

**软件学院学生实验报告**

**实验课程名称：** 操作系统 **教师：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目名称** | 操作系统用户接口实验 | | | **实验成绩** |  |
| **学生姓名** |  | **学 号** | 076 | **年级专业班级** |  |
| **小组成员** |  | | | **实验日期** | 2017年9月19日 |

**一、实验目的**

熟悉操作系统的各种命令、图形、程序接口

实现程序的编译运行

**二、实验软硬件环境**

虚拟机VM

ubuntu-16.04.LTS

1. **实验内容及要求**

安装操作系统（选做）

熟悉操作系统的命令、图形、程序接口

1. **实验记录**

* **熟悉操作系统的图形用户接口(GUI)**

首先，图形用户[界面](https://baike.baidu.com/item/%E7%95%8C%E9%9D%A2)（Graphical User Interface，简称 GUI，又称图形[用户接口](https://baike.baidu.com/item/%E7%94%A8%E6%88%B7%E6%8E%A5%E5%8F%A3)）是指采用图形方式显示的计算机操作用户[界面](https://baike.baidu.com/item/%E7%95%8C%E9%9D%A2)。与早期计算机使用的[命令行界面](https://baike.baidu.com/item/%E5%91%BD%E4%BB%A4%E8%A1%8C%E7%95%8C%E9%9D%A2)相比，图形界面对于用户来说在视觉上更易于接受。然而这界面若要通过在显示屏的特定位置，以”各种美观而不单调的视觉消息“提示用户”状态的改变“，势必得比简单的消息呈现花上更多的计算能力。图形用户界面由窗口、下拉菜单、对话框及其相应的控制机制构成，在各种新式应用程序中都是标准化的，即相同的操作总是以同样的方式来完成，在图形用户界面，用户看到和操作的都是图形对象，应用的是计算机图形学的技术。另外，还有[嵌入式](https://baike.baidu.com/item/%E5%B5%8C%E5%85%A5%E5%BC%8F)GUI具有其下面几个方面的基本要求：轻型、占用资源少、高性能、高可靠性、便于移植、可配置等特点。一般总体的图形用户界面都具有视窗、桌面、单一文件界面、多文件界面、标签、菜单、图标、按钮等。

* + **典型的实现程序 - 桌面环境**

**Unity** – ubuntu：

Unity提供了隐藏启动器、触摸侧边栏就显示的选项。用户还可以调高/调低显示启动器菜单的灵敏度。Unity很简单、运行速度快，但Unity在系统设置下却没有定制桌面的太多选项。想安装主题及定制另外不同的选项，比如说系统菜单是否应该总是可见，或者“从启动器图标一次点击最小化”，用户需要安装第三方工具。CCSM和Unity Tweak Tool是面向Unity桌面环境的非常流行的定制工具。

