Kiosk HOWTO Gene Wilburn, ITS Dept, Royal Ontario

Table of Contents

Kiosk HOWTO

Gene Wilburn, ITS Dept, Royal Ontario Museum <genew@rom.on.ca>

v1.0, October 1999

This document provides a guide for setting up a WWW−based kiosk using Linux, X11R6, FVWM2, Netscape Navigator 4.X, and a customized trackball. It outlines the methods that were used to create a public kiosk for the Hands On Biodiversity Gallery at the Royal Ontario Museum, Toronto.

5.Other Considerations

6.Future Directions

1.Introduction

Information display kiosks are useful in libraries, galleries and museums, educational institutions, municipal offices, visitor information booths, conferences, shopping malls, airports−−−in short any location where simple, easy access to information is desirable. Kiosks are normally set up with touchscreens or pointing devices such as trackballs, to allow people to select and view information that is attractively displayed and up to date.

There are many ways to create kiosks, from expensive solutions based on proprietary software to HTML−based open−source solutions. Browser−based technologies are particularly attractive because they are inherently multimedia, offering text, graphics, sound and streaming media, and the content is highly portable.

Linux provides a flexible and interesting platform for kiosk development. Linux is inexpensive to set up and it offers a wide range of options, from diskless display stations to self−contained, database−driven web servers. Due to the versatility of the underlying operating system, a well−designed Linux kiosk can be placed in a remote location and administered via a telephone or network link.

This HOWTO explores one method of setting up Linux as a standalone information kiosk, using Netscape Navigator 4.X and FVWM2 on a Red Hat Linux 6.X system. It is based on a kiosk I set up for use in the *Hands−on Biodiversity Gallery* in the Royal Ontario Museum (http://www.rom.on.ca), Toronto, Canada. The kiosk outlined in this HOWTO incorporates a trackball rather than a more expensive touchscreen. There are undoubtedly many other ways to create a Linux−based kiosk, but this one has worked reliably for us and it may provide a useful starting point for your own kiosk project.

1.1 Copyright, license and terms of usage

1.2 Acknowledgements

Our Linux kiosk derives heavily from the work done by the City of Charlotte, North Carolina, for their municipal website, *Charlotte's Web* (see http://www.charweb.org/webinfo/kiosk/). The Charlotte's Web kiosk project, which employs a touchscreen monitor, was set up in 1996 using an early version of Slackware Linux, Netscape Navigator 2.X, and FVWM. The webmasters at Charlotte have created a very useable kiosk implementation and have shared their complete setup via their website. This HOWTO, to a certain degree, represents an update of their work, altering the details to work with Red Hat 6.X, Netscape Navigator 4.X, FVWM2, and a custom−made trackball.

I worked with Debra Luneau (debral@rom.on.ca), the Royal Ontario Museum webmaster, to create the graphics overlays for Netscape. The examples we distribute at our ftp site are her work−−−and are the ones we use on the kiosk in the Biodiversity Gallery.

The Ontario Biodiversity section of the ROM website was designed by Hopscotch Interactive (www.hopscotch.ca). The original application was converted from a flat−file database system to a MySQL−database driven system by Gord Howells (gordonh@rom.on.ca), the ROM's database designer and programmer. The non−kiosk version of this mini−site may be viewed at http://www.rom.on.ca/ontario/.

2.Overview

Setting up Linux for use as a web−based kiosk is similar to setting up Linux as a graphical workstaion, with a few extra considerations. If the machine is to be a standalone unit, it must boot straight into kiosk mode. Furthermore, the browser environment must be modified in to disable standard dialog screens and to eliminate parts of the browser that allow inappropriate interaction, such as the Location Bar in Netscape.

Essentially a kiosk website is self−contained. Users navigate through the site via clicks with a pointing device. The pointing device can be a touchscreen, a trackball, or a mouse. Mice are difficult to secure in a public area.

A Linux kiosk can be connected back to a home site via a direct network or PPP link, requesting pages from the home site, or it can be a self−contained website, running Apache and any corresponding CGI modules.

