

## 标题：参赛 LiveCD 制作文档

### 摘要：

本文论述了基于 Asianux Workstation 3 操作系统的 LiveCD 的设计、创作思路、架构、制作过程、应用领域、过程中的难点和解决方法以及开发团队简介。

### 项目规格书

本 CD 是一个专为上网本（Netbook）设计的简易系统。系统的设计目标为：

- 用于局域网环境的上网。支持常见的无线网卡模块，支持以 DHCP 方式加入局域网。
- 预装 QQ for Linux 1.0 preview3。
- 预装 Firefox 3.0.4 以及国内用户常用的插件，与普通桌面操作系统相比，上网功能没有任何裁减
- 预装 Prism，可直接访问各种丰富的 Web App。桌面建立了 plukr、google 文档、google 日历的快捷方式。
- 采用 xfce 桌面环境。保留文件管理、X 终端、ssh、ftp、vim 等丰富的工具。
- 体积小(实际做出来的光盘为 299M 字节)，快速，能够迅速启动和关机。



## 创作思路

最初决定从最小化安装的 Asianux Workstation 3 系统开始搭建环境，后来发现这样做工作效率比较低，因此调整顺序为

- 安装一个 Red Flag Desktop 6 操作系统，作为制作 LiveCD 的环境
- 完善制作环境：安装虚拟机软件 VirtualBox 2.0.4，安装 sshd 服务
- 在 VirtualBox 中创建 Linux 2.6 虚拟机，此后的工作都在虚拟机中完成
- 在虚拟机中安装完整版的 Asianux Workstation 3（简称 aw3），此系统将作为制作 LiveCD 的基础操作系统
- 编译安装软件包
- 安装预编译软件包
- 定制界面
- 裁减系统
- 清理操作相关的历史记录文件
- 安装 LiveCD 制作工具，制作 iso 文件
- 测试 iso 文件

按照上述思路，我们把团队分成了两组，一人负责各种软件包的下载和编译、安装测试，两人一起负责虚拟机中的工作，此后的过程就比较顺利了。

## 光盘的应用领域

这张光盘是为经常上网的网虫们准备的，特别是那些时刻挂在 QQ 上的聊友，经常使用 plurk 日志汇报自己的工作、心情、衣食住行动态的微博客爱好者，经常使用在线文档协作办公的上班族，以及常常把系统弄崩溃需要救急的 Linux 爱好者。

光盘的特点有 3 个

- 第一就是全功能的 Web 体验。我们没有采用其他迷你型的 Web 浏览器，在预装了主流插件的 Firefox 3 中上网，感觉更加舒适，另外 QQ for Linux 也几乎是必备软件；
- 其次就是飞快的开关机速度，特别是关机，几乎是瞬间完成；
- 最后就是几乎全攻能的工具命令，包括 vim、lftp、ssh、tar、vnc、fdisk 等等，只删除了一个 mc（因为功能重复），这就很适合当应急光盘了。

