

窗口管理器 Openbox 入门指南

徐小东*

2008 年 8 月 11 日

*网名 Toy，在 <http://linuxtoy.org> 上可以找到他。

目录

1	前言	3
2	我喜欢 Openbox 的原因	3
3	如何安装 Openbox	3
3.1	从发行版安装	3
3.2	从源代码编译安装	4
3.2.1	编译依赖	4
3.2.2	编译及安装	4
4	运行 Openbox	5
5	配置 Openbox	5
5.1	使用 ObConf	5
5.2	创建菜单	6
5.2.1	一级菜单	6
5.2.2	几个特殊的菜单项目	7
5.2.3	二级菜单	8
5.3	设定键盘和鼠标绑定	8
5.3.1	配置文件	8
5.3.2	键盘绑定	8
5.3.3	鼠标绑定	9
5.4	控制应用程序	10
5.4.1	获取 name 和 class	11
5.4.2	使用通配符	11
5.4.3	实例	12
5.5	使用自动启动脚本	12
5.5.1	前提	12
5.5.2	创建	12
6	提示与技巧	13
6.1	Openbox 常用快捷键	13
6.2	不用敲 startx 直接进入 Openbox	13
6.3	设置壁纸	13
6.4	设置 GTK 主题	13
7	参考资源	14
8	版权声明	14

1 前言

也许你听说过 Blackbox 和 Fluxbox，那么，Openbox 又是什么？Openbox 跟它们很相似。据 Openbox 官方称，Openbox 是一个可高度定制且包含广泛标准支持的下一代窗口管理器。近来，我玩 Openbox 是越来越上心，自以为有些小得，遂立此文，以助新手快速入门。

2 我喜欢 Openbox 的原因

从了解到学习，再到天天使用 Openbox，我主要基于以下理由：

1. 速度非常快，资源占用极少。
2. 可高度定制化，能够对应用程序加以灵活控制。
3. 能够绑定键盘和鼠标。比如，你可以为程序设定启动快捷键，也可以为最小/大化窗口设置热键。对于鼠标，同样如此。
4. 具有自动启动脚本，能够随机自动启动各种程序。
5. 主题资源丰富，与 *box 类视觉样式兼容。Openbox 的外观是简约而不简单。
6. 支持会话，可深入 GNOME、KDE、Xfce 等桌面环境使用。换句话说，Openbox 可替代这些桌面环境中默认的窗口管理器，而其他桌面组件仍旧保持原样。

3 如何安装 Openbox

要安装 Openbox 有两种方法：其一是直接从所用的 Linux 发行版中安装；二是下载其源代码，自行编译安装。前者的优势是省事，不过有时候安装的 Openbox 并非最新版。而后者虽然稍显麻烦，但可以实现更加灵活的掌控。

3.1 从发行版安装

Openbox 已被包含到大多数流行的 Linux 发行版中。因此，使用该发行版的包管理工具来安装 Openbox 是一件十分容易的事情。

- Archlinux 用户执行以下命令可以安装 Openbox: `pacman -S openbox`
- 而 Debian/Ubuntu 用户可以执行下面的命令: `sudo apt-get install openbox`

- Fedora 用户也可以使用 yum 来安装 Openbox: `yum install openbox`

3.2 从源代码编译安装

然而，有时候从发行版安装的 Openbox 并非最新版本，或者在你所用的 Linux 发行版中不能找到 Openbox，这时候可以考虑从源代码编译安装它。

3.2.1 编译依赖

要从源代码编译安装 Openbox，首先要准备以下依赖包：

- C 编译器（比如 GCC）
- Libc
- Xlib
- Glib-2
- LibXML-2
- Pango
- Startup-notification（可选，推荐安装）
- XCursor（可选，推荐安装）
- Pkg-config

不要被这些依赖包所吓倒，其实多数 Linux 发行版已经默认安装了。你所要做的就是——检查和核对而已。

3.2.2 编译及安装

一旦准备好编译 Openbox 所需的依赖包，在下载并解包源代码文件后，便可按如下指令来编译并安装 Openbox：

```
./configure --prefix=/usr --sysconfdir=/etc  
make  
sudo make install
```

哈，很熟悉的配置、编译及安装三部曲。

4 运行 Openbox

如果你使用了 GDM/KDM 之类的图形登录管理器，那么在系统登录时从会话类型菜单中选择 Openbox 相关条目即可启动 Openbox。

另一种启动 Openbox 的方式是命令行。只需向主目录中的 `~/.xinitrc` 文件加入如下内容即可：

```
exec openbox
```

另外，包含会话支持功能的 Openbox 可以使用下面的内容代替：

```
exec openbox-session
```

5 配置 Openbox

当 Openbox 安装完成后，其默认的选项设置可能并不合你意。因此，你可以根据自己的使用习惯来对 Openbox 进行配置。要配置 Openbox，既可以采用图形化的工具，也可以直接编辑文件手动配置。

