

用VirtualBox在Windows-10上运行Motif编译

William June 2022

摘要

前不久写了一篇 “在 Windows10上运行 X-Window Motif”. 该文介绍如何用VcXsrv和Windows 10自身带有的扩充软件 WSL共同实现运行Motif软件. 测试完了以后发现这个方法有很多不足之处: 使用不太方便, 每次启动都要重新设置一些参数; 另外WSL和Windows 10之间交换数据和文件也比较麻烦. 本文介绍另一种方式: 在 Windows10上装VirtualBox, 然后运行虚拟Linux系统OpenSuSE. OpenSuSE本身就支持Motif. 经测试, 编译多个 “Motif Programming Manual” 书中提供的样例程序, 没有错误效果不错.

I. Windows10上装VirtualBox

目前Windows10装的是VirtualBox版本6.1.

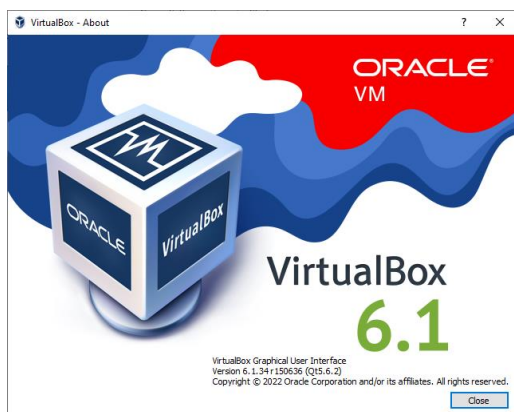


图1. VirtualBox V6.1

II. VirtualBox装虚拟Linux系统OpenSuSE

目前VirtualBox装的是openSUSE Linux, Leap-15.3 64-bit iso file.

openSUSE-Leap-15.3-DVD-x86_64-Current.iso

它可从以下link下载:

<http://download.opensuse.org/distribution/leap/15.3/iso/openSUSE-Leap-15.3-3-DVD-aarch64-Build38.1-Media.iso.mirrorlist>

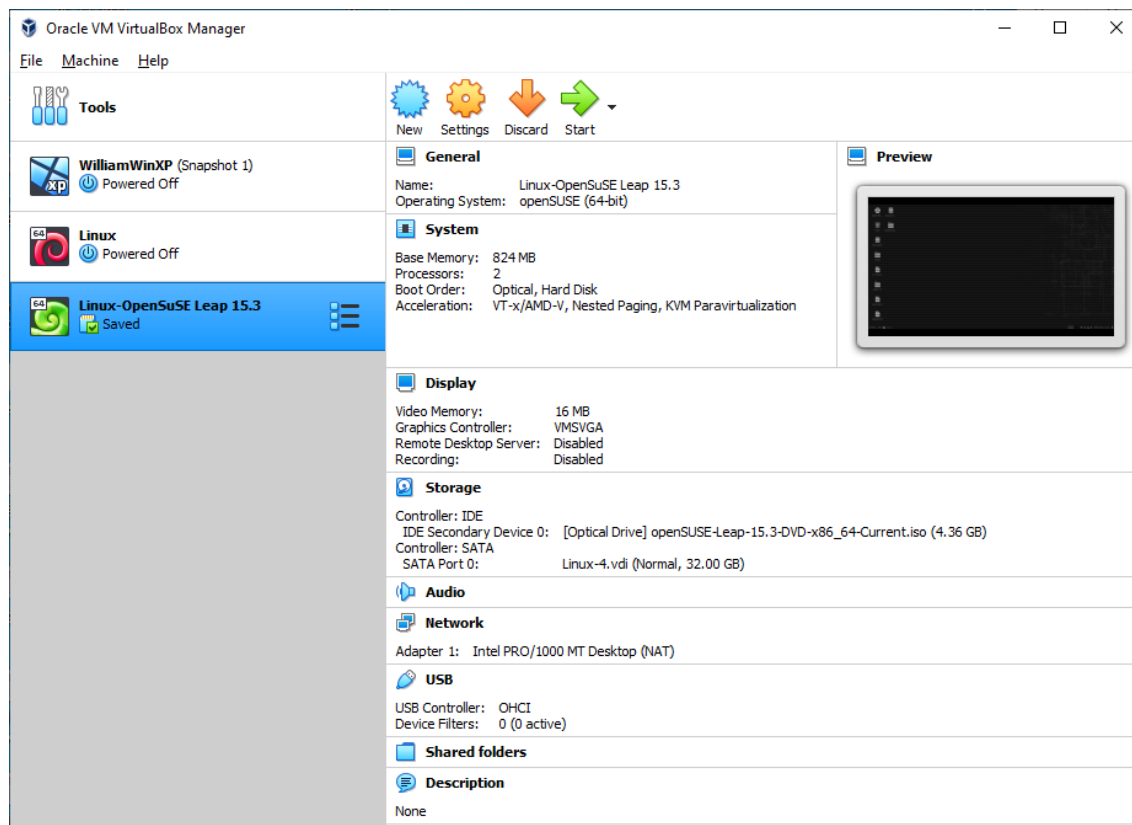


图2. 虚拟OpenSuSE Linux, Leap-15.3 64-bit

2.1 如何装虚拟Linux系统OpenSuSE

原理上装虚拟Linux系统系统是容易的, 但有几个关键点不能搞错. 以下描述几个主要关键点. 详细的可以参考附录, 那里有每一步的屏幕截图可参考.

2.1.1 按 “New” 开始装新的虚拟系统.

在图2上边有一个蓝色的按钮” New”, 由此开始装新的虚拟系统.

2.1.2 关于设置 “Storage”

设置 “Storage” 需要建立两个设备:

- 1) 把 CD-Drive 设置成下载的OpenSuSE ISO文件;
- 2) 建立一个可以扩充的虚拟硬盘, 初始容量可以设置为32 GB.

