

第28章 实用程序和工具

本节介绍一些小程序。利用这些程序可以更加有效地使用系统并使你的工作更实用。

28.1 实用程序

本节所描述的一些程序功能虽不太重要，但是却可以使你工作得更实用轻松。有些我们曾经提过，如 `xclock` 与 `xcalc`。但在这里将更加系统地描述这些功能。我们将看一些可以显示目前机器负载和告诉你有邮件送达等功能的新程序。

1. 一个模拟或数字时钟——`xclock`

`xclock` 有下列命令行选项：

- 指定窗口初始大小与位置：用 `-geometry geomspec`。
- 设定背景色：用 `-bg colour`。
- 设定前景色：用 `-fg colour`。
- 设定指针的颜色：用 `-hd colour`。
- 指针边缘高亮度显示：用 `-hl colour`，以高亮度显示 `colour` 颜色为指针的边缘。
- 使用数字时钟：`-digital`，告诉 `xclock` 使用一个 24 小时的数字时钟，以替换缺省的模拟型始终。
- 设定“时钟-滴答”频率：`-update num`，使时钟每 `num` 秒更新显示一次。处于指针状态下，若 `num` 小于 30 秒则以一个钻石形秒针每 `num` 秒移动一次。（缺省值为 60 秒）
- 设定半小时钟声：`-chime`，每半小时整使终端机铃响一次，每一个小时整铃响两次。

2. 桌上型计算器——`xcalc`

`xcalc` 的命令行选项如下：

(1) 指定窗口初始大小与位置命令：`-geometry geomspec`。

设定桌面颜色的选项为：

- 设定背景色：`-bg colour`。
- 设定前景色：`-fg colour`。

(2) 其他选项：

• 指定计算尺模式：`-analog`，执行程序模拟一个计算尺替换电子式计算器。这实在很少用，但一些更进一步的指令包含在下面，因为它不包含在联机帮助内。

• 指定 HP-10C 型计算器：`-rpn`（代表逆波兰记数法），告诉 `xcalc` 模拟 HP-10C 型计算器。

可以使用鼠标按钮操作计算尺。当鼠标指针进入窗口内，鼠标指针变成手型光标，用鼠标按钮操作计算尺有如下操作：

- 定位滑尺 - 左端：移动鼠标指针进入滑尺区，定位至你想要的地方，按左按钮，滑尺左端跳至鼠标指针位置。
- 定位滑尺 - 右端：与设定左端一样，但按右按钮。
- 定位计算尺的光标：定位鼠标指针至你想要的尺体位置，并按左按钮：尺的光标跳至鼠标指针位置。
- 滑动滑动条：定位光标至滑尺，按下中间按钮，并保持按着。拖动滑尺到你想要的位置，然

后释放按钮。

- 将尺的长度变为两倍: 在尺体上按中间按钮。
- 将尺的长度变为一半: 在尺体上按右按钮。

3. 显示机器的负载平均——xload

xload 可以显示系统平均负载的柱状图，它会定期地更新。

4. 邮件通知程序——xbiff

xbiff 是一个监视邮件文件并通知何时邮件到达的小程序。它显示一个邮箱的图案。当没有新邮件时邮箱的旗子是向下的，当有新邮件进来时，xbiff 响铃，竖起旗子，并使图案变成反相显示。

你可以在 xbiff 窗口上按任何鼠标按钮，强迫旗子放下。

xbiff 接受一般对颜色的命令行选项 (-bg、-fg、-bd) 与对窗口特性的命令行选项 (-display、-geometry、-bw)。其他选项有：

- 指定检查邮件频率：-update num，告诉 xbiff 每 num 秒检查邮箱一次，看是否有新的邮件到达(缺省值为 60 秒)。
- 指定一个特定邮件文件：-file filename，使 xbiff 检查在 filename 内的邮件，替换缺省名称的文件。缺省名称为 /usr/spool/mail/username，其中username为你的登录名称。

xbiff 特有的 -file 选项，对网络上的邮件集中处理和你的邮箱由另外的机器处理等两方面非常有用。下面的命令让你执行某个邮件机器（比如说mars）上的 xbiff程序，查看你的邮箱，并将显示送回你自己的机器 venus：

```
rsh mars xbiff -file /var/spool/mail/smith -display venus:0 &
```