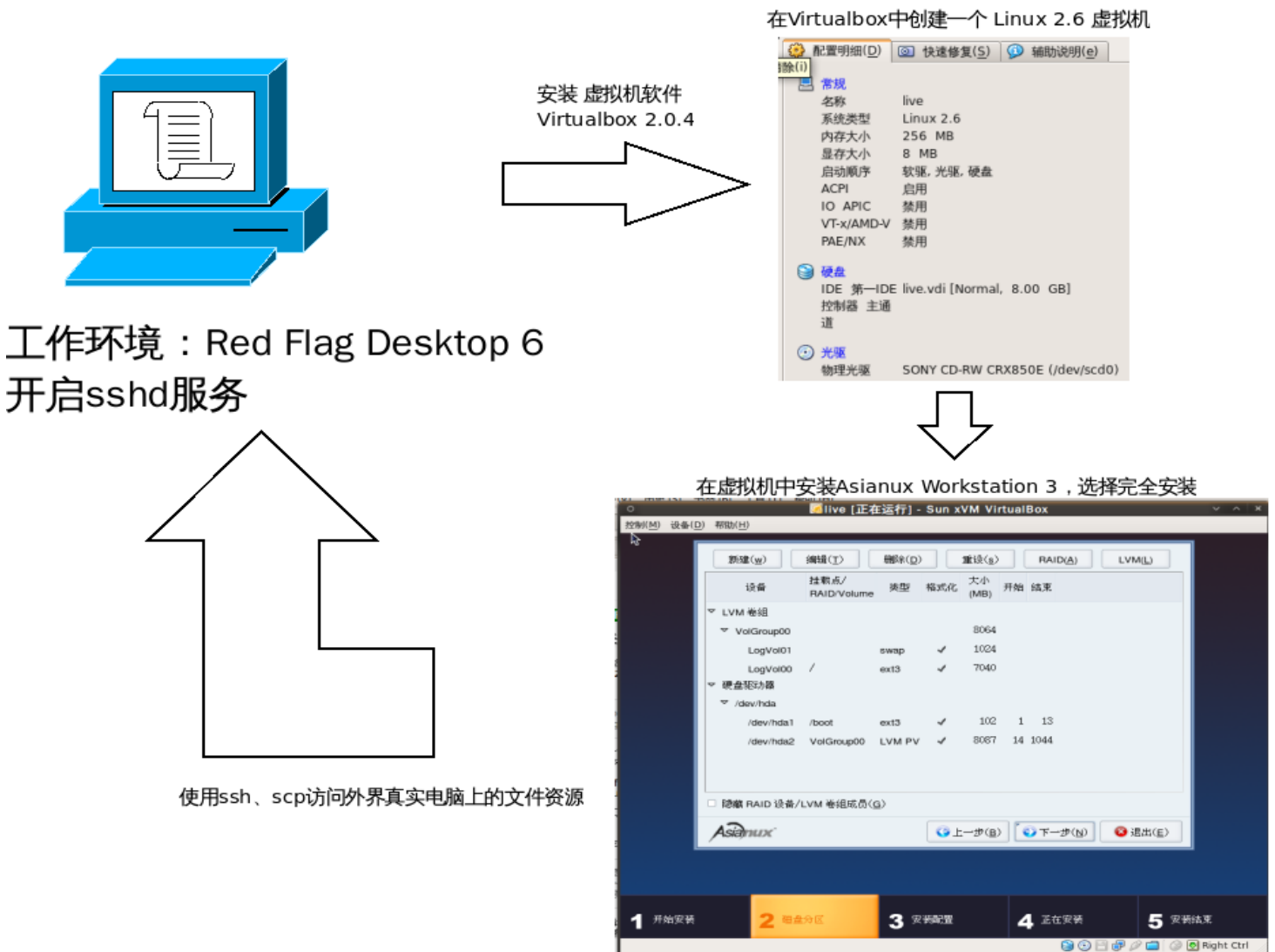
## 分析设计文档

### 设计架构

开发环境及开发工具

- 操作系统环境：Red Flag Desktop 6
- 主要工具：
  - VirtualBox 2.0.4
  - sshd 及 ssh 客户端
  - gcc 编译工具链
  - RPM 包管理器

开发环境的示意图如下



虚拟硬盘的容量选择为 8G，不能少于 4G，因为系统安装完成之后就占用了 3.7G，后续的很多操作都需要磁盘空间。  
安装过程中，网络选择为 DHCP 方式。当虚拟机中的系统装好之后，重启动后 IP 将被分配为 10.0.2.15。

## 虚拟网络的使用

Virtualbox 为每一个运行的虚拟机提供了一个虚拟的局域网环境。虚拟机将被内建的 DHCP 服务器分配成 10.0.2.15，而主机则被虚拟分配到 10.0.2.2 上。这样，从主机上复制文件，在虚拟机中就可以用

- scp 用户名@10.0.2.2:/路径/文件 虚拟机本地目录

同样的，在虚拟机中把文件复制到主机可以用

- scp 虚拟机本地文件 用户名@10.0.2.2:/主机目录

如果需要执行复杂的操作，可以在虚拟机中执行

- sftp 用户名@10.0.2.2

如果需要在虚拟机中远程操作主机，可以执行

- ssh 用户名@10.0.2.2

要注意：

上述 IP 地址都是虚拟的，只有在虚拟机中能够使用。也就是说，我们不能在外界主机上对 10.0.2.0 网段做任何操作，主机上也不存在这个网段。因此内外通讯的工作全部都要在虚拟机中执行。

## 系统的制作步骤

### 安装软件包

首先，确定我们所缺少的软件包清单。所需如下：

1. Mozilla Firefox 3.0.4 网页浏览器
2. QQ for Linux 1.0 preview3 聊天软件
3. 文泉驿正黑字体
4. xfce 4.4 桌面环境
5. xdg-utils 1.0.2 桌面整合工具
6. Xdialog 对话框

其中需要编译的有

- 文泉驿正黑字体，在 <http://wqy.sourceforge.net> 下载源代码
- xfce 4.4 桌面环境，在 <http://www.xfce.org/download/> 下载源代码
- xdg-utils 1.0.2-6 桌面整合工具，在 <http://portland.freedesktop.org/xdg-utils-1.0> 下载源代码
- Xdialog 脚本中常用的 X 对话框，在 <http://xdialog.free.fr> 下载源代码

这几个都是 GNU 软件。在外界系统中编译这几个软件包。每个软件包的编译的步骤都是如下

- tar xzf 软件包.tar.gz
- cd 软件包
- ./configure && make

然后复制进虚拟机中，在虚拟机中执行安装

- make install
- cd ..
- rm -Rf 软件包

说明一下：

本来预计虚拟机中也可以完成编译，因为选择的是完全安装。但是实际操作发现，编译环境是不如意的，一方面缺少 intltool，导致多数软件包不能通过 configure（这容易解决），另外还缺少相当数量的 devel 包，而且目前主流的软件包都需要 glibc 2.8 以上的环境。整个环境改造起来相当麻烦，因此直接在外部分编译，然后在虚拟机中安装。

QQ for Linux 是腾讯公司发布的免费软件，在 <http://im.qq.com/qq/linux> 可下载到 rpm 安装包。安装命令如下

- rpm -i linuxqq-v1.0-preview3.i386.rpm

Firefox 3.0.4 是基于 MPL 协议的开源软件，在 <http://www.mozilla.com> 网站上已经提供了预编译的压缩包，解压就安装成功。在安装之前，需要卸载现有的 Firefox 1.5，执行命令：

- rpm -e firefox

卸载之后，把下载的压缩包解压到 /usr/lib/firefox，然后运行其中的 firefox 程序，设定为系统默认的网页浏览器。接下来在线安装以下附加组件：



**Adblock Plus** 0.7.5.5

广告已成往事！



**Adblock Plus: Element Hiding Helper** 1.0.5

Helps you create element hiding rules for Adblock Plus to fight the text ads.



**CustomizeGoogle** 0.76

Enhance Google search results and remove ads and spam



**DownThemAll!** 1.0.3

Firefox的批量下载工具。



**NoScript** 1.8.5

为您的 Firefox 提供额外保护：NoScript 只允许在您选择的信任域(例如您标示出的网站)上执行 JavaScript、Java 和其...