3.Building a Kiosk Environment

3.1 FVWM2

FVWM2 (FVWM, version 2 –– www.fvwm.org) is a flexible, reliable, and lightweight X Window manager that provides excellent configurability for creating a kiosk environment. For kiosk use you must change some of its default settings. You need to modify the system.fvwm2rc file to remove things like virtual desktops, title bars, exit controls, etc, leaving a relatively minimalist FVWM environment. Because the kiosk is a standalone unit, modifying the system–wide settings does not present a problem.

You can also take advantage of FVWM's settable object properties to use X Window bitmaps (xpm files) to mask the parts of Netscape Navigator you don't want users to access.

The FVWM2 file you need to modify is /etc/X11/fvwm2/system.fvwm2rc. (If you use a different distribution than Red Hat, this file may be located elsewhere.) You may have to poke around in this file to find the sections listed below.

/etc/X11/fvwm2/system.fvwm2rc

In `/etc/X11/fvwm2/system.fvwm2rc` change

```
DeskTopSize 2X2
```

to

```
DeskTopSize 1X1
```

This eliminates the virtual desktops.

Next, in the Default Styles section, change

```
Style "*"           BorderWidth 7, HandleWidth 7
```

to

```
Style "*"           BorderWidth 0, HandleWidth 0, notitle, nohandles
```

This will disable most of the normal windowing controls.

Add the following two lines to the Default Styles section:

```
Style "xdaliclock"  StaysOnTop
Style "sxpm"        StaysOnTop
```

The Default Styles part of the file should look something like this when you finish:

```
# default Styles:
#Style "*"           BorderWidth 7, HandleWidth 7
Style "*"           BorderWidth 0, HandleWidth 0, notitle, nohandles
Style "*"           Icon unknown1.xpm, Color lightgrey/dimgrey
Style "*"           MWMFunctions, MWMDecor, HintOverride
Style "*"           DecorateTransient, NoPPosition
Style "*"           IconBox 0 -10 -280 -1
Style "*"           FocusFollowsMouse
Style "*"           RandomPlacement, SmartPlacement
Style "xdaliclock"  StaysOnTop
Style "sxpm"        StaysOnTop
```

You can use the xdaliclock and sxpm programs to cover up parts of Netscape Navigator. The ``StaysOnTop'' attribute is the active ingredient.

Next, find the following four lines and comment them out as shown:

```
#AddToFunc InitFunction         "I" Module FvwmButtons
#+                      "I" exec xsetroot -mod 2 2 -fg \#554055 -bg \#705070

#AddToFunc RestartFunction "I" Module FvwmButtons
#+                      "I" exec xsetroot -mod 2 2 -fg \#554055 -bg \#705070
```

Commenting out these lines will eliminate the nice task launcher from FVWM2, which is okay because you're after as simple an environment as you can create.

3.2 Netscape Navigator

Netscape Navigator is a better choice for setting up a kiosk than Netscape Communicator. Being simpler, there are fewer objects to hide.

Under Microsoft Windows, Netscape Navigator has a kiosk mode that eliminates most of the Navigator controls but there is, at the time of this writing, no built−in kiosk mode for the Unix version of Navigator.

The Unix version can, however, be invoked with a `−geometry` argument that allows us to always start Netscape full screen. This forms part of our strategy for a Netscape−based kiosk.

You need to decide on our kiosk screen resolution. Using a 19− or 20−inch monitor, we found 640x480 too large, 1024x768 too small, and 800x600 just about right for public viewing (you may decide differently). Once you decide on the resolution, you are ready to begin adjusting Navigator. For the rest of this HOWTO we assume 800x600 resolution. You'll need to adjust accordingly if you select a different resolution.

Note: Navigator is going to be invoked by root during bootup, so make all of the Netscape adjustments under the root account in the `/root/.netscape` directory.

Invoke Netscape and click into the `Edit, Preferences` dialog. Under `Appearance` check ``Show Toolbar as Text Only.'' Under `Appearance, Fonts` adjust both fixed− and variable−width fonts to at least 14pt or the screen display will be too small for comfortable reading.

Under `Preferences, Navigator` insert the URL of your kiosk home page. If you're running a local version of Apache or another web server, make this `http://localhost/` for top level access, or deeper into the html doctree if desired.