28.2 保存、显示和打印屏幕图像

X是一个图形系统，你可以利用它在窗口内显示文字与图形。X窗口的用户会经常想要捕捉屏幕上的某些图像，以便稍后可以重新显示它，或送至硬拷贝设备打印。下面将描述这些功能。

1. 保存窗口的图像——xwd

xwd 把窗口的图像保存到一个文件中。这个文件稍后可以被其他程序处理（例如打印）。

有许多种不同的使用 xwd的方法。你可以明确地指定一个输出文件（使用命令行选项 -out name），或者使用 xwd 打印图像到标准输出。你也可以明确地指定想打印的窗口，或可以让 xwd 提示你一下。

让我们举最简单的一个例子：在 xterm 下使用下列命令启动程序

```
xwd > outfile
```

只要 xwd 启动，光标便会变成“十字线”状。移动鼠标指针进入你想要打印的窗口，并按任何按钮：xwd 响一次铃表示它已经开始记录窗口图像，且在它完成时响两次铃。然后光标恢复正常。

2. 如何指定被打印的窗口

有两种方法可以通过命令行选项告诉 xwd 要打印哪一个窗口：

- -root：打印根窗口。例如，想得到屏幕的一个完整图片，使用命令：

```
xwd -root > screenpic
```

- -id win-id：打印窗口识别码为 win-id 的窗口。（每个由X服务程序建立的窗口皆有一个唯一的窗口识别码——它只是一个识别窗口的号码。）

使用这些选项的好处是你不必使用鼠标去指定你感兴趣的窗口。因为有时候在打印的同时，鼠标必须出现在某个特定的地方执行某项特定的功能，例如：呼叫一个下拉式菜单，或使一个命令按钮出现你想要的状态。举例来说：欲保存屏幕图像，必须用鼠标下拉 `uwn` 菜单，因而造成你没有办法同时使用鼠标去指定 `xwd` 的目标。下面是捕捉图像的步骤：

1) 在一个 `xterm` 窗口，键入命令

```
sleep 10.xwd -root > uwmfile
```

在捕捉图像之前，给你自己时间使屏幕进入到需要的状态。

2) 移动鼠标指针至背景窗口上。

3) 按下鼠标中间按钮，出现 `WindoeOps` 菜单，并保持按下状态。

4) 等一下，直到 `xwd` 响铃一次告诉你开始，完成则响铃二次。然后释放按钮。

2. 放大屏幕窗口图像——`Xmag`

`xmag` 可以快拍任何屏幕的图像，并以任意倍数放大它们。最简单的方法为采用交互方式指定你感兴趣的区域：

1) 启动程序（用 `xmag` 命令）。`xmag` 显示一个闪动的矩形外框。

2) 将矩形框放置到想要的地方。

3) 按任何按钮。`xmag` 快拍矩形框内的区域，并显示一个清新的窗口外框，里面正显示放大的快拍图像。

4) 仿照启动 `xclock` 后所采用的方法一样使用鼠标指针与按钮放置外框。例如，按鼠标左按钮将窗口放置在鼠标指针目前所在的位置。

5) `xmag` 在你刚放置的窗口内，重绘放大的图像。

你现在可以选择如下动作之一：

- 按 `Q` 或 `q` 或 `Ctrl-C`，接着指定 `xmag` 窗口内的选项，跳出 `xmag`。

- 按鼠标中间按钮或左按钮以移开目前放大的窗口，`xmag` 再次显示出它的闪烁矩形，所以你可以放大屏幕上另一个区域。

- 按下鼠标左按钮：`xmag` 显示出在鼠标指针下的像素坐标，即像素的号码（它是一个该像素颜色的内部表示法），以及像素的 `RGB` 值，亦即像素颜色的红，绿，蓝成分。当你移动鼠标指针，这些显示随之更新，直到你释放按钮。