5.1 使用 ObConf

ObConf 是一个图形化的 Openbox 配置工具，可以完成一些基本的配置，包括：

- 安装、选用或打包 Openbox 主题
- 配置外观 (像窗口按钮布局、字体等) 及窗口行为
- 配置虚拟桌面及桌面边距
- 配置 Dock

可以使用以下指令来安装 ObConf。

- 在 Archlinux 中，可以执行：`pacman -S obconf`
- Debian/Ubuntu 用户可以执行：`sudo apt-get install obconf`
- Fedora 用户可执行：`yum install obconf`

鉴于 ObConf 的操作非常直观，在此就不赘述了。大家花点时间琢磨一下，很快就可以上手。

5.2 创建菜单

最直接配置 Openbox 的方式是手工编辑其配置文件。不过，在编辑之前，一些语法规则是需要掌握的。Openbox 默认的菜单文件 (系统级) 位于：

```
/etc/xdg/openbox/menu.xml
```

当你在桌面右击鼠标时，将显示 Openbox 的 Root 菜单。通过此菜单，用户可以执行启动应用程序、注销、退出等操作。

为了避免更新 Openbox 程序时该文件被覆盖，复制一份到：

```
mkdir ~/.config/openbox
```

```
cp /etc/xdg/openbox/menu.xml ~/.config/openbox/
```

从 menu.xml 文件的扩展名我们可以看出这是一个 XML 文件。使用任何文本编辑器皆可打开该文件。除了文件首行的 XML 声明外，所有的 Openbox 菜单项目都是由 `<openbox_menu>` 标签所包围的。

5.2.1 一级菜单

首先，让我们来创建一个一级菜单。要定义一个菜单，我们需要使用 `<menu>` 标签。该标签具有 `id`、`label`、以及 `execute` 属性：

- `id`: 每一个菜单项目都必须指定一个唯一的 `id`，以用来区分其他菜单项目。
- `label`: 用来描述一个菜单的名称。
- `execute`: 执行一个命令，常用于创建动态菜单。

例如，我们可以将 Openbox 的 Root 菜单定义如下：

```
<menu id="root-menu" label="Openbox Root Menu">  
...  
</menu>
```

如果一个菜单中没有任何项目，我想是不能称其为真正的菜单吧。所以，我们还需要为菜单创建具体的菜单项目。要定义菜单项目，我们可以使用 `<item>` 标签。与 `<menu>` 标签一样，`<item>` 标签也具有描述菜单项目名称的 `label` 属性。我们试着扩展上面的例子：

```
<menu id="root-menu" label="Openbox Root Menu">  
<item label="URxvt">  
...  
</item>  
...  
</menu>
```

现在，我们已经有了一个名为 URxvt 的菜单项目。根据该名称的命名初衷，我们当然希望通过该菜单项目能够启动 URxvt 终端程序。要完成这

个任务，我们需要使用 `<action>` 标签。该标签主要用来执行一个操作，比如启动程序。`<action>` 标签具有 `name` 属性。我们继续扩展先前的例子：

```
<menu id="root-menu" label="Openbox Root Menu">
<item label="URxvt">
<action name="Execute">
...
</action>
</item>
...
</menu>
```

该例中 `name` 属性已经包含了“Execute”值，其作用是启动一个程序。

我想，你已经猜到接下来我们需要做什么了。要启动程序，没有具体的命令是不行的。这可以通过 `<command>` 标签来完成。我们仍以前面的例子来说明：

```
<menu id="root-menu" label="Openbox Root Menu">
<item label="URxvt">
<action name="Execute">
<command>urxvt</command>
...
</action>
</item>
...
</menu>
```

值得一提的是，命令可以附带路径或选项参数。

5.2.2 几个特殊的菜单项目

在此，我们介绍几个较为特殊的菜单项目：

- 分隔线：可以使用 `<separator>` 标签在菜单项目间绘制一条分隔线。`<separator>` 标签同样具有 `label` 属性。需要注意的是，因为 `<separator>` 标签没有具体的内容，所以在关闭该标签时，应照下面的方式进行：

```
<separator />
```

这是另一个例子，包含 `label` 属性：

```
<separator label="tools" />
```

- 重新配置 Openbox：使用该菜单项目的好处是，当你对 Openbox 的配置文件进行修改后，不必注销系统便可即时生效。该菜单项目定义如下：

```
<item label="Reconfigure">
  <action name="Reconfigure" />
</item>
```

- 重新启动 Openbox:

```
<item label="Restart">
  <action name="Restart" />
</item>
```