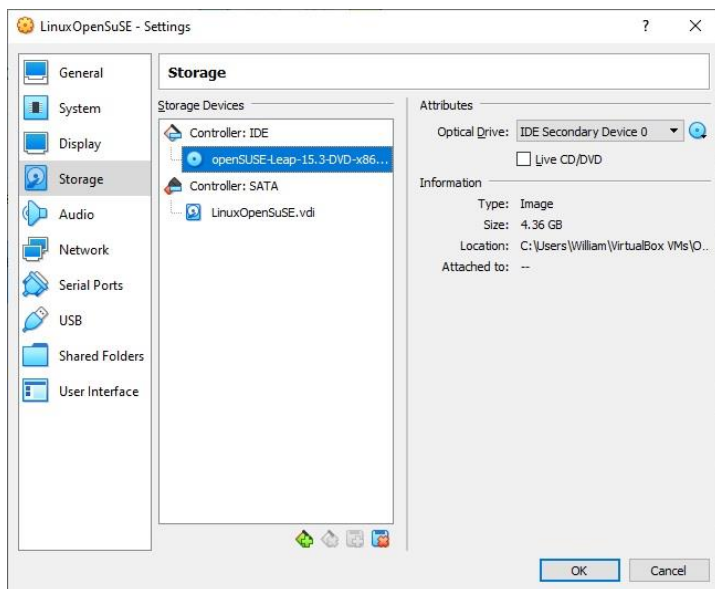


图3. 设置 “Storage”：需要建立两个设备

2.1.3 在装置操作系统的过程中选择 Graphics Interface

“Settings” 完了以后就可以按 Start 开始安装操作系统. 在装操作系统的过程中会问你：要选择何种界面？这时应该选择Desktop with Xfce. 曾经试过其他两种, 都因缺少部分软件不能进行到底.

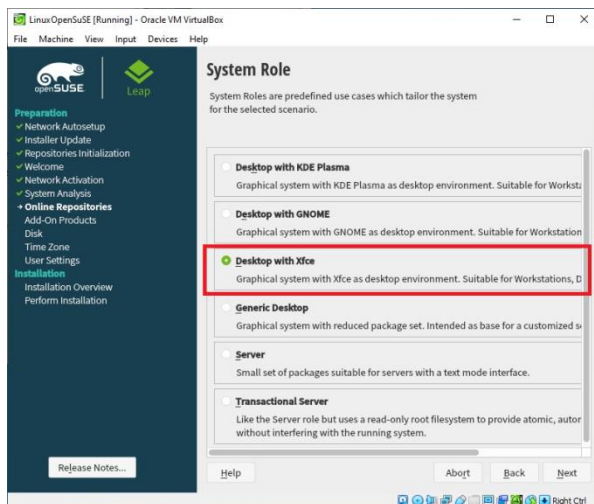


图4. 选择Desktop with Xfce.

III. 虚拟OpenSuSE如何与Windows 10共享文件

最直观想到的方式就是通过USB flash memory来交换两个不同操作系统的信息。但是实际操作并不理想，虚拟OpenSuSE很难Mount USB flash memory，有时接通有时接不通，很不可靠。经查找，原来VirtualBox本身就提供虚拟系统与Windows 10共享文件的方法，非常方便。这里简单描述一下：

VM VirtualBox上边的菜单中有 Devices。在这Devices以下，可以看到有多种共享文件的方法。如Share Folders，Drag and Drop等等。我选用的是 Drag and Drop。我觉得这个方法非常方便，可以随便把文件在两个不同的系统中拉来拉去。也就是随意拷贝了，非常方便。见图3。

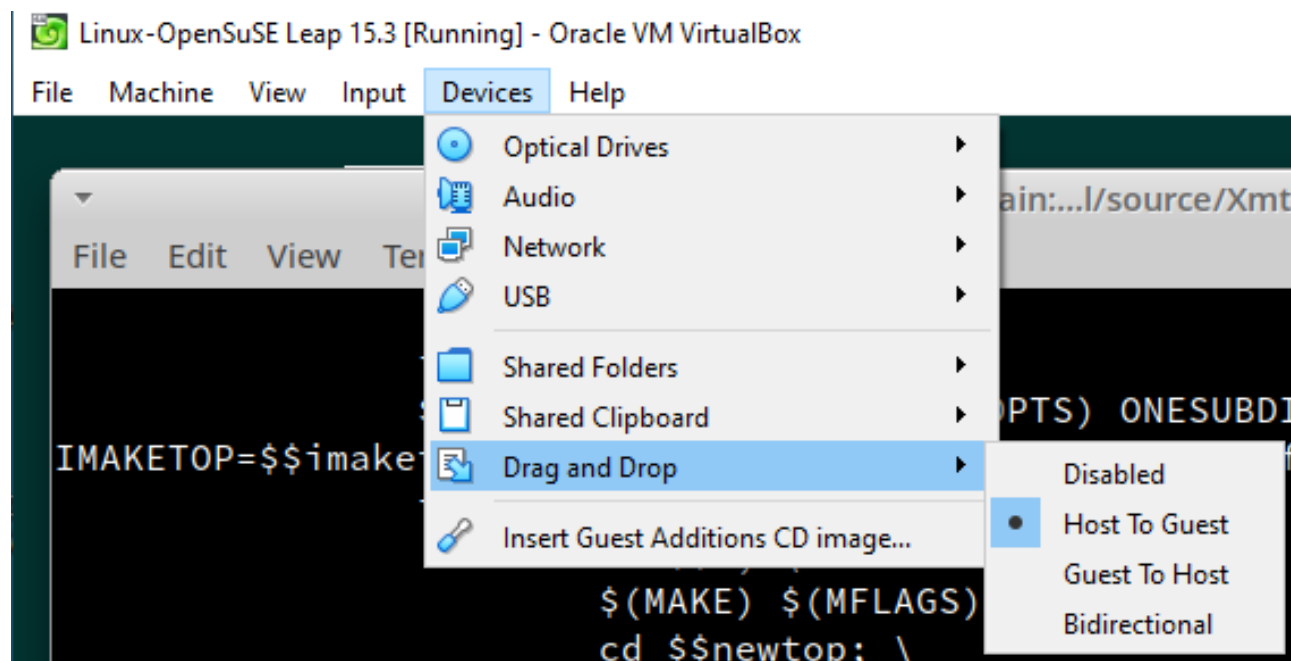


图5. 虚拟OpenSuSE如何与Windows 10共享文件

III. Install Motif developing packages

3.1 虚拟Linux系统OpenSuSE Leap-15.3系统能运行 Motif

虚拟Linux系统OpenSuSE系统装完以后就能运行 Motif 的现有程序, 如 xclock等. 但是如要编译建立你自己的程序, 还必须安装C语言编辑器和两个 motif有关的 software packages:

```
motif;  
motif-devel.
```