程序记录快拍图像只是为了立即再显示——没有任何方法可以取得它以便打印图像至一个文件。

3. `xmag` 的命令行选项

缺省的操作方法有一些限制，你必须以交互方式指定被放大的区域，区域的形状与大小被固定，放大率都是 5。但事实上你可以使用命令行选项改变上述所有的特性：

- 指定不同的放大率：`-mag num`，放大图像 `num` 倍（`num` 须为整数）。例如：`-mag 2`，将产生一个为原来两倍宽与两倍高的图像。

- 指定大小与放大区域的位置：使用 `-source geomspec`。例如：

```
-source 300x100 + 450 + 762
```

`xmag` 不提示任何信息，但将直接跳到它显示放大窗口外框的步骤，假如 `geomspec` 只有位置部分，则大小部分缺省为 `64 × 64`。

- 仅指定放大区域的大小：使用 `-source geomspec`、`geomspec` 仅由大小部分组成，就像 `-source 300 × 100`

`xmag` 将显示一个闪烁的 `300 × 100` 的矩形外框，等你定位与按按钮。

4. 保存一部分屏幕的图像

xwd 仅对单一的完整窗口操作。假如想捕捉一个窗口的某部分或某些窗口，你必须采用两个阶段处理：

- 1) 使用 xmag 程序选择你要的区域，并在一个单一窗口（亦即 xmag 本身）显示它。
- 2) 用 xwd 打印 xmag 窗口至一个文件。假如你想打印与最初一样大小的图像，别忘了给

xmag 指定 -mag 1。

5. 显示先前打印的图像——xwud

xwud “反转储”一个先前曾经被打印至一个文件的图像，也就是再次将图像显示于屏幕上。缺省的作法是它从标准输入读入打印文件，所以反打印一个你想要的文件如下：

```
xwud < screenpic
```

xwud 显示闪烁的窗口外框，准备让你使用窗口管理器。当你已放置好它，图像即被显示。当然你可以利用窗口管理器移动窗口、重定大小、图标化等，就如同处理其他窗口一样。

注意 在联机帮助内提到的 -inverse 可能无法正常运作。

6. 打印先前打印的图像——xpr

xpr 能够把一个先前曾打印的图像转化为可送至硬拷贝机打印的格式。它支持各种打印机——PostScript、DEC LN03、LA100以及 IBM PP3812。由于缺省是写到标准输出，所以典型的用法应该是：

```
xpr -device ln03 < screenpic | lpr
```

xpr 接受数个选项，包括控制图像在纸上的位置，大小与配置和指定输出的打印机型号：

- 指定打印机型号：使用 -device type，这里的 type 是指 ln03、la100、ps (PostScript) 或 pp (指 PP3812)。
- 控制图像的大小：由于缺省 xpr 以最大的尺寸打印图像，你可以用 -width num 或 -height num 指定最大的宽度或高度，num 的单位为 inch (不必是整数)。
- 对图像标刻度：你可以使用 -scale num 对图像标刻度，num 为整数。作法与 xmag 的 -mag 选项类似，但大小可能扭曲，因为打印机可能有不同的分辨率，亦即每个 inch 有不同的点（像素）数，例如 -scale 2 表示每个屏幕像素被打印机印成 2×2 点的方形，所以假如你的屏幕为 75 dpi 且你的打印机为 300 dpi，打印图像仅为屏幕图像大小的一半 ($75 \times 2/300 = 0.5$)。
- 加图像标题：使用 -header string 或 -trailer string 分别将一个文字字符串印在图像的上面或下面。

7. 利用单一操作命令打印与打印一个窗口——xdpr

xdpr 是一个使用 xwd 打印窗口图像，接着使用 xpr 格式化打印图像，最后使用 lpr 输出图像的一个程序。它接受所有这三种程序的命令行选项。事实上 xdpr 是一个 shell 脚本，它将上述三种程序包装在一起以便使用。它真正的工作为检查命令行上的各个选项且传送它们给适当的程序。

28.3 使用X的应用程序

本节将介绍一些包含在 core 版本内的实用程序，它们的主要功能和窗口系统并无密切关系，但使用它们却可提供一个相当不错的用户界面，这些应用程序如下：

- Xedit ——一个以窗口为基础的一般目的文字编辑器。
- Xman ——一个联机帮助或系统文件的浏览工具。

- Xmh —— 一个 mh 邮件处理程序的用户窗口。

28.3.1 文字编辑器——Xedit

Xedit 是一个非常简单而具有窗口界面的文字编辑器。对 Xedit 所显示的选择框按按钮，你可以完成某些操作。但通过使用键盘你可以执行的更多操作，特别是使用控制字符。在这个程序内大多数以键盘为基础的功能由一个标准软件提供。