首先Ubuntu重新安装ubuntu unity：

1,Ctrl+alt+T启动终端或者Ctrl+alt+F1登录字符界面，执行以下命令重新安装Ubuntu unity：

sudo apt-get install ubuntu-desktop

sudo apt-get install unity

sudo apt-get install unity-common

sudo apt-get install unity-lens\*

sudo apt-get install unity-services

sudo apt-get install unity-asset-pool

备注：后面两行命令执行时如果提示已经安装过了，那就不需要了。

2,更新ubuntu及组件

sudo apt-get update

sudo apt-get upgrade

3,移除所有无效的包和缓存

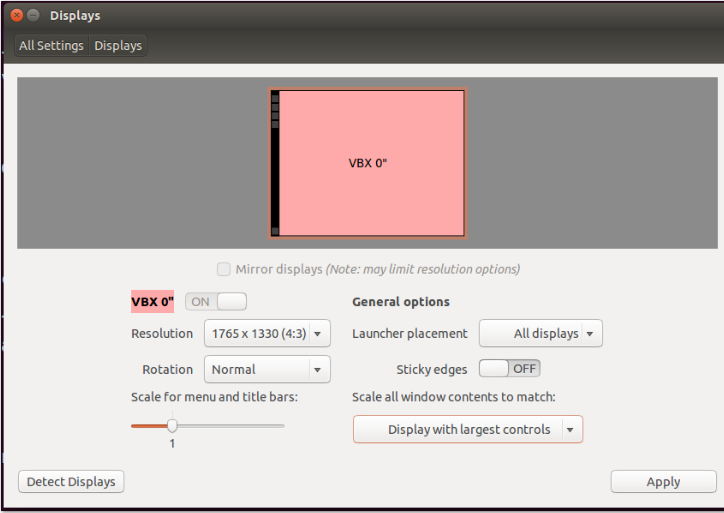
sudo apt-get autoremove

sudo apt-get autoclean

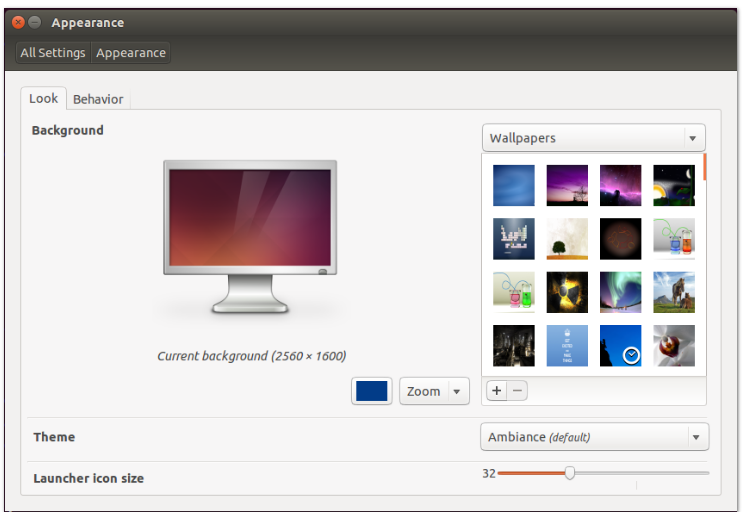


**自定义Unity：**

在ubuntu 14.04中，Unity 有一些以前没有的可定制项。登入你的 Unity，进入“设置”并选择“显示”，你将看到以下画面：



在“设置”中选择“外观”选项，可以看到如下画面。



还可以运用正式的Unity Tweak Tool，这款工具它集大量Unity桌面定制项目于一体。这些定制项大多能通过默认的Unity设置，命令行操作或者是即使是编辑有时候也很难寻找到的配置文件来实现。

其存在的优缺点：

**优点：**

· 界面直观

· 使用了平视显示器（HUD）等新技术

· 可以由第三方应用程序来定制

**缺点：**

· 实施的通知机制很差劲

· 用于定制的默认工具少得可怜

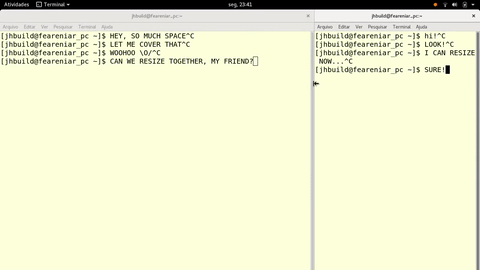
· 不一致的用户界面

**Gnome**：GNOME是一套纯粹自由的计算机软件，运行在操作系统上，提供图形桌面环境，包含了 Panel （用来启动此程式和显示目前的状态）、桌面 （应用程式和资料放置的地方）及一系列的标准桌面工具和应用程式，并且能让各个应用程式都能正常地运作。



GNOME的发展前景中其应用软件可以使用很多编程语言，但是作为GNOME发行版一部分的GNOME桌面纯粹用C写成。关于是否使用其它高级语言[C#](https://baike.baidu.com/item/C%23)，[Python](https://baike.baidu.com/item/Python)和Java等正在深入讨论。这些语言都已经用于开发GNOME应用程序，但是如果用于GNOME核心应用的开发就必须在所有GNOME安装中加入相应的虚拟机，这会抬高可以运行GNOME桌面的计算机的最低配置要求。