**PDF Download** 2.0.0.0

允许您选择对PDF文件的操作：下载，以外部查看器查看或者以HTML形式查看！



**Refractor for Prism** 0.2.1

Create Prism applications directly in Firefox



**Tab in Textarea** 0.6

Insert tab characters in textareas.

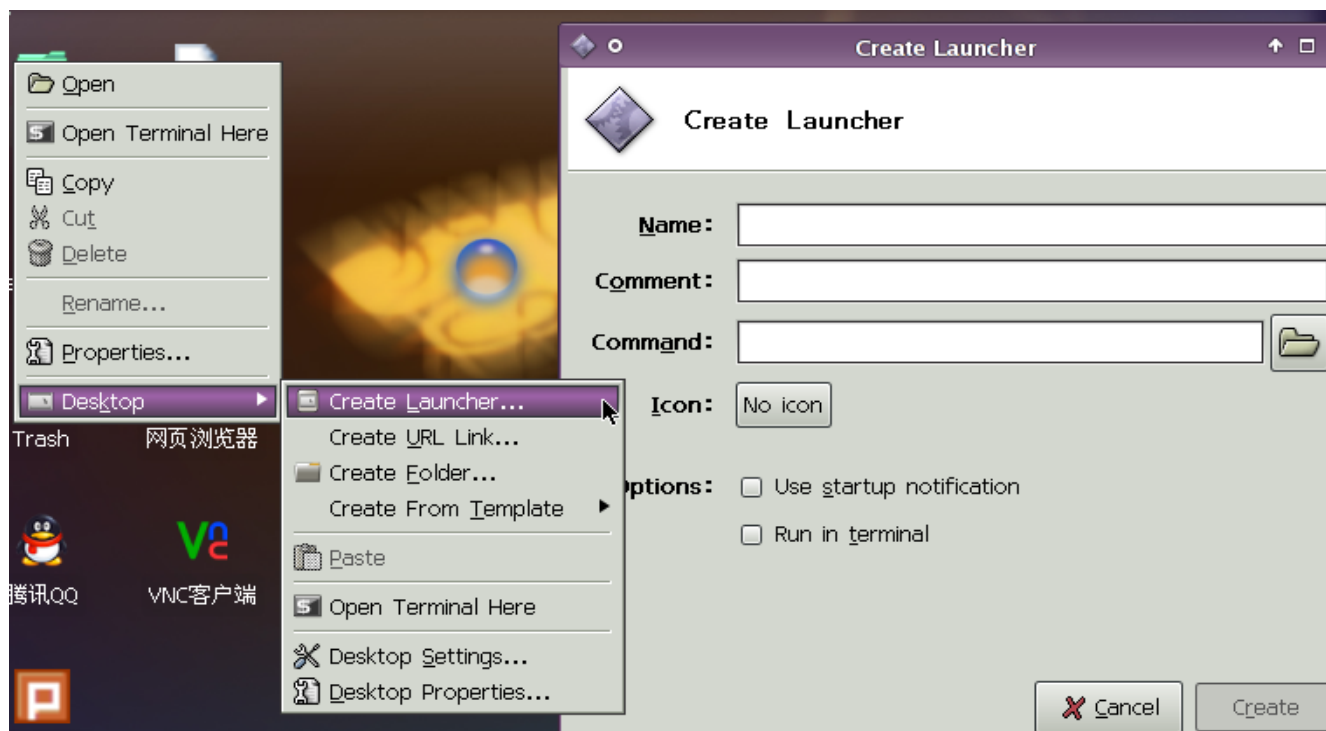


**Tab Mix Plus** 0.3.7.3

加强标签页浏览功能。

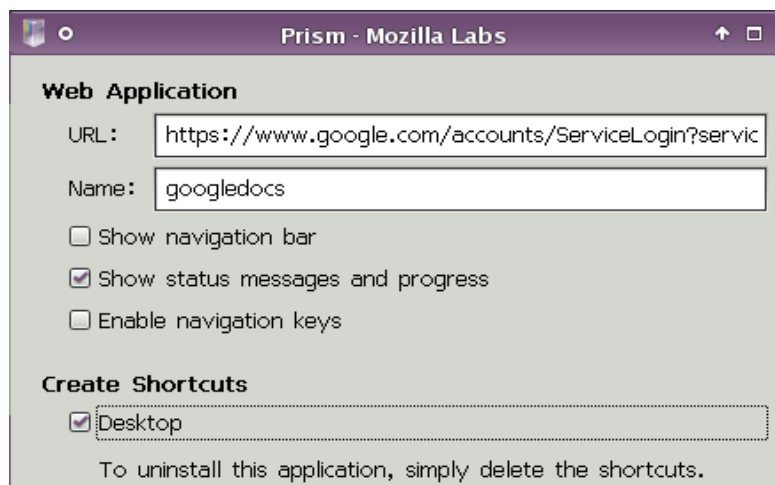
在线安装时，点菜单的工具 -> 附加组件 -> 获取附加组件，在搜索栏内填入附加组件的名称，就可找到相应的组件，点击开始在线安装。