1. 启动与结束程序

假如你想编辑一个名叫 foobar 的文件，在 xterm 窗口键入命令：

```
xedit foobar &
```

(假如你不想编辑已存在的文件，只要键入 “ xedit & ” 即可。)

窗口分成三部分：

- 上面是命令菜单，具有 Quit、Save 等命令。
- 中间部分是信息窗口，xedit 在里面显示错误信息与状态，你也能把它当作一个抓取区来使用，可在里面键入一段你想剪贴到其他地方的文字。
- 下面较大的部分是编辑窗口，它显示你正在编辑或建立的文本。

当你想结束程序时，在 Quit 命令上按左按钮，假如你做了一些改变但未保存，则 xedit 会在信息窗口内输出警告信息：

```
Unsaved changes, Save them, or press Quit again.
```

2. 插入文字

首先确定鼠标指针在编辑窗口内，然后键入你要的文字（键盘上任意的打印字符），在你键入时，文字被直接地插入到窗口中。当键入时，你会注意到新插入的文字是不断地推送一个在它前面的一个小脱字符 (^) 光标，这个光标是插入点或简称为点。任何你键入的或贴入的文字都将在这个点的位置插入。插入点总是位于两个字符间，而不在一个字符上面（就像一个正常终端机光标的状况）。

在接近一行的尾端插入文字时，假如键入的字太长以致无法在这行放下时，xedit 将自动移至下一行。假如剪短该字使得可以适合它原来的行，它将再跳回上一行。

这个作用与 xedit 的段落定义息息相关：一个段落的终结是换行字符。折行仅作用于一个段落内，且能正确地处理特殊状况，例如：在移动一个单字到这行时造成这行必须折到下一行的状况等等。

(1) 特殊插入操作

下面有一些插入新行的特殊命令，各种命令稍为有些不同：

- 插入一个新行，且插入点移至新行：按 Return 键便可以插入一个新行，事实上就像插入任何打印字符一样。且假如你不使用 xedit 的折行，这是移至新行的正常方法。
- 插入新行，且对齐：按 LINEFEED 键插入一新行并移动插入点到下一行，但任何你键入的打印文字将与上一行的文字对齐(对键入表格等很有用)。
- 插入一新行，但不移动插入点：按 Ctrl-O。可以插入一个新行，但点仍留在同一行(当你想分开一行并加文字到被分成二行中的第一行尾端时使用)。

(2) 剪切与粘贴

就像在 xterm 中一样的，你可以选择并“剪”出一段文字。但是，有一个讨厌的问题，当选择字或行时，你必须迅速按二次或三次按钮——你不能在按钮按下或释放时暂停。

在插入点的地方贴入文字时，你可以像以前的按中间按钮，或按 Meta-Y 键。

3. 移动插入点

移动插入点最简单的方法是使用鼠标，移动窗口文字光标到任何你想要的地方，并按鼠标左按钮。插入点的 ^ 标志即跳到新位置。

但通常使用键盘移动插入点也很简单。当你正在编辑时，它会打断你的节奏，因为必须拿起鼠标，移动它，最后再次将你的手移回键盘。xedit 提供一次移动一个字符、单字、行或页的方式移动插入点，方法如下。

(1) 一次一字符移动插入点

这是移动点的最基本方式：

- 向前一个字符：按 Ctrl-f 或右箭头。
- 向后一个字符：按 Ctrl-b 或左箭头。
- 向上一个字符：按 Ctrl-p 或上箭头。
- 向下一个字符：按 Ctrl-n 或下箭头。

假如你接近一行的开头，按几次 Ctrl-b 后，你将回到前一行。按几次 Ctrl-f 后可回复至原位。假如你是在一页的最上面（最下面）一行，类似的效果会发生。移至前（或下）一行，将引起文字滚动使你移至的行能被看得见。

(2) 一次移动一个单字、一行或一个段落

这些是编辑文字时所须的自然移动功能。

- 向前一个单字：按 meta-F 或 meta-f。
- 向后一个单字：按 meta-B 或 meta-b。
- 移至行尾：按 Ctrl-E。
- 移到行的开头：按 Ctrl-A。
- 向前一个段落：按 meta-]。
- 向后一个段落：按 meta-[。