其如今最新的相关消息，GNOME 3.26.1 成功发布。如下图所示：



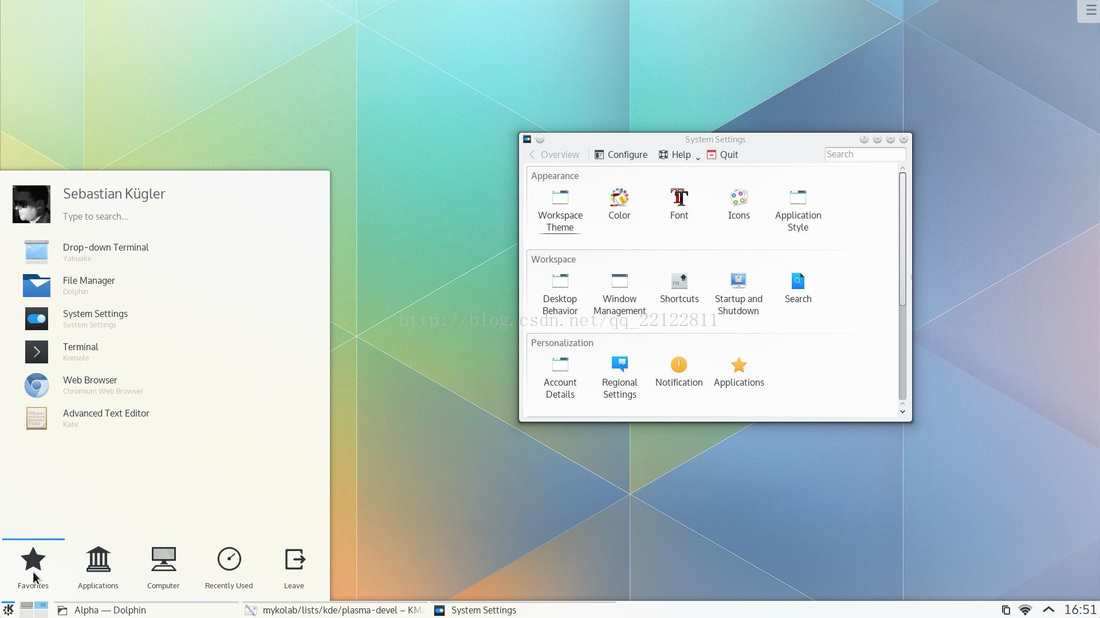
Mutter 3.26.1发布了在今年早些时候提到的更智能的窗口快照调整功能。它还增加了对沙盒Snap包的支持，并修复了在会话之间未被记住的监视器布局的问题。在其他地方，GNOME Shell 3.26.1增加了对无头模式的支持，改进了窗口预览的键盘导航，并使窗口组切换器“更加一致”

如果在Ubuntu 17.10上，可以个性化定制Linux桌面：Gnome 3 具体可参考该网站<http://www.linuxidc.com/Linux/2015-12/125970.htm>。其介绍了配置和定制Gnome外壳的方法，但由于Gnome 3采用了模块化设计方式，还可以把Gnome外壳完全换成别的系统，这种系统提供了全然不同的桌面，配置和定制方面有着全然不同的可能性。



KDE：

其开发框架以前在UNIX/X11下创作应用程序是一个非常困难并且单调乏味的过程。KDE 认识到了在一个计算平台上，平台和对于这个特定平台用户可用的一流应用程序的集合是同等重要的。从这些观点出发，KDE 项目已经开发了一流的复合文档应用程序框架，实现了最先进的框架技术并且因此把它自己直接置身于和诸如微软的 MFC / COM / ActiveX 技术等流行开发框架相竞争的位置。KDE 的 KParts复合文档技术使得开发人员可以快速创建一流的应用程序以实现最尖端的技术。KD环境中的一些默认应用程序包括：Dolphin（文件管理工具）、Konsole（终端）、Kate（文本编辑工具）、Gwenview（图片查看工具）、Okular（文档及PDF查看工具）、Digikam（照片编辑和整理工具）、KMail（电子邮件客户软件）、Quassel（IRC客户软件）、K3b（DVD刻录程序）、Krunner（启动器）及更多。



存在的优缺点：

**优点：**

· 最先进最强大的桌面环境

· 高度可定制

· 外观新颖而优美

· 与速度较慢/旧的硬件兼容

**缺点：**

· Kmail等一些组件对普通用户来说实在过于复杂。

· Kmail配置的改进对新手来说并不容易。

而上述提到的Gnome可能适合往往不经常调整系统的那些用户。这就是为什么Gnome甚至不包括一些简单的调整选项，比如更改主题、甚至更改字体。就这两种基本的调整而言，用户就需要安装gnome调整工具。所以，总的来说，Gnome在默认情况下不是具有很高的可定制性，但可以通过第三方应用程序/工具来实现。

Gnome用作Fedora中的默认桌面环境，提供在几款流行的Linux发行版中，比如Ubuntu、Debian、OpenSUSE和另外许多发行版。其存在的优缺点：