- 退出 Openbox:

```
<item label="Exit">
  <action name="Exit" />
</item>
```

- 注销会话:

```
<item label="Session Logout">
  <action name="SessionLogout" />
</item>
```

注意，需要启动带会话支持的 Openbox 才有效。

5.2.3 二级菜单

掌握了创建一级菜单的方法，二级菜单则如法炮制即可。当二级菜单创建完毕之后，可按以下方法加入一级菜单：

```
<menu id="tool-menu" />
```

5.3 设定键盘和鼠标绑定

5.3.1 配置文件

Openbox 默认的键盘和鼠标绑定文件位于：

```
/etc/xdg/openbox/rc.xml
```

同样复制一份到 .config/openbox/:

```
cp /etc/xdg/openbox/rc.xml ~/.config/openbox/
```

5.3.2 键盘绑定

Openbox 的键盘绑定使用 <keyboard> 标签，形如下面的内容：

```
<keyboard>
...
</keyboard>
```

要定义一个具体的键盘绑定，我们需要使用 `<keybind>` 标签。该标签具有 `key` 属性，其作用是指定快捷键。常用的修饰键如下：

- S - Shift
- C - Ctrl
- A - Alt
- W - Win (即 Windows 徽标键)

例如，假设我要为最大化窗口指定快捷键为 Alt-F6，那么，可以定义如下：

```
<keyboard>
<keybind key="A-F6">
<action name="MaximizeFull" />
</keybind>
```

...

```
</keyboard>
```

又如，我想为 URxvt 指定启动快捷键 Win-u，可作下面的定义：

```
<keybind key="W-u">
<action name="Execute" />
<command>urxvt</command>
</keybind>
```

其实，Openbox 已经默认定义了很多键盘绑定，如切换活动桌面、提升窗口等等，具体内容都可以在上面的 `rc.xml` 文件中找到。

5.3.3 鼠标绑定

鼠标绑定使用 `<mouse>` 标签定义。具体如下：

```
<mouse>
```

...

```
</mouse>
```

与键盘绑定不同的是，鼠标绑定会要求你首先设置情景，也就是触发鼠标绑定的对象，即 `<context>` 标签。常用的 Context 如下表所示：

名称	说明
Frame	除桌面外的任何窗口
Client	应用程序窗口，不含窗口边框
Desktop	桌面
Root	与 Desktop 相似，通常用于 Root 菜单
Titlebar	窗口标题栏
Top, Bottom, Left, Right	窗口的上、下、左、右四边
TLCorner, TRCorner, BLCorner, BRCorner	窗口的四角
Icon	窗口图标
Iconify	最小化按钮
Maximize	最大化按钮
Close	关闭按钮
AllDesktops	所有桌面按钮
Shade	折叠按钮
MoveResize	移动并调整窗口大小

扩展上面的例子：

```

<mouse>
<context name="Titlebar">
...
</context>
</mouse>

```

Titlebar 说明这是针对标题栏的鼠标操作。

然后，即可使用 `<mousebind>` 标签进行具体的鼠标绑定。该标签具有 `button` 和 `action` 属性：

- `button`: 该属性指定使用哪一个鼠标按钮来触发鼠标绑定，如左键、右键、滚轮等。
- `action`: 用来指定一个鼠标触发的事件，如按下、单击、双击等。

例如，当我们双击一个窗口时让其最大化，可以作如下定义：

```

<mouse>
<context name="Titlebar">
<mousebind button="Left" action="DoubleClick">
<action name="ToggleMaximizeFull" />
</mousebind>
</context>
</mouse>

```

5.4 控制应用程序

记得我们曾经在介绍如何将终端窗口嵌入桌面时，使用过一个名叫 Devil's

Pie 的窗口匹配工具。其实，Openbox 就具有类似的功能。通过 Openbox，我们可以控制每一个应用程序的初始状态。比如，URxvt 启动后就没有边框，Gvim 启动后直接放置到第二个桌面，等等。

跟键盘绑定和鼠标绑定一样，控制应用程序的设置也位于 rc.xml 文件中。应用程序的设置使用 <applications> 标签定义：

```
<applications>
```

```
...
```

```
</applications>
```

具体到一个单独的应用程序，则使用 <application> 标签，该标签具有 name、class 及 role 属性：

- name: 窗口名称，用来指定一个确定的窗口。
- class: 窗口类名，其作用同上。
- role: 可选属性，对窗口作进一步的匹配，比如可用来区分是针对窗口还是针对对话框进行控制。

其中，name 属性和 class 属性可以仅使用其一，也可同时使用。举例如下：

```
<applications>  
<application name="" class="" role="">
```

```
...
```

```
</application> ...
```

```
</applications>
```