这两个packages将装载许多 X-Window 的 xxx.h 文件和 Library, 装载完成以后才能用以下命令编译:

```
cc -o first first.c -lXm -lXt -lX11 <CR>
```

3.2 Installing Packages gcc, motif and motif-devel

A) Terminal进入SuperUser模式

```
su - <CR>
```

这时候需要输入 Administrator Password. 如果在装机的时候定义自己就是Administrator, 那就是你自己的 password.

B) Installing gcc Package (它包含了 cc 编辑器)

```
zypper install gcc <CR>
```

C) Installing Motif Package

```
zypper install motif <CR>
```

D) Installing Motif Develop Package

```
zypper install motif-devel <CR>
```

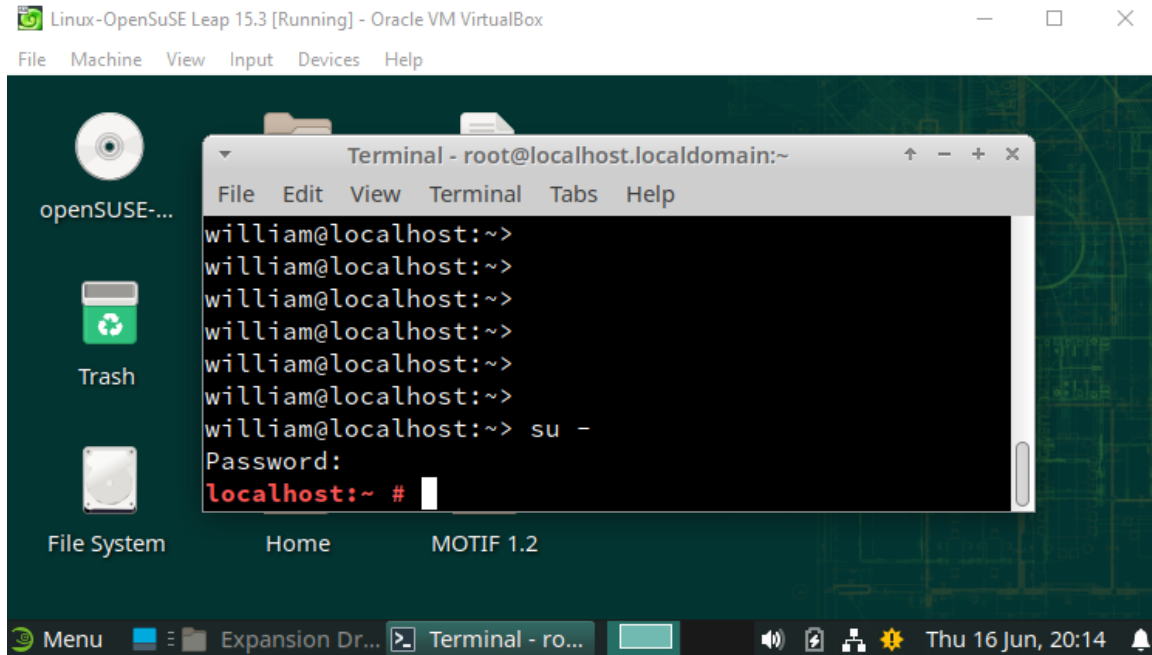


图6. 进入SuperUser模式

```

Checking for file conflicts: .....[done]
( 1/30) Installing: kbproto-devel-1.0.7-1.23.x86_64 .....[done]
( 2/30) Installing: libMrm4-2.3.4-3.3.1.x86_64 .....[done]
( 3/30) Installing: libbz2-devel-1.0.6-5.11.1.x86_64 .....[done]
( 4/30) Installing: libexpat-devel-2.2.5-3.6.1.x86_64 .....[done]
( 5/30) Installing: libxcb-composite0-1.13-3.5.1.x86_64 .....[done]
( 6/30) Installing: libxcb-damage0-1.13-3.5.1.x86_64 .....[done]
( 7/30) Installing: libxcb-dpms0-1.13-3.5.1.x86_64 .....[done]
( 8/30) Installing: libxcb-record0-1.13-3.5.1.x86_64 .....[done]
( 9/30) Installing: libxcb-res0-1.13-3.5.1.x86_64 .....[done]
(10/30) Installing: libxcb-screensaver0-1.13-3.5.1.x86_64 .....[done]
(11/30) Installing: libxcb-xf86dri0-1.13-3.5.1.x86_64 .....[done]
(12/30) Installing: libxcb-xtest0-1.13-3.5.1.x86_64 .....[done]
(13/30) Installing: libxcb-xvmc0-1.13-3.5.1.x86_64 .....[done]
(14/30) Installing: printproto-devel-1.0.5-1.22.x86_64 .....[done]
(15/30) Installing: pthread-stubs-devel-0.4-1.25.x86_64 .....[done]
(16/30) Installing: renderproto-devel-0.11.1-1.22.x86_64 .....[done]
(17/30) Installing: xproto-devel-7.0.31-1.22.x86_64 .....[done]
(18/30) Installing: zlib-devel-1.2.11-3.21.1.x86_64 .....[done]
(19/30) Installing: libUil4-2.3.4-3.3.1.x86_64 .....[done]
(20/30) Installing: libXau-devel-1.0.8-1.26.x86_64 .....[done]
(21/30) Installing: libICE-devel-1.0.9-1.25.x86_64 .....[done]
(22/30) Installing: freetype2-devel-2.10.1-4.8.1.x86_64 .....[done]
(23/30) Installing: libxcb-devel-1.13-3.5.1.x86_64 .....[done]
(24/30) Installing: libSM-devel-1.2.2-1.23.x86_64 .....[done]
(25/30) Installing: fontconfig-devel-2.12.6-4.3.1.x86_64 .....[done]
(26/30) Installing: libX11-devel-1.6.5-3.21.1.x86_64 .....[done]
(27/30) Installing: libXt-devel-1.1.5-2.24.x86_64 .....[done]
(28/30) Installing: libXrender-devel-0.9.10-1.30.x86_64 .....[done]
(29/30) Installing: libXft-devel-2.3.2-1.33.x86_64 .....[done]
(30/30) Installing: motif-devel-2.3.4-3.3.1.x86_64 .....[done]
localhost:~ # zypper install motif-devel
  