优点：

· 简单、易于使用

· 可通过插件来扩展功能

缺点：

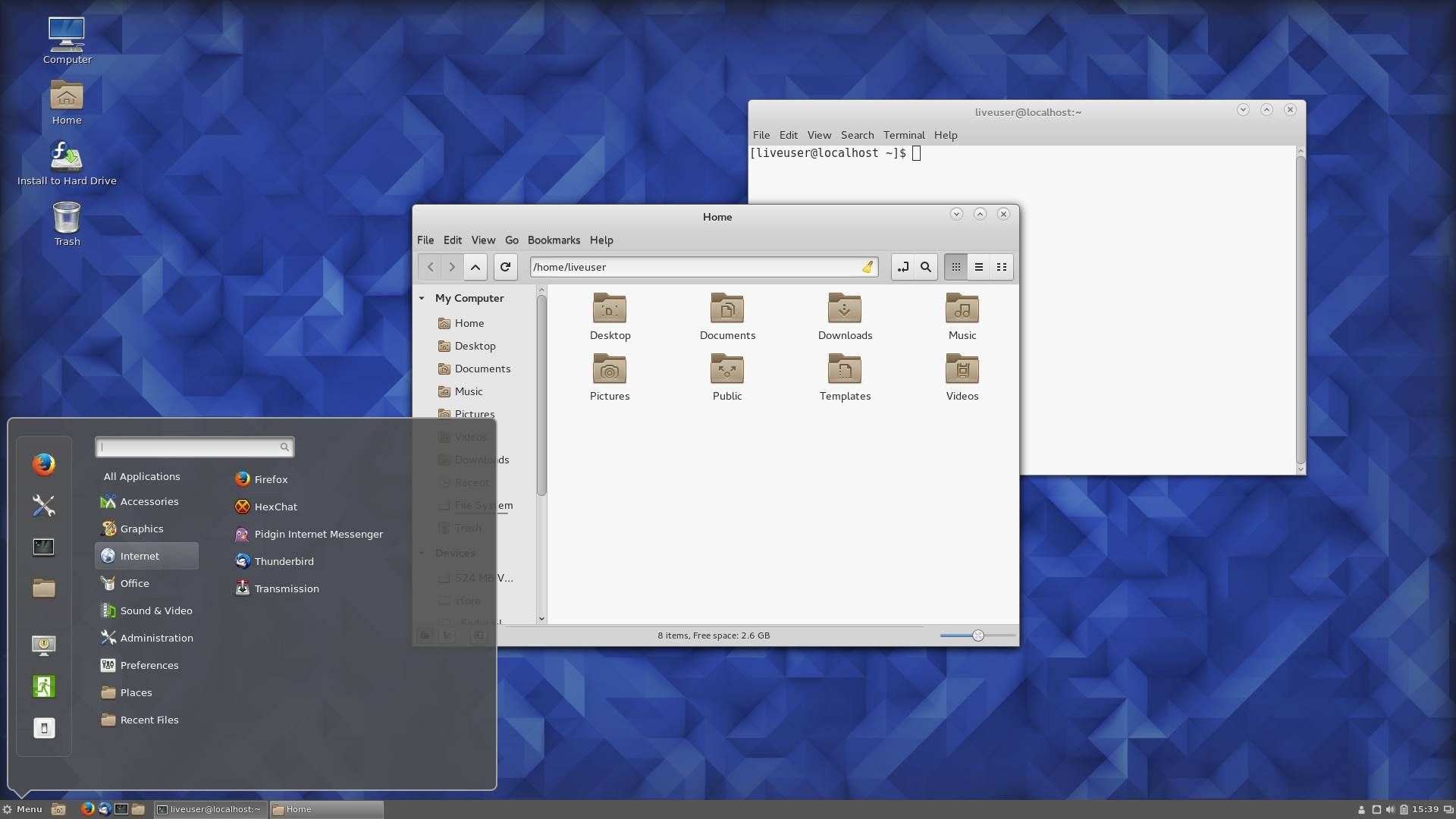
· 缺少其他桌面环境拥有的许多功能

· 插件管理很差

Cinamon：

Cinnamon拥有Gnome和Unity等桌面环境所没有的种种功能。Cinnamon是高度可定制的桌面环境，不需要任何外部插件、窗口组件和调整工具来定制桌面。Cinnamon甚至可以通过设置管理器本身来下载并安装主题，甚至不需要打开互联网浏览器。