Save these options, then click `View` on the Navigator menu bar. Deselect `Personal Toolbar` and `Location Bar`.

Notice how much more kiosk−like the interface has become? When you exit Navigator, these settings will be saved.

3.3 Overlay Masks for Navigator

If you look closely at your adjusted Navigator screen, you'll see four areas that need to be covered and protected from mouse or trackball clicks:

1. The Toolbar at the top
2. The twister at the left−hand side of the navigation toolbar
3. The rest of the navbar to the right of Home
4. The security key in the lower left−hand corner of Navigator.

Here's a rough schematic of the areas you need to mask:

```
+-----------------------------------------+
|             # 1 Toolbar                 |
+--+------------+-------------------------+
|#2|            |  #3 Navbar              |
+--+            +-------------------------+
|                                         |
|                                         |
|                                         |
|                                         |
|                                         |
|                                         |
|                                         |
|                                         |
|                                         |
|                                         |
+--------+                                |
|#4 Key  |                                |
+-----------------------------------------+
```

To mask over the top areas (#1−3), we'll need three graphic images. To mask the security key (#4) we'll use xdaliclock.

Creating the Graphical Masks

The easiest way to create graphical masks to cover areas #1−3 is to grab a screenshot of Navigator in full 800x600 mode and then crop out areas that correspond to #1−3. I used GIMP for this.

You can then re−design the basic graphics, changing the colors if you wish, or putting your own logo on them. Save them as:

1. topbar.xpm
2. navleft.xpm
3. navright.xpm

The xpm format is the native X Window bitmap format. You will use the X Window program *sxpm* to place them on the screen.

Creating the masks with just the right number of pixels and just the right placement on the monitor requires some experimentation. For your convenience, I have placed the xpm files we created plus our configuration files in at the following web site: http://www.rom.on.ca/kiosk/HOWTO/. You may find it simpler to download our masks and use them as a starting point for your own graphics.

3.4 /root/kiosk.xinitrc

Just as you would normally create a .xinitrc file for your personal use, you must create an X Window session initialization file for our kiosk application. The following script implements a number of features:

- Screensaver initialization (optional)
- Placement of graphic bitmap over Navigator menu bar (topnav.xpm)
- Placement of graphic bitmap over navigation bar twiser (navleft.xpm)
- Placement of graphic bitmap over rest of navbar (navright.xpm)
- Xdaliclock placement over Navigator security key (adjusted for 800x600 screen resolution)
- Netscape startup and loop checks to make sure Navigator is running, and to re–run it if it isn't.

This is the main file that runs the kiosk.

```
#!/bin/sh
# kiosk.xinitrc -- Initialization script for kiosk X Window session

# Start screensaver
/usr/X11R6/bin/xscreensaver-command -exit
/usr/X11R6/bin/xscreensaver &

# Overlay over Navigator (mask #1)
sxpm /root/navtop.xpm &

# Overlay small graphic over Toolbar twister - left-hand side (mask #2)
sxpm -g +0+23 /root/navleft.xpm &

# Overlay larger graphic to cover everything to the right of
# Home on the Toolbar (mask #3)
sxpm -g +275+23 /root/navright.xpm &

# Place xdaliclock over top of Netscape security key, lower lhc (mask #4)
/usr/X11R6/bin/xdaliclock -g 38x20+0+578 -font fixed -noseconds \
-bg gray -fg black &

sleep 2

# Start fvwm2 window manager
fvwm2 &

# restart netscape if it ever exits
while true ; do
        if [ -f /.netscape/lock] ; then
                rm /.netscape/lock
        fi
        /usr/bin/netscape -geometry 800x600
done

# This should never be reached
```

3.5 /root/xkiosk

You can now start testing your kiosk setup. Since you will need to do this repeatedly, manually, as well as from an initialization script, create the following as a shell script named /root/xkiosk:

```
/usr/X11R6/bin/xinit /root/kiosk.xinitrc -- /usr/X11R6/bin/X \
-xf86config /root/kiosk.XF86Config bc
```

Because you will be testing often, and reusing this later when you automate the startup, create this command as a shell script called `/root/xkiosk`. Be sure to `chmod ug+x` it.