```

图7. zypper install motif-devel <CR>显示的结果


```
File Edit View Terminal Tabs Help
william@localhost: /usr/include> cd Xm
william@localhost: /usr/include/Xm> ls
AccColorT.h      Display.h      IconFileP.h   ProtocolsP.h   Text.h
AccTextT.h       DisplayP.h     IconG.h       PushBG.h       TextInP.h
ActivatableT.h   DragC.h        IconGP.h      PushBGP.h      TextOutP.h
ArrowBG.h        DragCP.h       IconH.h       PushB.h        TextP.h
ArrowBGP.h       DragDrop.h     IconHP.h      PushBP.h       TextSelP.h
ArrowB.h         DragIcon.h     JoinSideT.h   RCLayoutP.h    TextStrSoP.h
ArrowBP.h        DragIconP.h    LabelG.h      RCMenup.h      ToggleBG.h
AtomMgr.h        DragOverS.h    LabelGP.h     RepType.h      ToggleBGP.h
BaseClassP.h     DragOverSP.h   Label.h       RowColumn.h    ToggleB.h
BulletinB.h      DrawingA.h     LabelP.h      RowColumnP.h   ToggleBP.h
BulletinBP.h     DrawingAP.h    LayoutT.h     SashP.h        ToolTipCT.h
ButtonBox.h      DrawnB.h       List.h        Scale.h        ToolTipT.h
ButtonBoxP.h     DrawnBP.h      ListP.h       ScaleP.h       TraitP.h
CacheP.h         DrawP.h        MainW.h       Screen.h       Transfer.h
CareVisualT.h    DrawUtils.h    MainWP.h      ScreenP.h      TransferP.h
CascadeBG.h      DropDown.h     Manager.h     ScrollBar.h    TransferT.h
CascadeBGP.h     DropDownP.h    ManagerP.h    ScrollBarP.h   TransltnsP.h
CascadeB.h       DropSMgr.h     MenuProcP.h   ScrolledW.h    TravCont.h
CascadeBP.h      DropSMgrP.h    MenuShell.h   ScrolledWP.h   Tree.h
ClipWindowP.h   DropTrans.h    MenuShellP.h  ScrollFrameT.h TreeP.h
ColorObjP.h      DropTransP.h   MenuStateP.h  SelectioB.h    TxtPropCv.h
ColorP.h         Ext18List.h    MenuT.h       SelectioBP.h   UnhighlightT.h
Colors.h         Ext18ListP.h   MenuUtilP.h   SeparatoG.h    UnitTypeT.h
ColorSP.h        Ext.h          MessageB.h    SeparatoGP.h   VaSimpleP.h
Column.h         ExtObjectP.h   MessageBP.h   Separator.h    VendorSEP.h
ColumnP.h        ExtP.h         MultiList.h   SeparatorP.h   Vendors.h
ComboBox2.h      FileSB.h       MultiListP.h  ShellEP.h      VendorSP.h
ComboBox2P.h     FileSBP.h      MwmUtil.h     SlideC.h       VirtKeys.h
ComboBox.h       FontS.h        NavigatorT.h   SlideCP.h      VirtKeysP.h
ComboBoxP.h      FontSP.h       Notebook.h    SpecRenderT.h  XmAll.h
Command.h        Form.h         NotebookP.h   SpinB.h        Xmfuns.h
CommandP.h       FormP.h        Outline.h     SpinBP.h       Xm.h
Container.h      Frame.h        OutlineP.h    SSpinB.h       XmIm.h
ContainerP.h     FrameP.h       Paned.h       SSpinBP.h      xmlist.h
ContainerT.h     Gadget.h      PanedP.h      TabBox.h       XmosP.h
ContainerT.h     GadgetP.h     PanedWP.h     TabBoxP.h      XmosP.h
```