你可以看到一个规律：Ctrl-char 对单个字符操作。而 meta-char 对单字作相同的事。例如 Ctrl-F 向前一个字符，meta-F 向前一个字。

(3) 大量移动——以页与文件为单位

假如你想获得较大的跳越，就像在 xterm 里面一样，你可以使用滚动条。然而，滚动文字并不会移动插入点。当你键入或清除任何内容时，文字将自动地卷回插入点，让你可以看到你作了什么。

如果你想滚动显示内容且移动插入点，方法如下：

- 往前一页：按 Ctrl-V。
- 往后一页：按 meta-V。
- 到文件的开头：按 meta->。
- 到文件的尾端：按 meta-<。

4. 删除不想要的文字

有两种删除文字的技巧，删除只将文字移除，销毁则除了移除文字外，还将文字保存至“剪切缓冲区”，以便稍后你可以取回。销毁命令仅对较大单位的文字作用（最小单位为单字）。因为假如你删除一个字符，它就像没有销毁它一样很容易再打入。

在下面的叙述中，“下一个”意思是正好在插入点之后，“前一个”意思是正好在插入点之前。

(1) 删除字符

- 删除前一个字符：按 Delete 或 Backspace 或 Cth-H。
- 删除下一个字符：按 Ctl-D。

假如你在一行的开头按下 Delete 键，它将删除前一行的尾端的新行字符，而将两行合并成一行。

(2) 删除文字

- 删除下一个单字：按 Meta-D。
- 删除前一个单字：按 Meta-H 或 Shift-Meta-Delete Shift-Meta-Backspace。
- 从光标位置删除到行尾：按 Ctrl-K。
- 从光标位置删除到段尾：按 Meta-k。
- 删除目前选定的文字区块：按 Ctrl-W。

(3) 救回删除的文字；拷贝与移动文字

只要你曾经销毁某些文字，你就能按 Ctrl-Y 救回它，最近被删除的文字在插入点的地方被插入。但有点须注意：

- 只有最近被删除的文字可以被救回。你无法将一连串被删除的文字取回。
- 救回的文字是被插入到目前插入点的位置，不是该段文字原先被移除的地方。
- 若是需要的话你可以救回相同的文字许多次，即按 Ctrl-Y 并不影响缓冲区的内容。

基于“删除”动作的原理，你可以用它来移动或拷贝文字区域。

- 移动文字：先销毁它，再将点移动到你想重新放置的地方，最后再救回。
- 拷贝文字：先销毁它并救回它，使拷贝的来源不变；再将点移动到你想拷贝的地方，最后再一次救回。

5. 撤消改变

假如你作了某些改变(删除或删除或键入或贴入文字)，但稍后发现并不是想要的，你可以撤消它。在 undo 框里按左键，取消最近的改变。undo 本身也是一个改变，所以假如你再次按 undo，它将取消前一个 undo 的作用；你可以永远地像这样一直切换下去。

undo 仅对最近的改变有作用。假如你想更往回追溯，你可以利用 more 框连续地撤消更前面的改变。

6. 使用文件

前面曾提到在启动程序时你可以指定欲编辑的文件，事实上当程序执行时，你也能抓取文件。为达到这个目的你将会使用介于 load 与 undo 间的文字框，称为“文件名框”。

- 要把文字保存到一个文件，在 save 框上按左按钮；xedit 会将编辑窗口内的文字存到文件名为文件名框内文件名的文件；若没有名称，xedit 会在信息窗内发出 save : no filename specified——noting saved 因此在你再次在 save 上按左按钮之前你必须将鼠标指针移到文件名框并键入文件名。当它已经保存好文件 xedit 会发出确认信息。
- 编辑一个不同文件，在文件名框键入文件名，并在 load 上按左按钮。若它无法取用文件，xedit 会发出错误信息。
- 再目前的文本处插入一个文件：按下 meta-I，xedit 下拉一个小窗口。在上端的文字框，键入你想插入的文件名，并在 DoIt 上按左按钮。该文件的内容即插入目前的插入点位置。

7. 搜索指定的文字字符串

假如你想在你编辑文本中找出一个指定字符串出现的地方：

- 1) 在我们称为搜索字符串框的里面，在 Search >> 的右方键入字符串。
- 2) 在 Search >> 上按左按钮：插入点便会移至文本中下一个出现该字符串的地方。