由于种种出色的所需功能，Cinnamon对任何刚接触Linux的新用户来说都非常方便。许多用户放弃使用Linux，是因为他们并不了解Linux的工作方式，但是我强烈建议新手应从Cinnamon桌面环境开始入手。



许多流行的Linux发行版提供了各自版本的Cinnamon，比如Ubuntu、Fedora、OpenSUSE、Gentoo和Arch Linux。Cinnamon是Linux Mint的默认桌面环境。

优点：

· 极其成熟完美

· 对新用户来说很熟悉

· 由于派生出一些核心应用程序，让Gnome更易于使用

· 高度可定制

缺点：

有时可能有软件错误

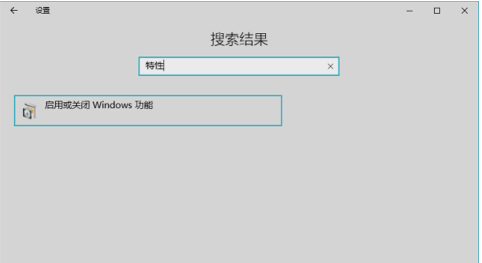
* **操作系统的命令行接口(CLI)**

它是在图形用户界面得到普及之前使用最为广泛的用户界面，它通常不支持鼠标，用户通过键盘输入指令，计算机接收到指令后，予以执行。也有人称之为字符用户界面（CUI）。不断增强命令行接口的脚本语言和宏语言可提供丰富的控制与自动化的系统管理能力，例如Linux系统的Bash或是Windows系统的Windows PowerShell。

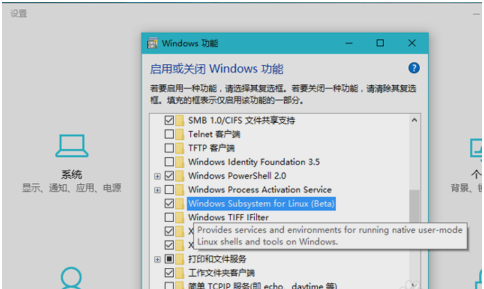
* + **典型的实现程序 - 命令行解释器**
    - **Bash**

Bash 的安装和使用：

在 Windows 10 内部预览版 build 14316 中，点击“开始”——“设置”，在打开的设置窗口搜索框中输入“特性”两个字，将会出现“启用或关闭 Windows 功能”条目，点击启动它。



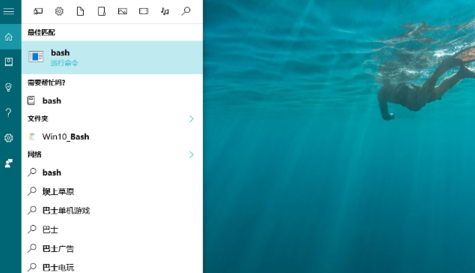
将弹出“Windows 功能”对话框，在对话框窗口列表中向下滚动，直到发现“Windows Subsystem for linux (Beta)”条目，点击选中其前面的复选框。



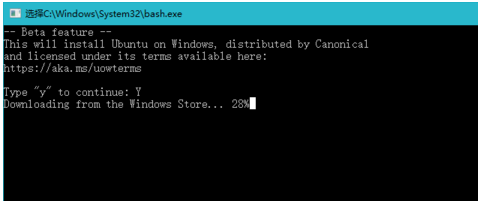
接下来将下载、安装和应用相关的组件，这些操作进程结束后，显示“立即重新启动”按钮，点击它重新启动电脑。



重新启动进入 Windows 10 的桌面，在小娜搜索框中键入：Bash，小娜顶端出现“Bash 运行命令”条目，回车（或点击）启动它。



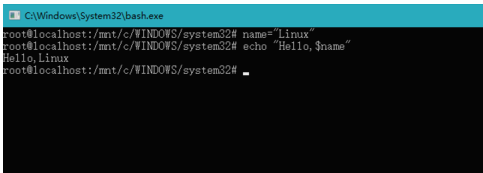
将会出现类似 Windows 传统 cmd.exe 界面的窗口，提示此 Bash 来自 Ubuntu，并且需要下载相关文件，键入“Y”将开始从微软商店中下载。在这里键入“Y”，下载开始。



经过一段时间的下载，Ubuntu 的 Bash 安装完成，提示符也变成了：

        root@localhost:/mnt/c/WINDOWS/System32#

在“#”号后面，可以键入 Bash 命令测试。



* + - **command.com**
    - **cmd**

command 是精简dos. Cmd则是win2000或更高版的系统中新增的功能，表面看来是一样，但cmd比command功能要强大得多!