图8. Motif Package新装的 *.h 文件

```
File Edit View Terminal Tabs Help
william@localhost: /usr/lib64>
william@localhost: /usr/lib64>
william@localhost: /usr/lib64> ls *X* -all
lrwxrwxrwx 1 root root      16 Mar  6 2021 libGLX_indirect.so.0 -> libGLX_mesa.so.0
lrwxrwxrwx 1 root root      16 Mar  6 2021 libGLX_mesa.so -> libGLX_mesa.so.0
lrwxrwxrwx 1 root root      20 Mar  6 2021 libGLX_mesa.so.0 -> libGLX_mesa.so.0.0.0
-rwxr-xr-x 1 root root 504792 Mar  6 2021 libGLX_mesa.so.0.0.0
lrwxrwxrwx 1 root root      15 Mar  5 2021 libGLX.so.0 -> libGLX.so.0.0.0
-rwxr-xr-x 1 root root 133632 Mar  5 2021 libGLX.so.0.0.0
lrwxrwxrwx 1 root root      22 Dec 10 2020 libQt5XcbQpa.so.5 -> libQt5XcbQpa.so.5.12.7
lrwxrwxrwx 1 root root      22 Dec 10 2020 libQt5XcbQpa.so.5.12 -> libQt5XcbQpa.so.5.12.7
-rwxr-xr-x 1 root root 1562312 Dec 10 2020 libQt5XcbQpa.so.5.12.7
lrwxrwxrwx 1 root root      27 Jun  9 2020 libQt5XmlPatterns.so.5 -> libQt5XmlPatterns.so.5.12.7
lrwxrwxrwx 1 root root      27 Jun  9 2020 libQt5XmlPatterns.so.5.12 -> libQt5XmlPatterns.so.5.12.7
-rwxr-xr-x 1 root root 3847376 Jun  9 2020 libQt5XmlPatterns.so.5.12.7
lrwxrwxrwx 1 root root      15 Jun  1 2021 libX11.so -> libX11.so.6.3.0
lrwxrwxrwx 1 root root      15 Mar  1 2021 libX11.so.6 -> libX11.so.6.3.0
-rwxr-xr-x 1 root root 1314376 Mar  1 2021 libX11.so.6.3.0
lrwxrwxrwx 1 root root      19 Jun  1 2021 libX11-xcb.so -> libX11-xcb.so.1.0.0
lrwxrwxrwx 1 root root      19 Mar  1 2021 libX11-xcb.so.1 -> libX11-xcb.so.1.0.0
-rwxr-xr-x 1 root root 6000 Mar  1 2021 libX11-xcb.so.1.0.0
lrwxrwxrwx 1 root root      15 May 25 2018 libXau.so -> libXau.so.6.0.0
lrwxrwxrwx 1 root root      15 May 25 2018 libXau.so.6 -> libXau.so.6.0.0
-rwxr-xr-x 1 root root 14552 May 25 2018 libXau.so.6.0.0
lrwxrwxrwx 1 root root      16 Sep 30 2018 libXaw7.so.7 -> libXaw7.so.7.0.0
-rwxr-xr-x 1 root root 476920 Sep 30 2018 libXaw7.so.7.0.0
lrwxrwxrwx 1 root root      12 Sep 30 2018 libXaw.so.7 -> libXaw7.so.7
lrwxrwxrwx 1 root root      22 May 25 2018 libXcomposite.so.1 -> libXcomposite.so.1.0.0
-rwxr-xr-x 1 root root 10320 May 25 2018 libXcomposite.so.1.0.0
lrwxrwxrwx 1 root root      19 May 25 2018 libXcursor.so.1 -> libXcursor.so.1.0.2
-rwxr-xr-x 1 root root 43736 May 25 2018 libXcursor.so.1.0.2
lrwxrwxrwx 1 root root      19 May 25 2018 libXdamage.so.1 -> libXdamage.so.1.1.0
-rwxr-xr-x 1 root root 10336 May 25 2018 libXdamage.so.1.1.0
lrwxrwxrwx 1 root root      17 May 25 2018 libXdmcp.so.6 -> libXdmcp.so.6.0.0
-rwxr-xr-x 1 root root 22744 May 25 2018 libXdmcp.so.6.0.0
```

图9. Motif Package 新装的 Library 文件

IV. Motif 编程官方参考书 “Motif Programming Manual”

4.1 这本书在网上可以找到

这本书在网上可以找到两个版本：一个版本是PDF格式, 可以下载; 第2个版本是HTML格式, 阅读起来比较容易而且很容易把样例程序拷贝下来.

Download PDF:

<https://www.oreilly.com/openbook/motif/vol6a/Vol6a.pdf>

HTML Reading mode:

<https://www.oreilly.com/openbook/motif/vol6a/Vol6a.html/>

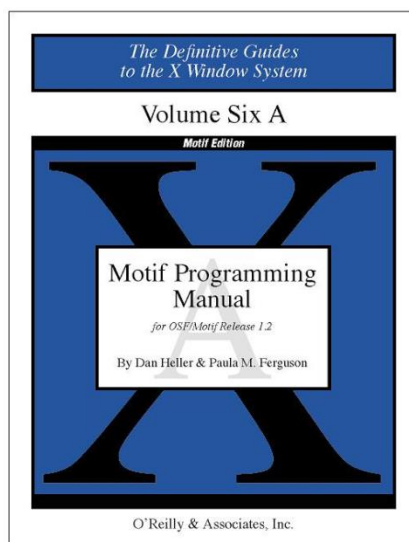


图10. Book Motif Programming Manual

4.2 简单测试程序 hello.c

```
/* hello.c -- initialize the toolkit using an application context and a
 * toplevel shell widget, then create a pushbutton that says Hello using
 * the varargs interface.
 */
#include <Xm/PushB.h>
```



```

main(argc, argv)
int argc;
char *argv[];
{
    Widget      toplevel, button;
    XtAppContext app;
    void        button_pushed();
    XmString     label;

    XtSetLanguageProc (NULL, NULL, NULL);

    toplevel = XtVaAppInitialize (&app, "Hello", NULL, 0,
        &argc, argv, NULL, NULL);

    label = XmStringCreateLocalized ("Push here to say hello");
    button = XtVaCreateManagedWidget ("pushme",
        xmPushButtonWidgetClass, toplevel,
        XmNlabelString, label,
        NULL);
    XmStringFree (label);
    XtAddCallback (button, XmNactivateCallback, button_pushed, NULL);

    XtRealizeWidget (toplevel);
    XtAppMainLoop (app);
}

void
button_pushed(widget, client_data, call_data)
Widget widget;
XtPointer client_data;
XtPointer call_data;
{
    printf ("Hello Yourself!0);
}