搜索是由插入点开始，并且 xedit 缺省的搜索方向是向前，而不往后搜索。假如你要从插入点往回搜索，你可以在 << 上按左按钮。

8. 替换字符串

假如你想将出现许多次的一个字符串 (旧) 置换成为另一个字符串 (新)：

- 1) 先搜索出文本中第一次出现欲替换字符串的地方，如前一小节所述。
- 2) 在 all 框右方的替换字符串框内键入新字符串。
- 3) 在 replace 上按左按钮，旧字符串即被替换成新字符串，且插入点移到下一个出现旧字符串的地方。
- 4) 假如你也要替换它，再次在 replace 上按。假如你不想替换这个，但想改变它后面的，在 search >> 上按按钮直到你要改变的地方，即可以再次替换它。

在你往回移动文件时你无法执行替换，换言之你无法用简单的方法替换在插入点之前的字符串。

9. 杂项功能

- 重绘文本显示：按 Ctrl-L。
- 向前滚动一行：按 Ctrl-Z。
- 往回滚动一行：按 meta-Z。
- 跳至指定的行数：在信息窗口内，键入你要跳往的行数，用鼠标选择你正在打的文本，并在 jump 上按左按钮，插入点便会跳至指定行的开头。假如在屏幕上有应用程序的文本字段含有你要指定的行数数字，你可以利用它，而不必依赖 xedit 编辑信息窗口。

28.3.2 邮件/信息处理系统——xmh

xmh 是一个架在 mh 邮件/信息处理系统之上的 X 界面。当你启动它时，xmh 会建立一个图像的窗口。程序的窗口相关画面如编辑文本与管理窗口方框，均与 xedit 及 xman 十分类似，事实上是由相同的内部结构所提供的。正因为这样，且由于大多数描述是与程序的邮件功能有关而较少与 X 有关，我们不进一步讨论。假如你想知道如何使用程序，联机帮助中有一个简单但内容丰富的描述。

28.4 示例和游戏程序

MIT 发行的 core 版本提供了少数的示例程序和仅有的一个游戏程序。它们展现出某些窗口系统的威力，且能给人一种强烈的美好印象——特别是在彩色屏幕上。

28.4.1 找出通过随机迷宫的路径——maze

maze 在窗口中产生一个随机的迷宫，它会自动找出从入口通过迷宫到达出口的路径。它会追踪它走过的轨迹，当它从一个死巷中退出时则将轨迹消除。你可以用鼠标按钮启动、暂停、继续或停止程序，就如同在联机帮助中所描述的。maze 不提供颜色。

注意 中间按钮对暂停与重新启动的作用并不可靠。

28.4.2 担任鼠标指针的大眼睛——xeyes

xeyes 在窗口中绘出两个大眼睛，且它们永远看着鼠标指针。当鼠标指针移动时眼睛也随着调整，甚至当移动鼠标指针到两眼之间，它们会变成斗鸡眼！

你可以明确地给窗口的每个元素(瞳孔、背景、外框等)设定颜色。

注意 `xeyes` 会使你的系统执行速度变慢。

28.4.3 智慧盘游戏——puzzle

`puzzle` 是一种古老的游戏，有 15 个编号的小方块被一个 4×4 框架围住。你必须移动小方块使它们按照数字顺序排列。

你可以使用鼠标控制游戏：

- 启动游戏：在控制条的左上边框内按按钮可以重新打乱小方块。
- 移动小方块：将鼠标指针移到与空白位置相同的行或列的小方块上，按按钮以移动该小方块与它之前的所有小方块进入空的位置(所以移动后，空位置在你按按钮的地方)。

• 由 `puzzle` 自己去解：在控制条的右手边框上按按钮。

• 离开(跳出)：在控制条的中间按中间按钮。

`puzzle` 的命令行选项

- 使用一个大小不是 4×4 的框架，使用选项 `-size width x height`，其中尺寸以小方块为单位。
- 改变小方块被移动的速率，使用选项 `-speed num`。此处 `num` 是每秒移动的数目(缺省值为 5)。

28.4.4 打印一个大X标志——xlogo

`xlogo` 建立一个窗口并在它里面显示一个 X 标志。假如你重定窗口大小，标志再次绘出，且尽可能地将窗口填满。

28.4.5 跳动的多面体——ico

`ico` 产生一个窗口且在它里面有一个 12 面体——一个具有 12 面的实心体。这个多面体是会动的，它在窗口内不断地碰撞跳跃。在单色显示器上你可能只有一个直线构成的图(使用选项 `-i` 可以反白显示)，但是，在彩色显示屏幕上，你可以看到实心的彩色面。