至此，所有必需的软件包已经安装完毕。接下来为常用的程序创建桌面快捷方式。操作如下：



然后为热门的 Web App 创建 Prism 快捷方式。操作方式如下：



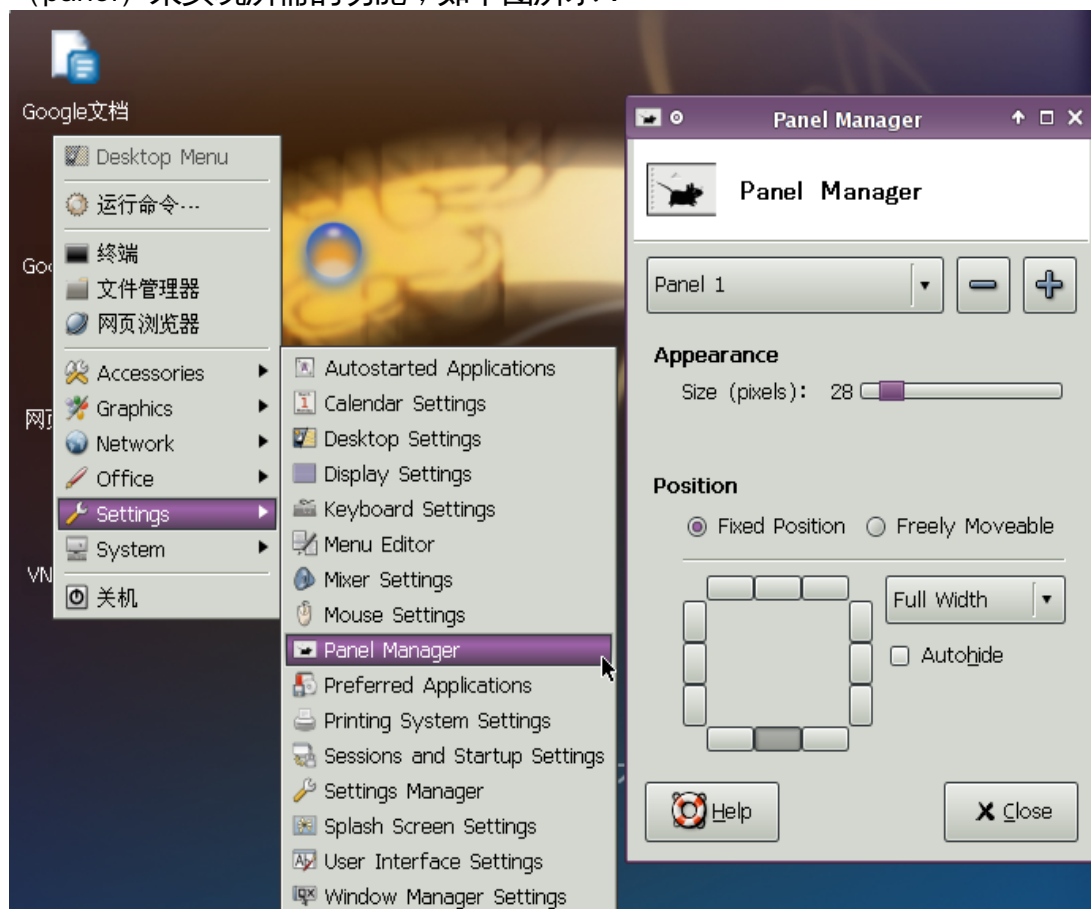


注意：Name 中不要出现中文，这是插件本身的 Bug。如果我们需要带有中文的快捷方式，可以先创建好 Prism 之后，再修改快捷方式的名称。

## 定制界面

上述软件包安装成功后，系统所需的功能基本上都已经具备了，可以开始定制界面了。首先我们需要从 KDE 更换 XFCE。从图形环境注销，在登录界面中，把会话（Session）修改成 xfce，然后登录，如果能够进入新的桌面环境，则表示安装的 xfce 已经可以正常工作。

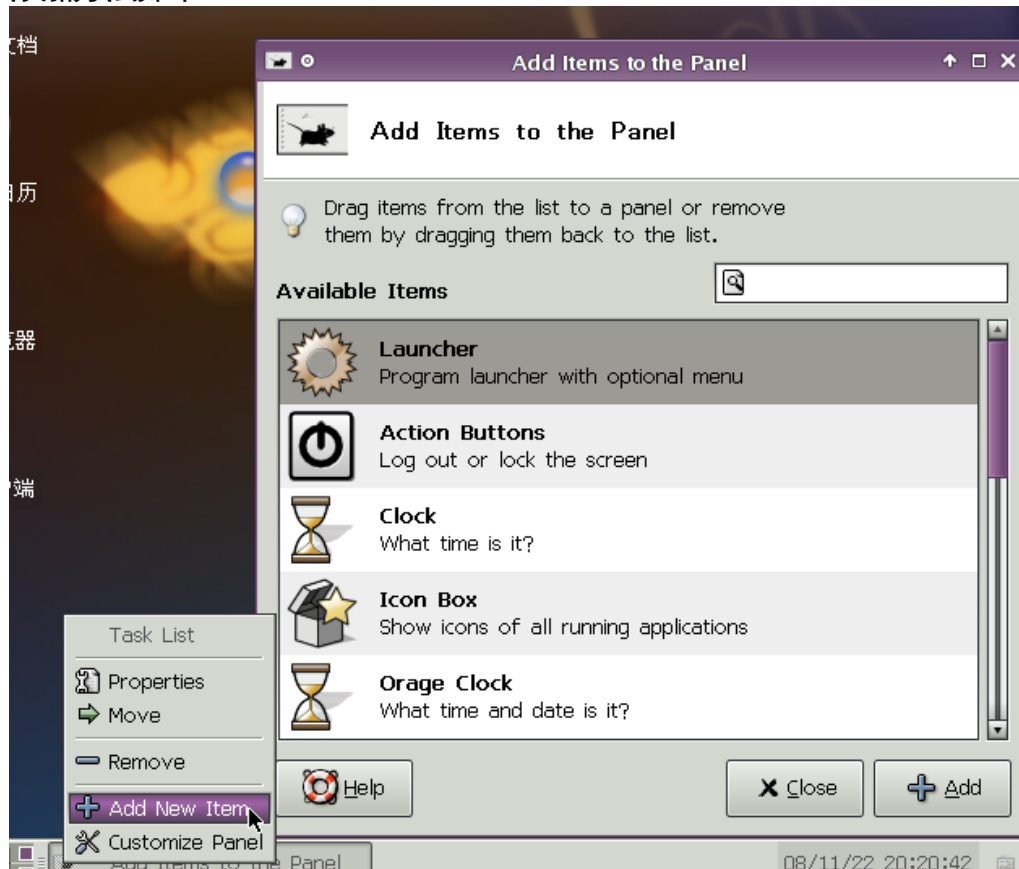
默认的 xfce 界面不够美观，我们需要调整一下。只需要更换一下 xfce 主题和壁纸，外观就会面目一新。另一个需要调整的是任务栏。事实上，xfce 不存在任务栏的概念。我们可以在 xfce 中添加一个面板（panel）来实现所需的功能，如下图所示：