4.Initializing the Kiosk

Under normal (i.e. kiosk) conditions, Linux should boot straight into Kiosk mode. In the event of a power failure, accidental reboot, or scheduled reboot, you need to configure the system to do this.

For our kiosk we elected to use init level 4, which is normally unused, for kiosk mode. We made the following adjustment to /etc/inittab:

4.1 /etc/inittab modifications

```
# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - Kiosk mode
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:4:initdefault:
```

4.2 /etc/rc.d/rc.4/S99xkiosk runlevel script

To initialize the kiosk after everything else has been initialized, including, in our case, Apache and MySQL, create the file /etc/rc.d/rc.4/S99xkiosk:

```
#!/bin/bash
#
# S99xkiosk   Initialization of kiosk for runlevel 4
#
# Author:           Gene Wilburn <genew@rom.on.ca>
#

# Display message on console
echo "Starting up the X Window Kiosk ... "

if [ -f /.netscape/lock] ; then
   rm /.netscape/lock
fi

# Logging of X activity
echo %%%%%%%%%%%%Reboot%%%%%%%%%% >> /var/log/xlog
/root/xkiosk
```

```
# This point should only be reached by pressing Ctrl−Backspace
/sbin/shutdown −r now

# All done.
```

Once this file has been created and inittab has been adjusted to boot into init level 4, Linux will boot directly into kiosk mode. On our kiosk we disable telnet and ftp and install ssh so we can maintain and administer the kiosk remotely across the network with ssh and scp.

4.3 Booting normally from the console

There are times, especially during testing, when you'd like to work from the console in a normal init level 3 mode. To do this, press **Ctrl−Backspace** to reboot the system. When LILO: appears type:

```
LILO: linux init 3
```

and log in as usual.

5.Other Considerations

5.1 Screensavers

Depending on where you deploy your kiosk, you may not want all the default xscreensaver images to appear. For our *Hands−On Biodiversity Gallery* we wanted to display only the animated fractal images that suggested life science themes.

After testing out several screensaver images, we decided that we to select randomly between two choices: *coral* and *forest.*

To limit xscreensaver to displaying these two, we created the file .xscreensaver in the filesystem root (/) with the following options:

/.xscreensaver

```
programs: \
        coral −root \n\
        forest −root \n
```

5.2 Sound Files

Our kiosk machine contains a Sound Blaster AWE64 card with attached speakers. We chose this card specifically because it works well with Linux sound drivers.

The MySQL database that drives our Biodiversity Gallery kiosk points to a collection of sound files that a visitor may listen to, including a ``bird song quiz.'' The downside of using Linux Netscape for a kiosk is that some of the multimedia aspects are primitive, compared to Windows and Macintosh.

To enable sound, we implemented a Unix Netscape plugin called Xswallow written by Caolan McNamara (http://skynet.csn.ul.ie/~caolan/docs/XSwallow.html). All the sound plugins we tested create a separate web page as the sound is being played (unless the sound file is embedded). Of the programs we tested, Xswallow worked the best and had the cleanest display page.

Although the extra page is inconvenient, we decided that having the stability of Linux was more important than the extra click required to return to the previous screen.

An additional problem is that when a sound is selected, the Xswallow page displays ``Click to Abort Swallow of type audio/basic''. Clicking merely returns a blank page. I made a slight change to the author's source code for UnixShell.c, changing

```
        char *text2 = "Click to Abort Swallow";
```

to read

```
        char *text2 = "Click BACK Button to return from playing file";
```

The phrase ``of type audio/basic'' appears to come directly from Netscape rather than Xswallow.

After recompiling, I added the plugin to Netscape.

A last note about sound files: the original set of files we received were a mix of .au and .wav files. Our experience was that the .au files worked the most reliably in Linux and we converted all sound files to .au format.