试试下面的例子：

```
ico -nodeges -faces -colors red blue yellow green
```

你可以在一个根窗口内(背景窗口)设定一个跳跃的 12 面体，而不必靠它自己使用 `-r` 选项指定特定窗口。另外，`ico` 可指定多面体的面数，不是只有 12 面体。如果你想得到一个完整的列表可以键入下列命令：

```
ico -objhelp
```

28.4.6 动态几何图案——muncher 与 plaid

`muncher` 与 `plaid` 重复地绘出变化多端而有趣的几何图案。

28.7 显示信息和状态的程序

下面，我们将讨论一些提供窗口系统本身信息和目前状态的程序。包括检查系统上窗口各种属性的工具和一个观察 X 事件结构是如何工作的程序。这些程序有下列用途：

- 当你使用系统时，这些工具程序所提供的有关系统内部组织和操作的信息，可以帮助你

了解发生了什么事。

- 当你要做某种处理时，可以借助所给的信息来确认系统的组件。例如你必须知道一个窗口的window-id，才可以用xwd 来打印它。
- 可把从这些工具获得的信息，当成定制系统工具程序的输入值。

28.7.1 列出X服务程序的特征——xdpyinfo

xdpyinfo列出有关X服务程序和服务程序所控制屏幕的各种项目的信息。

28.7.2 获取有关窗口的信息

有三个程序可以提供目前在你显示器上窗口的信息。它们从不同的角度看系统：

- 1) 打印窗口的层次——xlswins。
- 2) 对单一窗口详细的信息——xwininfo。
- 3) 列出窗口的属性——xprop。

1. 打印窗口的层次 ——xlswins

在X系统上的窗口被安排成树状的层次，根窗口（也叫背景窗口）在最上层，在其下才是应用窗口，每一个应用窗口可以拥有它自己的子窗口层次。

xlswins 打输出这个树状结构，从根窗口或所指定的窗口起至其下所有的树状结构，对每一个窗口，xlswins 列出窗口的 window-id 并用括号括住它的名称（如果有的话），子窗口则在下面的几行依序以缩进两格的方式列出。下面对 xman应用程序列出子树输出命令，以演示xlswins如何剖析系统的结构。

```
xwd -id 0x60005d | xwd
```

注意 并非所有的窗口都可打印输出信息，如果你碰到这种无信息可供打印窗口，将获得类似下面的信息：

```
x Protocol error: BadMatch, invalid parameter attributes
Major opcode of failed request: 73 (X_GetImage)
```

如果在已经有好几个应用程序的系统上，你将可以看到 uwm 和xterm 菜单相关的子树，或者与 xmh和 xedit 的命令按钮相关的子树。

2. 关于单一窗口的详细信息——xwininfo

xwininfo能够针对特定的窗口给你大量的信息，你可以借助于命令行的选项告诉 xwininfo 你要哪一部分的信息。

用和xwd 相同的方法来指定你感兴趣的窗口：

- 交互式（缺省）：开始时，xwininfo给你一个十字线光标，你可以将它移到你所需的窗口再按鼠标按钮。
- 使用命令行选项：你可以用选项-root 指定根窗口，或用选项-id window-id 指定其他的窗口。

窗口信息可分为以下几类：

- 窗口的window-id：window-id 是窗口系统识别每一个窗口的参考，就如同先前所看到的，几个程序（包含 xwininfo它本身）给一个数字代表 window-id。例如要打印一个窗口，可以先执行xwininfo得到它的window-id，然后将此window-id 做为xwd 的id选项之参数。
- 层次的信息：可以看到这个窗口的父窗口的 id，几个子窗口的 id，以及根窗口的 id，在

