Manager Icons collection: *http://wm-icons.sourceforge.net/*.

[2] OliveBSD Project: *http://g.paderni.free.fr/olivebsd/*.

[3] Kevin Lo: *Building an OpenBSD Live CD*, ONLamp.com article *http://www.onlamp.com/pub/a/bsd/2005/07/14/openbsd_live.html*.

[4] Michael W. Lucas: *Absolute OpenBSD: UNIX for the Practical Paranoid*, No Starch Press, San Francisco (2003).

Instituto de Matematicas, UNAM, Area de la Investigacion Cientifica, Circuito Exterior, Ciudad Universitaria, México DF, CP 04510, MEXICO / Q-Systems, MEXICO

E-mail address: `micho@matem.unam.mx`
`http://quetzal.matem.unam.mx`
`http://www.matem.unam.mx/~micho`