5.3 Trackballs

Not having the funding for a touchscreen display, we opted for a trackball (and no keyboard) for our kiosk navigation. Our colleague, Andy Rauer at the Ontario Science Centre in Toronto, tipped us off to using ``industry−strength'' trackball components from Happ Conrols Inc., Elk Grove Village, Illinois http://www.happcontrols.com/ (check under their Interactives section).

One of the things we needed to do was disable any right−mouse button click equivalents from being used in Netscape Navigator because right−click events invoke user dialogs.

By building our own trackball, we were able to wire it any way we wanted. We wired in two push−button controls, both with a left−mouse button equivalent. The buttons are arranged on either side of the trackball, allowing ease of use from right−handed and left−handed visitors. Our gallery designers then took the trackball parts and built them into the gallery's kiosk enclosure.

For our kiosk we purchased the following components for building a heavy−duty trackball designed to withstand visitor use (and abuse):

- A 2 1/4−inch Atari Trackball Assembly (we selected red)
- Trackball Interface Kit for IBM (Microsoft serial mouse compatible)
- Illuminated Push Buttons (red)

From a local electronics vendor, we purchased a Hammond Manufacturing plastic handy case (Cat no. 1591ABK) for housing the electronics board for the trackball.

Anticipating additional kiosks, and potential part failure, we ordered four sets of each of the above for spare parts and testing.

5.4 Designing or modifying HTML pages for kiosks

In the best of all possible worlds, you will have the opporunity to design your kiosk pages from scratch. Bearing in mind that using a kiosk is a different experience from sitting at a desk browsing the web from a personal workstation, the following guidelines are useful design principles:

- Keep text passages very short and use large fonts.
- Avoid screens that require scrolling.
- Use colorful, large, eye−catching images on every page
- Create embedded navigation if possible, e.g., return to previous page, go to next page, go to kiosk menu, etc., should all be designed right into the pages themselves.
- Avoid anything that requires a keyboard. Design for clicking.
- If you use sound files, keep the duration short. Embed them into the page if possible.
- Make everything punchy and snappy. Design for short attention spans and pass−through traffic. Save your theses and exegeses for elsewhere.

Unforunately you will sometimes inherit pages that were designed for web browsing rather than kiosk browsing. In these cases it may be too time−consuming or too difficult to redesign them as kiosk pages.

When implementing standard web pages for a kiosk display:

- Remove all external A HREF's, including MAILTO's.
- Remove all unnecessary verbiage, logos, etc. from the pages.
- Check font sizes and increase if necessary.
- Remove any animated GIF's that don't pertain directly to the pages being viewed.
- Actually, remove any animated GIF's, period.
- Do anything else you can to simplify the page design.
- Check into using server−side includes to add some fundamental navigation or a consistent graphic to tie the pages together.

6.Future Directions

Our previous attempts at creating browser−based kiosks with Windows−based PC's were unsatisfactory. The units were unstable, freezing up frequently. The Linux approach has given us a reliable, robust kiosk in a public gallery that has stood up well to visitor usage. Our success with this project has led us to consider other kiosk possibilities.

6.1 Remote Kiosks

One of our future goals is to implement remote kiosks in the greater Toronto area where people can browse some of our site information and, hopefully, be attracted enough by it to visit the ROM in person. These might be deployed at places such as conventions, trade shows, shopping malls or special exhibits. We are also thinking in terms of deploying kiosks in more distant locations, perhaps in public libraries.

For this project we would likely implement modem−based Linux boxes that could ``call home'' to update files on a routine basis and could be dialled into for maintenance and updates.

6.2 Touchscreens

When the technology gets cheaper, we would like to experiment with flat−panel touchscreen displays for kiosks. The combination of flat−panel display and very small PC units would enable us to deploy kiosks with a very small footprint.

6.3 Keyboards

Ultimately we may need to deploy kiosks that utilize keyboards. This will present a new set of challenges in terms of blocking unwanted keystrokes. If anyone would like to contribute information on this, or to any other aspect of the Kiosk−HOWTO, please email me at genew@rom.on.ca or my alternative email address njo@interlog.com.

6.4 Collaboration

I would be very interested in collaborating with other kiosk builders to extend the Kiosk−HOWTO for other situations. Kiosks are a bit like Perl: ``There is more than one way to do it."