xlswins 中可得到相同的信息，但在这里只能得到最近一层子窗口的id，而不是整个子树。

- 几何细节：窗口的大小和位置，以及它的四个角的位置。
- 和服务程序有关的重配置参数：像“gravity”和“backing store”这些状态，当窗口改变大小或从被遮盖的状态下重新显露出来时，服务程序需要用到这些参数，这些参数你自己不会用到，但可参考它了解系统如何运作。
- 事件参数：这些参数也是给服务程序而非给用户用的。
- 窗口管理器信息：我们曾经提过应用程序借助于给窗口管理器一些提示来完成通信（这些提示包括应用程序所希望窗口的大小，以及重定大小时的限制等）。xwininfo在这个部分的输出便是告诉你这些“提示”的资料。Program supplied location 为应用程序建议它自己应该摆哪里。如果你曾给过位置，不论是在命令行或源文件，都会在 User supplied location 出现，在窗口大小方面同理可推，resize increments 解释了为什么有些窗口（例如xterm 和xft）不能把大小定为任意数目的像素，因为应用程序已经告诉了窗口管理器在重定大小时按多少个像素的倍数放大或缩小（xterm 和xft 它们的大小和所使用的字体有关），你也可以由这个参数知道目前窗口的位置，所以稍后你可以在同一位置上重建它。

3. 列出窗口的属性——xprop

“属性”是指一小段有关窗口的数据，xprop 可列出一个指定窗口的所有“属性”，也可以打印一个字体的属性。可以经由常用的方式来选择窗口（按鼠标按钮或使用 -root 或-id 选项），如果是指定字体，则用选项 -font fontname。

显示出来的格式为：对每一个属性，均有一个属性名称，在其后用小括号括住的为属性的类型或格式，最后则为属性的值。大部分你所看到的属性类型为 STRING，属性的值用“ ”括起来，其他的属性类型的格式是专用的，从属性的值很容易了解它的意义，对字体显示的格式稍有不同，它没有属性类型，但属性值的意义也很明显。

注意 xprop 的输出相当的复杂，我们并不需要了解其所有的内容，视需要而定。

以下让我们看看从应用窗口、根窗口、字体所获得不同的输出：

其他的属性如下：

- WM_COMMAND：执行启动这个应用程序的命令行，被切成一个个用双引号括起来的单字。
- WM_CLIENT_MACHINE：执行这个客户程序的机器名称（这个例子应用程序和服务程序在同一部机器上执行，所以机器名称为 venus）。
- WM_CLASS：显示应用程序的 instant name 和 class name，instant name 是命令行中 -name 选项的值。
- WM_ICON_NAME：应用程序的图标所要显示出来的名称（窗口管理器必须能够支持方可）。
- WM_NAME：很奇怪的是，这不是应用程序的名称，而是由 -title 选项指定的窗口标题名称，有些窗口管理器会把标题名称显示在应用窗口的标题栏上。

注意 上述命令行中 -name 和 -title 两个选项使用得很广，但它们并非通用的选项，应用程序在撰写时必需要使用到 X Toolkit（工具箱）才能把这两个选项当成标准选项来用。

有趣的选项如下：

- RESOURCE_MANAGER：这个根窗口属性是 resource 结构的输入源之一，我们将在下面

详细地讨论它。

- 几个CUT_BUFFER：当你切取一段文本（做剪贴动作常用），这段被切取的文本被放在一个切取缓冲器中，这些缓冲器被当作是根窗口的属性来保存，对于切取，缓冲器是循环使用的，例如上次用5号缓冲器，则下次用6号，接下来7号、0号、1号等等，但是粘贴则一定使用上次切取动作所用的缓冲器。

28.7.3 观察X的事件——xev

事件或多或少驱动着整个窗口系统。所有的输入，不论是鼠标或键盘，均由事件来掌握。事件也被用来驱动窗口的重新配置和显示。xev 程序让你看到当不同的动作发生时，会产生什么事件，以及和事件有关的信息。

由于 xev提供大量的系统内部操作细节，你如果想要测试系统，这是一个很有用的程序，有两个说明联机帮助上未提到的选项可以影响到 xev 的行为：

- -bs option：此选项改变 xev 对服务程序是否使用backing store，使用backing store 将减少暴露窗口事件的次数（也就是减少应用程序重新更新它自己窗口的次数），正确的选项内容为always、whenmapped和notuseful。
- -s：使用 save-unders，也就是说，要求服务程序保存那些被 xev的窗口遮盖之窗口的内容。

如果将鼠标指针移入 xev窗口且按下键盘上的某一个键，则一个（或多个）键盘事件会发生，事件的信息包含了keycode 和keysym，这是最容易观察该机器上某一个键是什么键码的方法。执行xev，按一个键，xev 便会给出信息，这对定制键盘非常有用。