5.4.1 获取 name 和 class

当我们针对应用程序进行设置时，首先需要获取该程序的 name 或 class。这可以通过执行下面的命令来完成：

```
xprop WM_CLASS
```

该命令执行后，鼠标指针将变成十字型，然后在程序窗口中单击即可。

例如，单击 Gvim 获得的结果如下：

```
WM_CLASS(STRING) = "gvim", "Gvim"
```

这说明，该窗口的 name 为 gvim，class 为 Gvim。

5.4.2 使用通配符

Openbox 在匹配窗口的 name、class 及 role 时，可以使用通配符 * 和 ?。其中，* 用来匹配任意多个字符，而 ? 仅能匹配单个字符。

5.4.3 实例

让 Gvim 启动后直接放置到第二个桌面，可定义如下：

```
<application name="gvim">  
<desktop>2</desktop>  
</application>
```

去掉 URxvt 的窗口边框，作下面的定义：

```
<application name="urxvt">  
<decor>no</decor>  
</application>
```

让 Mirage 启动后窗口最大化，可定义为：

```
<application name="mirage">  
<maximized>>true</maximized>  
</application>
```

去掉所有窗口的边框：

```
<application name="*">  
<decor>no</decor>  
</application>
```

上面的例子仅作抛砖引玉之用。其实，通过 Openbox 这项特性，你还可以设置窗口的位置、是否最小化、全屏显示等诸多属性。在 rc.xml 文件的 applications 部分有详细的说明，大家只要照做即可。

5.5 使用自动启动脚本

通过 Openbox 的自动启动脚本，我们可以随机启动一些程序，像输入法、面板等等。

5.5.1 前提

使用自动启动脚本的前提是，你必需使用包含会话功能支持的 Openbox，即 openbox-session。

5.5.2 创建

Openbox 默认的自动启动脚本文件位于：

```
/etc/xdg/openbox/autostart.sh
```

你可以在建立自己的自动启动脚本时参考参考。

使用下列命令来建立一个自己的 autostart.sh 文件：

```
vim /.config/openbox/autostart.sh
```

比如，我们要开机即加载面板程序 Pypanel，可以加入下面的内容：

```
pypanel &
```

很简单，是不是？如果要加入多个程序，则分行写即可。

6 提示与技巧

6.1 Openbox 常用快捷键

- 在桌面按鼠标中键，将显示窗口列表菜单
- Win-d: 隐藏屏幕上所有的窗口
- Alt-Tab: 切换窗口
- Win-(F1-F4): 依次转到第1-4个工作区
- Alt-F4: 关闭窗口
- Alt-Esc: 将下面的窗口提升到上面

6.2 不用敲 startx 直接进入 Openbox

在没有使用图形化登录管理器的情况下，要进入 Openbox，我们需要敲 startx。如果使用 bash ¹的话，在 ~/.bash_profile 中加入下列内容，可以免敲 startx:

```
if [[ -z "$DISPLAY" ]] && [[ $(tty) = /dev/vc/1 ]]; then
startx
logout
fi
```

6.3 设置壁纸

有多种工具可以设置壁纸，我使用的是 feh:

```
feh -bg-scale /path/to/wallpaper.png
```

示例中的壁纸路径及名称需要换成你自己的。

然后将:

```
eval 'cat $HOME/.fehbg' &
```

加入 Openbox 的 autostart.sh 文件。

6.4 设置 GTK 主题

不用使用任何工具，在 ~/.gtkrc-2.0 中加入:

```
gtk-theme-name = "Infini-Herbe"
```

这将设置 GTK 程序的主题。将引号中的内容换成你自己喜欢的主题名称（下同）。

¹zsh 用户可以加到 ~/.zprofile 文件。

gtk-icon-theme-name = “ALLGREY”

设置所用的图标主题。

gtk-cursor-theme-name = “Vanilla-DMZ-AA”

设置鼠标指针主题。

gtk-font-name = “Luxi Sans 10”

设置 GTK 程序用的字体。

gtk-toolbar-style = *GTK_TOOLBAR_ICONS*

设置工具栏样式，我喜欢只显示图标。

7 参考资源

- Openbox 主页: <http://icculus.org/openbox/>
- Openbox 文档: <http://icculus.org/openbox/index.php/Help:Contents>
- Openbox 主题: <http://www.box-look.org/index.php?xcontentmode=7402>
- ObConf: <http://icculus.org/openbox/index.php/Obconf>
- Archlinux Wiki: <http://wiki.archlinux.org/index.php/Openbox>
- Gentoo Wiki: http://gentoo-wiki.com/HOWTO_Openbox
- Ubuntu Wiki: <https://help.ubuntu.com/community/Openbox>

8 版权声明

本文档以“署名-非商业性使用-禁止演绎 2.5 中国大陆”的创作共用协议发布。故你可以自由散布它，但必须保留此版权信息。