```



图11. Hello.c

4.3 编译命令

```
cc -O -o hello hello.c -lXm -lXt -lX11
```

4.4 Run Program

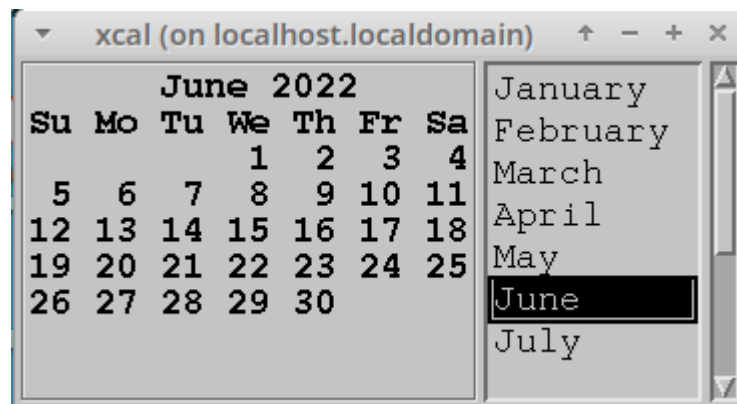
```
./hello <CR>
```

4.5 Add Fallbacks for Resources

It is generally a good idea to provide fallbacks for resources that are essential to the operation of your application. An example of how fallback resources can be used by an application is shown in the following code fragment:

```
String fallbacks[] = {
    "Demos*background: grey",
    "Demos*XmList.fontList: -*-courier-medium-r---12-*",
    "Demos*XmText.fontList: -*-courier-medium-r---12-*",
    /* list the rest of the app-defaults resources here ... */
    NULL
};
...
toplevel = XtVaAppInitialize (&app, "Demos", NULL, 0,
    &argc, argv, fallbacks, NULL);
...
```

4.6 测试程序日历 xcal.c



```
/* xcal.c -- display a monthly calendar. The month displayed is a
 * single Label widget whose text is generated from the output of
 * the "cal" program found on any UNIX machine. popen() is used
 * to run the program and read its output. Although this is an
 * inefficient method for getting the output of a separate program,
 * it suffices for demonstration purposes. A List widget displays
 * the months and the user can provide the year as argv[1].
 */
#include <stdio.h>
#include <X11/Xos.h>
#include <Xm/List.h>
#include <Xm/Frame.h>
#include <Xm/LabelG.h>
```

```

#include <Xm/RowColumn.h>
#include <Xm/Separator.h>

int year;
XmStringTable ArgvToXmStringTable();
void FreeXmStringTable();

char *months[] = {
    "January", "February", "March", "April", "May", "June",
    "July", "August", "September", "October", "November", "December"
};

main(argc, argv)
int argc;
char *argv[];
{
    Widget toplevel, frame, rowcol, label, w;
    XtAppContext app;
    Display *dpy;
    extern void set_month();
    XmFontList fontlist;
    XmFontListEntry entry;
    XFontStruct *font;
    XmStringTable str;
    int month_no;

    XtSetLanguageProc (NULL, NULL, NULL);

    toplevel = XtVaAppInitialize (&app, "Demos", NULL, 0,
        &argc, argv, NULL, NULL);

    /* Create a fontlist based on the fonts we're using. These are the
     * fonts that are going to be hardcoded in the Label and List widgets.
     */
    dpy = XtDisplay (toplevel);
    font = XLoadQueryFont (dpy, "-*-courier-bold-r-*--18-*");
    entry = XmFontListEntryCreate ("tag1", XmFONT_IS_FONT, font);
    fontlist = XmFontListAppendEntry (NULL, entry);
    font = XLoadQueryFont (dpy, "-*-courier-medium-r-*--18-*");
    entry = XmFontListEntryCreate ("tag2", XmFONT_IS_FONT, font);
    fontlist = XmFontListAppendEntry (fontlist, entry);
    XtFree (entry);

    if (argc > 1) {
        month_no = 1;
        year = atoi (argv[1]);
    }
    else {

```

```

    long time(), t = time(0);
    struct tm *today = localtime (&t);
    year = 1900 + today->tm_year;
    month_no = today->tm_mon+1;
}

/* The RowColumn is the general layout manager for the application.
 * It contains two children: a Label gadget that displays the calendar
 * month, and a ScrolledList to allow the user to change the month.
 */
rowcol = XtVaCreateWidget ("rowcol",
    xmRowColumnWidgetClass, toplevel,
    XmNorientation, XmHORIZONTAL,
    NULL);

/* enclose the month in a Frame for decoration. */
frame = XtVaCreateManagedWidget ("frame",
    xmFrameWidgetClass, rowcol, NULL);
label = XtVaCreateManagedWidget ("month",
    xmLabelGadgetClass, frame,
    XmNalignment, XmALIGNMENT_BEGINNING,
    XmNfontList, fontlist,
    NULL);

/* create a list of month names */
strs = ArgvToXmStringTable (XtNumber (months), months);
w = XmCreateScrolledList (rowcol, "list", NULL, 0);
XtVaSetValues (w,
    XmNitems, strs,
    XmNitemCount, XtNumber(months),
    XmNfontList, fontlist,
    NULL);
FreeXmStringTable (strs);
XmFontListFree (fontlist);
XtAddCallback (w, XmNbrowseSelectionCallback, set_month, label);
XtManageChild (w);
XmListSelectPos (w, month_no, True); /* initialize month */

XtManageChild (rowcol);
XtRealizeWidget (toplevel);
XtAppMainLoop (app);
}

/* callback function for the List widget -- change the month */
void
set_month(w, client_data, call_data)
Widget w;
XtPointer client_data;

```

```

XtPointer call_data;
{
    register FILE *pp;
    extern FILE *popen();
    char text[BUFSIZ];
    register char *p = text;
    XmString str;
    Widget label = (Widget) client_data;
    XmListCallbackStruct *list_cbs =
        (XmListCallbackStruct *) call_data;

    /* Ask UNIX to execute the "cal" command and read its output */
    sprintf (text, "cal %d %d", list_cbs->item_position, year);
    if (!(pp = popen (text, "r"))) {
        perror (text);
        return;
    }
    *p = 0;
    while (fgets (p, sizeof (text) - strlen (text), pp))
        p += strlen (p);
    pclose (pp);

    /* display the month using the "tag1" font from the
     * Label gadget's XmNfontList.
     */
    str = XmStringCreateLtoR (text, "tag1");
    XtVaSetValues (label, XmNlabelString, str, NULL);
    XmStringFree (str);
}

/* Convert an array of string to an array of compound strings */
XmStringTable
ArgvToXmStringTable(argc, argv)
int argc;
char **argv;
{
    XmStringTable new =
        (XmStringTable) XtMalloc ((argc+1) * sizeof (XmString));

    if (!new)
        return (XmStringTable) NULL;

    new[argc] = 0;
    while (--argc >= 0)
        new[argc] = XmStringCreate (argv[argc], "tag2");
    return new;
}