然后在面板上添加以下项目：

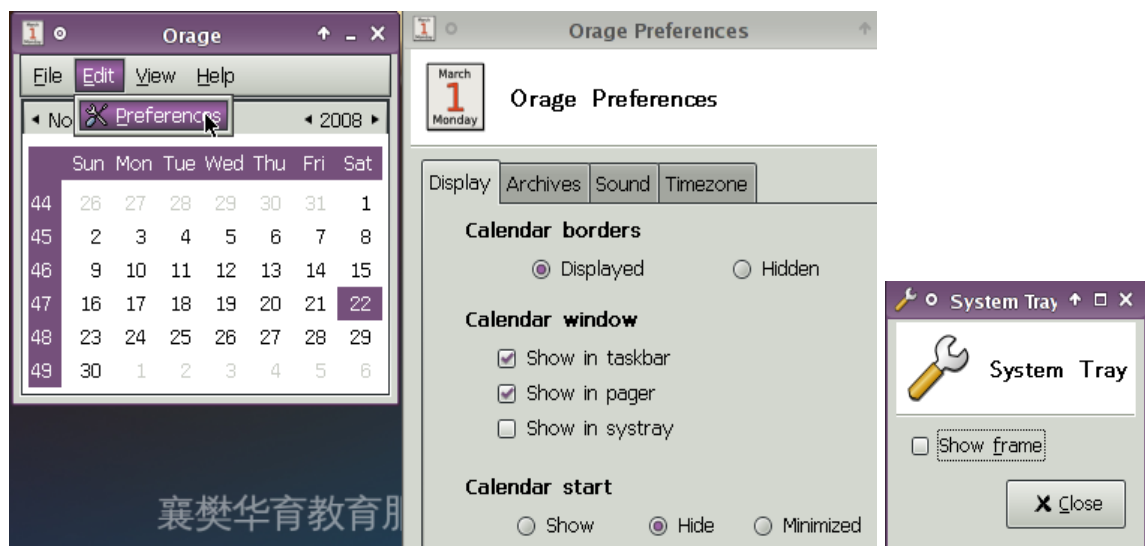
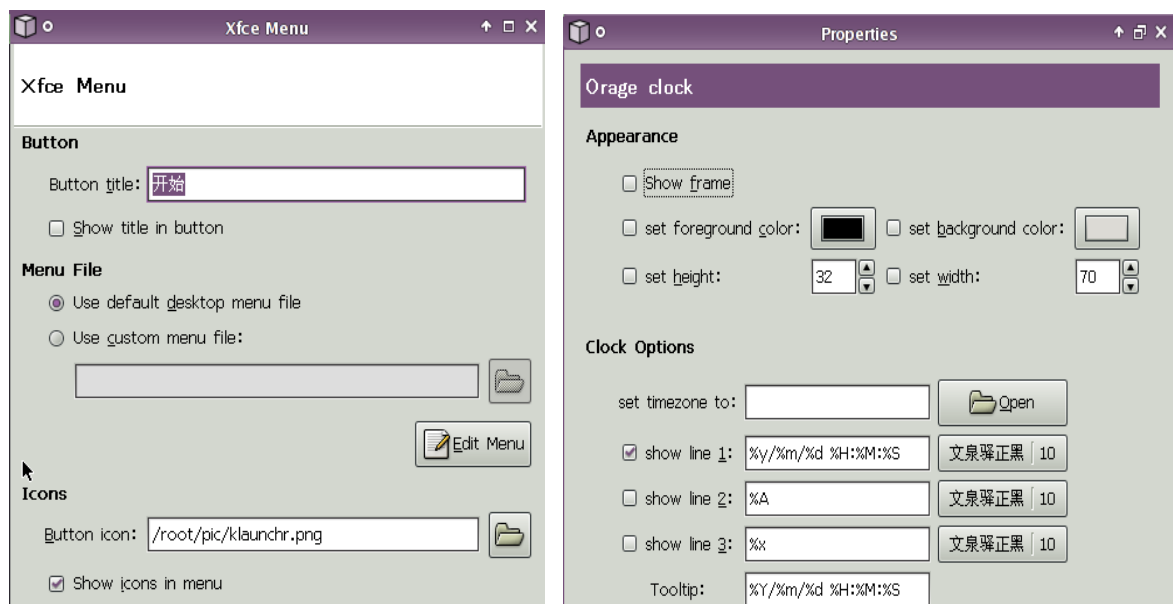
- Xfce menu，作为开始菜单
- Launcher，作为快速启动面板
- Pager，作为桌面切换工具（默认共 4 个桌面）
- Task List，作为任务列表栏
- Orage Clock，作为时钟
- System Tray，作为系统通知区

添加方法如下：



接下来，需要做一下美化，对面板中的各个组件设置如下：





至此，系统功能已经完成了。

## 裁减系统

如果不裁减的话，现在系统容量已经达到了 3.7G，很难把这么大的系统压缩到一张 CDROM 中。接下来开始精简。

首先删除帮助文件和手册。执行

- `cd /usr`
- `rm -Rf `find | grep -e "/doc" | tr "\ " "``
- `rm -Rf `find | grep -e "/info" | tr "\ " "``
- `rm -Rf `find | grep -e "/man" | tr "\ " "``

接下来删除不需要的头文件和源代码

- `cd /usr`
- `rm -Rf `find | grep -e "/src" | tr "\n" " "``
- `rm -Rf `find | grep -e "include" | tr "\n" " "``

最后卸载不需要的软件包，包括 KDE

- `rpm -e `rpm -qa | grep -e "-devel" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "gcc" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "kdebase" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "kde-i18n" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "php" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "ruby" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "perl" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "python" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "snmp" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "mod_" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "anaconda" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "selinux" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "chewing" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "freeradius" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "mysql" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "ocfs" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "ldap" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "dbmail" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "gimp" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "gnome-python" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "vfs" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "cyrus" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "docbook" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "emacs" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "sane" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "dhcp" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "httpd" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "heart" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "libdbi" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "-server" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "postgresql" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "sendmail" | tr "\n" " "``
- `rpm -e `rpm -qa | grep -e "ftpd" | tr "\n" " "``
- `rpm -e crontabs mc vnc-server tclx tcl-html db4-tcl audiofile vim-X11 squid ypbind ypserv yp-tools ww3m webalizer wireshark words`
- 在 rpmmanager 中
  - 全选 Development、Korean-support、Legacy-software-development、printing、Virtualization、web-browsers 中的软件，然后卸载
  - Japanese-support 中只留下 scim-input-pad 和 scim-qtimmm
  - 全选三个 middle 开头的中间件软件分类，然后删除

上述的操作中，某些包会因为依赖关系而卸载失败，这时不要强制删除，否则可能会导致问题。

删除不需要的字体，简体中文字体只留下文泉驿，繁体留下 ming(宋体)，其余删除，然后执行 fc-cache -fv，更新字库缓存。

删除多国语言支持文件，在下列目录中：

- /usr/share/i18n/locales、 /usr/share/locale、 /usr/lib/locale、 /usr/local/share/locale

删除全部不以 translit、en 和 zh 开头的目录和目录。

删除这个 65M 的大文件：

- /usr/lib/locale.archive

这样，系统就被减小到 1.5G。

## 设置自动进入 xfce

如果此时重启动系统，那么我们是没法进入 xfce 图形界面的，因为现在已经没有桌面管理器了（原来的是 kdm，已经被卸载）。这时会频繁的出现启动错误信息。为了防止出错，我们需要修改/etc/inittab 文件，寻找

- id:5:initdefault:

替换为

- id:3:initdefault:

原来的设置表示默认进入 X 环境，改成 3 则是进入字符界面。这样可以防止系统因尝试 xdm 管理器而导致频繁的打印失败消息。

现在要进 X 环境，必需运行命令

- startxfce4

为了能够在没有 kdm 的情况下，实现登录之后自动进入 X 环境，我们需要修改/etc/profile，在最后加一行：

- exec startxfce4

这样，启动系统后，首先会停留在字符界面，系统显示“Login:”然后等待输入用户名密码。当登录成功后，系统又会自动的运行 startxfce4 进入图形界面。

## 编写自动登录脚本

更进一步，对于 LiveCD 而言，账号密码没有太大的意义，能实现自动登录更好。为了达到这个目标，我们需要编写一个脚本/sbin/autologin，内容为

- #!/bin/sh
- exec 0</dev/\$1 1>/dev/\$1 2>&1
- shift
- exec \$\*

然后将权限改为 755

- chmod 755 /sbin/autologin

再修改文件 /etc/inittab,使 root 在 tty1 实现自动登录，找到

- 1:2345:respawn:/sbin/mingetty tty1

替换为

- 1:2345:respawn:/sbin/autologin tty1 login -f root

简要说明：

首先，要将 file descriptor 和某 tty 对应好,0</dev/\$1 ,相当于 0</dev/tty1 表示由 tty1 读入 stdin,1>/dev/\$1,相当于 1>/dev/tty1 表示 stdout 输出到 tty1,2>&1,表示 stderr 也加到 stdout。然后再将左移(shift)参数一次（默认为一次，shift 后面没带参数），参数列表就成了\$1->deleted, \$2->\$1, \$3->\$2，然后由 exec 执行所有的参数：login -f username。login 命令的参数-f 表示不需验证用户的口令，用户是预验证的。

重启动，会发现系统不会再要求登录了，直接进入 xfce 桌面。那么现在，我们基本完成了磁盘系统的制作了。

## 制作 ISO 光盘镜像文件

现在，我们需要把这个磁盘上的系统转换为 LiveCD。我们使用红旗网站上提供的 livecd 工具以及启动内核。

首先安装 live 内核：

- rpm -i kernel-live 2.6.24-4\_1.i386.rpm

也可以使用自己编译的内核来作为 LiveCD 的内核，但是要注意：Linux 的 LiveCD 功能是源自于 Linux Live Script (<http://www.linux-live.org>) 这个项目的，此项目目前最多支持到是 2.6.26.6 内核，更高版本的内核目前没有 lzma 压缩文件系统的补丁支持，因此不能使用。此网站也提供了一个功能更全面的预编译内核，下载地址在