```

```

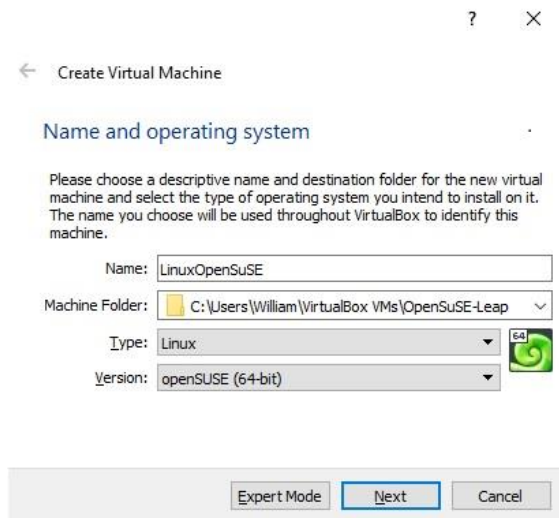
/* Free the table created by ArgvToXmStringTable() */
void
FreeXmStringTable(argv)
XmStringTable argv;
{
    register int i;

    if (!argv)
        return;
    for (i = 0; argv[i]; i++)
        XmStringFree (argv[i]);
    XtFree (argv);
}

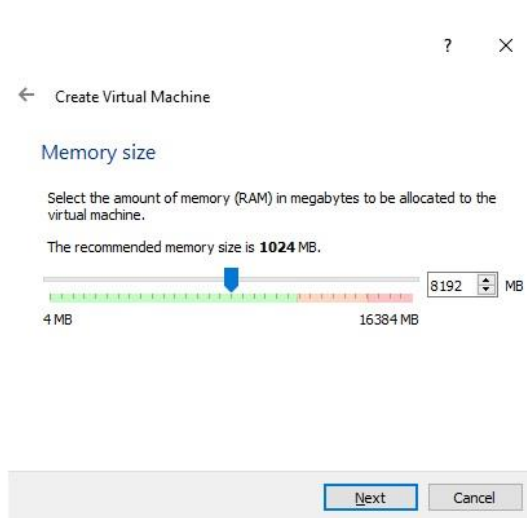
```