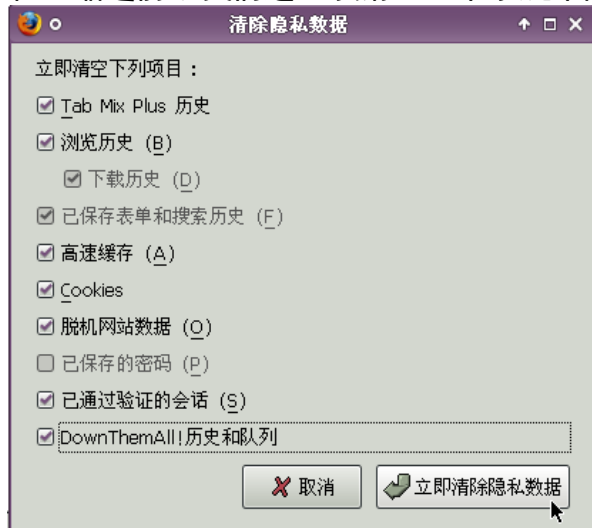
- <ftp://ftp.slax.org/Linux-Live/kernels/2.6.27.6/linux-2.6.27.6-i486-1.tgz>

安装好内核之后，把 livecd 工具解压到/tmp/live 目录，然后运行目录中的 install 脚本。注意：必需放在/tmp 目录，否则以后此工具也可能被打包到光盘中。

准备工作完成后，接下来开始制作 iso 文件。第一步是用 lzma 工具压缩几乎整个系统，当然不是压缩所有的文件，具体的内容，可以查看/tmp/live 目录下的.config 文件

- LIVECDNAME 定义 iso 文件的名字
- RAMOSIZE 定义 RAMDISK 的容量。此 RAMDISK 用于启动时加载内核及驱动。默认容量是 6666K 字节，如果编译的驱动很多，可能会因 RAMDISK 装不下而启动失败。调整此参数，需要同时修改几个文件：
  - .config
  - ./cd-root/boot/isolinux.cfg
  - ./cd-root/boot/syslinux.cfg
  - ./cd-root/boot/DOS/config
- MKMOD 定义了哪些目录将要被压缩进光盘中，不要随意修改此目录
- EXCLUDE 定义了哪些目录或文件不会被压缩，我们把此项修改为
  - EXCLUDE="/etc/fstab /etc/mtab /var/run /var/lock /var/lib/rpm /root/.bash\_history /root/.recently-used.xbel"

在压缩之前，我们还应该清理一下系统中的垃圾，比如 Firefox 所有的缓存、Cookie、上网痕迹：



最后开始压缩

- cd /tmp/live
- ./build

此时系统对几乎整个系统进行压缩，将花费相当长的时间。如果采用的是红旗网站提供的 livecd 制作工具，那么压缩之后，接着会自动生成 iso。而如果采用 Linux Live Script 官方工具，我们还需要执行命令

- cd /tmp/live\_data\_xxxx/RedFlag-LinuxCD

- `./make_iso`

这样才会生成 iso 文件。

生成的速度相对比较快。然后把 iso 复制到外界主机中：

- `scp live.iso root@10.0.2.2:/root`

至此，我们的 LiveCD 就制作出来了，接下来在外界主机中新建一个虚拟机用于测试。把 live.iso 作为虚拟光盘，启动测试虚拟机。

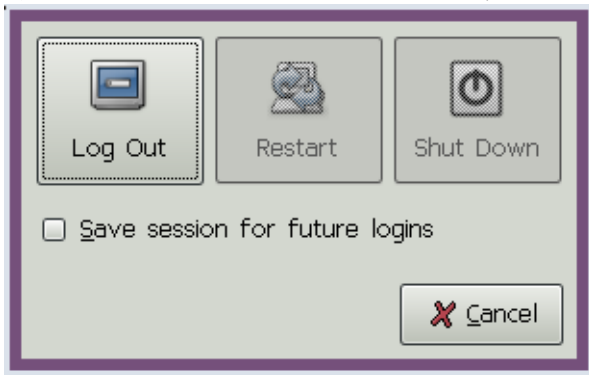
经过测试，我们的 cd 成功的进入了图形桌面，各个软件功能完全正常。

## 缺陷的处理

虽然光盘的功能都可用，但是还有一些小缺陷。

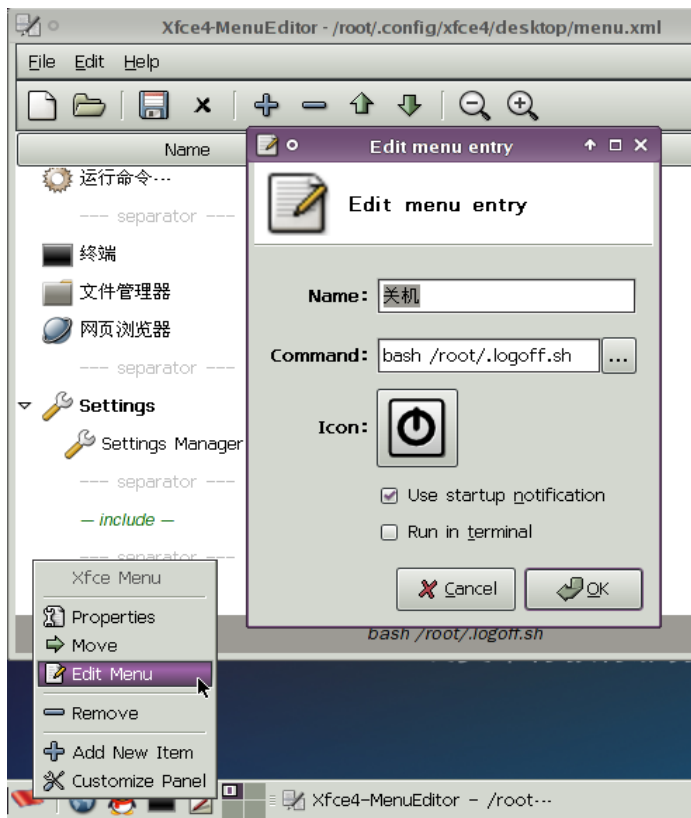
### 1、不能关机

最严重的问题是：桌面环境中不能关机。当选择了关机之后，显示的是这样的对话框：



其中，重新启动和关机都被禁用了。测试发现，在命令行中其实是可以关机和重新启动的。分析原因，发现这是由于缺少桌面管理器，xfce 环境是从 profile 中启动所造成的。

解决方法：修改主菜单，删除原来的关机项，添加新项目，指向一个自己编写的对话框。操作如下：



然后编写对话框脚本/root/.logoff.sh，内容如下：

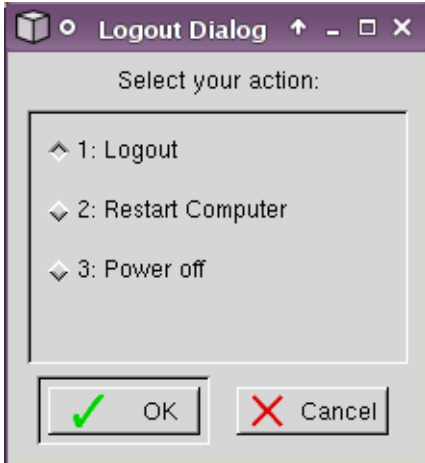
```
#!/bin/sh
TMP=/dev/shm/logout_select
Xdialog -title "Logout Dialog" \
  -radiolist "Select your action:" 15 30 3 \
  1 "Logout" on \
  2 "Restart Computer" off \
  3 "Power off" off \
  2>$TMP
ret=$?
case $ret in
  0)
    result=$(cat $TMP)
    case $result in
      1)
        killall sh
        ;;
      2)
        reboot -f
        ;;
      3)
        poweroff -f
        ;;
    esac
  *)

```

- echo "Cancel."
- ;;
- esac

此脚本需要 Xdialog，这就是我们编译 Xdialog 的原因了。

这样，新的关机对话框就出来了：



## 二、删除文件错误提示

在启动时，在 initrd 阶段就出现了一个删除文件错误：

```
* starting cdrom filesystem support
* starting squashfs support
* starting aufs support with brs=1
* starting linux filesystem support
* starting windows filesystem support
* creating /dev entries for block devices
rm: cannot remove '/dev/pty??*': No such file or directory
* starting USB support
* starting PCMCIA CardBus support
* looking for data directory (searching for livecd.sgn file)
* using RedFlag-LiveCD data from /mnt/hdc/RedFlag-LiveCD
* setting up directory for changes
* changes not used or not writable, using memory only
* setup union directory (using aufs)
* inserting all modules and creating live filesystem
```

经过分析，发现这是 LiveCD 制作工具带来的。解决办法是修改 livecd 工具，文件 initrd/liblinuxlive 第 730 行：

- rm /dev/pty??\* /dev/tty??\* # remove unneeded pty and tty devices

改成

- rm /dev/pty??\* /dev/tty??\* 2>/dev/null # remove unneeded pty and tty devices



### 三、touch 错误提示

另一个问题是在 Asianux 的 sysinint 阶段的错误：

```
INIT: version 2.86 booting
      Welcome to Asianux Workstation 3
      Press 'I' to enter interactive startup.
Setting clock (localtime): Fri Nov 21 16:45:27 CST 2008 [ OK ]
Starting udev: [ OK ]
Loading default keymap (us): [ OK ]
Setting hostname localhost.localdomain: [ OK ]
No devices found
Checking filesystems [ OK ]
Remounting root filesystem in read-write mode: [ OK ]
Mounting local filesystems: [ OK ]
Enabling local filesystem quotas: [ OK ]
Enabling /etc/fstab swaps: [ OK ]
INIT: Entering runlevel: 3
Entering non-interactive startup
Applying Intel CPU microcode update: [ OK ]
touch: cannot touch `/var/lock/subsys/microcode_ctl': No such file or directory
Starting sysstat: [ OK ]
Checking for hardware changes [ OK ]
touch: cannot touch `/var/lock/subsys/kudzu': No such file or directory
Bringing up loopback interface: [ OK ]
Bringing up interface eth0:
```

出现此问题的原因是：/var/lock 目录是一个 tmpfs 的挂载点，不可能在关机后存下任何东西。解决的办法是：修改 Asianux 的系统启动脚本/etc/rc.d/rc.sysinint，在 892 行处插入  
mkdir -p /var/lock/subsys

解决了这几个问题之后，系统就很完美了。

### 指导教师评语

指导教师：闻波(wenbob@gmail.com)

评语：光盘的定位比较实用，而且系统小巧快速，显得干净利落。该小组各个成员都有不少的 Linux 使用经验，分析处理问题时显得比较老道，往往能够迅速找到症结。

### 团队简介

队名：xfsh01

成员及分工

郝俊兵(队长) 规划、编译、安装、美工、问题诊断

姚俊健 配置、裁减、测试

郭文龙 配置、裁减、测试

团队邮箱：sekongsoft@126.com