Appendix

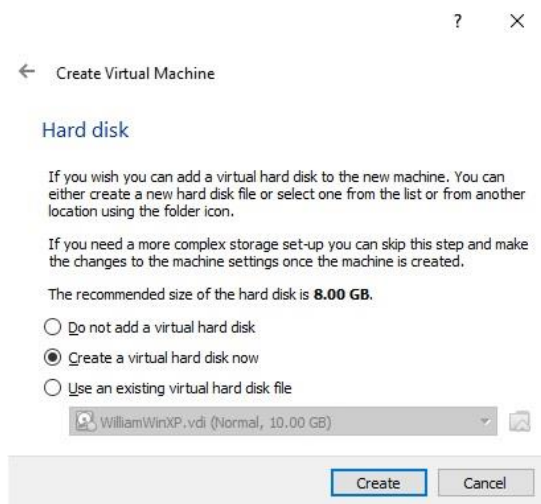
装虚拟Linux系统每一步的屏幕截图。



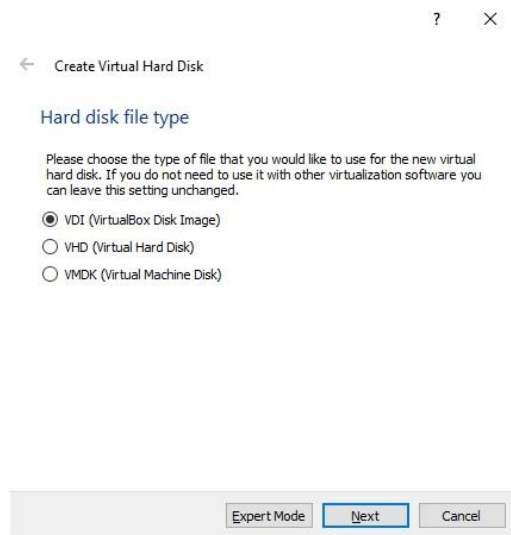
图A-1. 设置新系统名



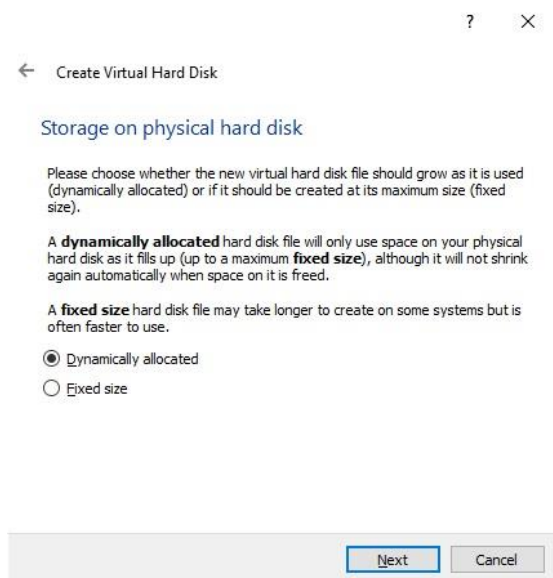
图A-2. 设置内存是8GB



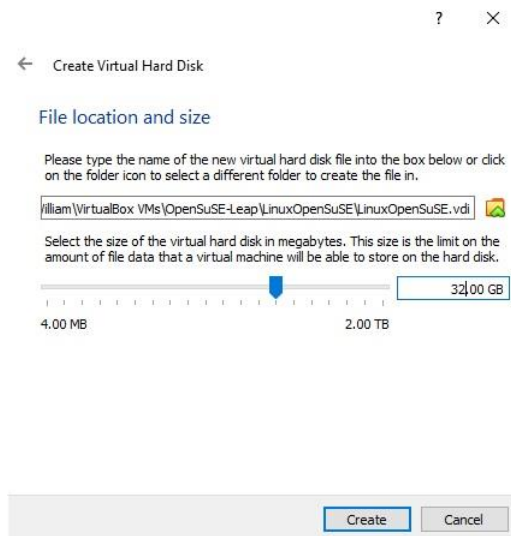
图A-3. 建立一个新硬盘



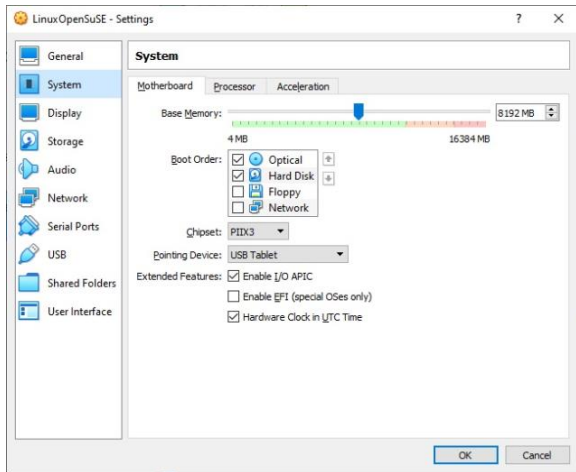
图A-4. 选择硬盘的格式是VDI



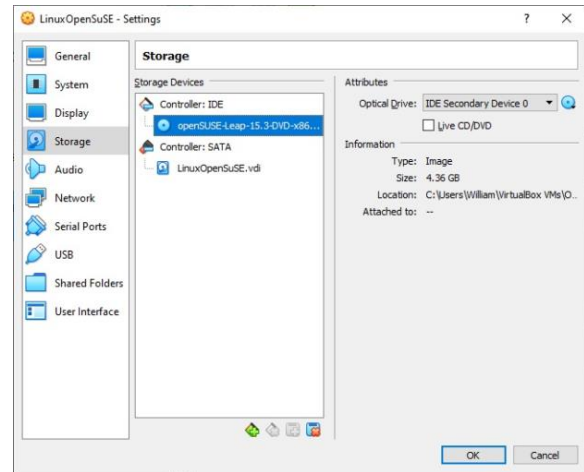
图A-5. 新硬盘可动态增加容量



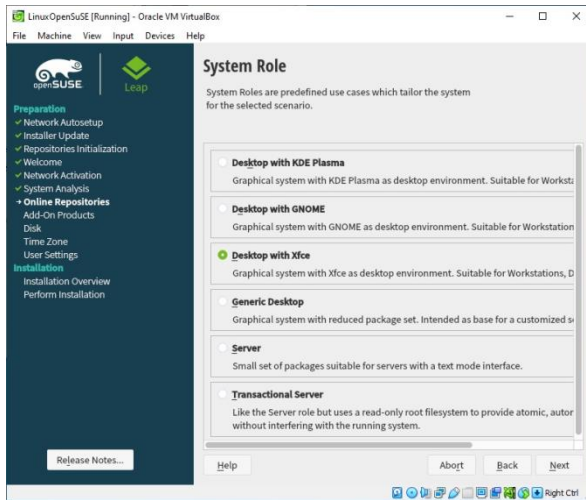
图A-6. 硬盘的初始容量是32GB



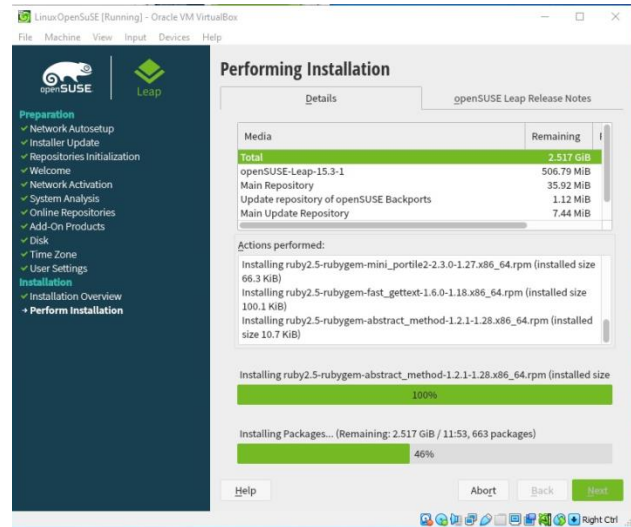
图A-7. 主板Chipset:PIIX3



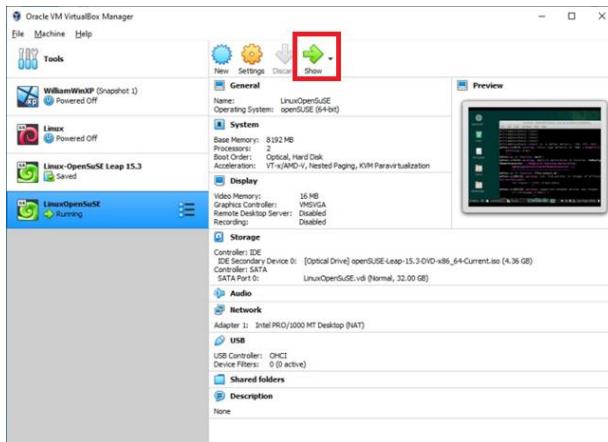
图A-8. 把 CD设置成下载ISO文件



图A-9. 选择Desktop with Xfce



图A-10. 开始安装操作系统



图A-11. "Start" 启动SuSE