Cmd.exe有很多命令行参数，具体情况如下：

CMD [/A | /U] [/Q] [/D] [/E:ON | /E:OFF] [/F:ON | /F:OFF] [/V:ON | /V:OFF] [

[/S] [/C | /K] string]

/C 执行字符串指定的命令然后中断;

/K 执行字符串指定的命令但保留;

/S 在 /C 或 /K 后修改字符串处理;

/Q 关闭回应;

/D 从注册表中停用执行AutoRun命令;

/A 使向内部管道或文件命令的输出成为ANSI;

/U 使向内部管道或文件命令的输出成为Unicode

/T:fg 设置前景/背景颜色(详细信息，请见 Color /?);

/E:ON 启用命令扩展;

/E:OFF 停用命令扩展;

/F:ON 启用文件和目录名称完成字符;

/F:OFF 停用文件和目录名称完成字符;

/V:ON 将c作为定界符启动延缓环境变量扩展。

/V:OFF 停用延缓的环境扩展。

请注意，如果字符串有引号，可以接受用命令分隔符“&&”隔开的多个命令。并且，由于兼容原因，/X与/E:ON相同，/Y与/E:OFF相同，并且/R与/C相同。忽略任何其它命令选项。

如果指定了/C或/K，命令选项后的命令行其余部分将作为命令行处理，在这种情况下，会使用下列逻辑处理引号字符("):

(1)如果符合下列所有条件，那么在命令行上的引号字符将被保留:

不带 /S 命令选项;

整整两个引号字符;

在两个引号字符之间没有特殊字符，特殊字符为下列中的一个: <>()@^|;

在两个引号字符之间有至少一个空白字符;

在两个引号字符之间有至少一个可执行文件的名称。

(2) 否则，老办法是，看第一个字符是否是一个引号字符，如果是，舍去开头的字符并删除命令行上的最后一个引号字符，保留最后一个引号字符之后的文字。如果/D未在命令行上被指定，当CMD.EXE开始时，它会寻找以下 REG\_SZ/REG\_EXPAND\_SZ注册表变量。如果其中一个或两个都存在，这两个变量会先被执行。

HKEY\_LOCAL\_MACHINE\\Software\\Microsoft\\Command Processor\\AutoRun

和

HKEY\_CURRENT\_USER\\Software\\Microsoft\\Command Processor\\AutoRun

命令扩展按默认值是启用的。你也可以使用 /E:OFF，为某一特定调用而停用扩展。你可以在用户登录会话上启用或停用cmd.exe所有调用的扩展，这要通过设置使用Regedit32.exe 的注册表中的一个或两个REG\_DWORD值:

HKEY\_LOCAL\_MACHINE\\Software\\Microsoft\\Command Processor\\EnableExtensions

和

HKEY\_CURRENT\_USER\\Software\\Microsoft\\Command Processor\\EnableExtensions到0x1或0x

0。用户特定设置比机器设置有优先权。命令行命令选项比注册表设置有优先权。

命令行扩展包括对下列命令所做的更改和添加:

Del 或Erase、Color、CD或ChDir、MD或MkDir、Prompt、Pushd、Popd、Set、SetLocal、End Local、IF、For、Call、Shift、Goto、Start（同时包括对外部命令调用所做的更改）、As soc、Ftype。

有关详细信息，请键入“HELP 命令名”。

延迟变量环境扩展不按默认值启用。你可以用/V:ON或 /V:OFF命令选项，为CMD.EXE的某个调用而启用或停用延迟环境变量扩充。你可以在机器上或用户登录会话上启用或停用CMD. EXE所有调用的完成，这要通过设置使用Regedit32.exe的注册表中的一个或两个REG\_DWORD

值:

HKEY\_LOCAL\_MACHINE\\Software\\Microsoft\\Command Processor\\DelayedExpansion

和

HKEY\_CURRENT\_USER\\Software\\Microsoft\\Command Processor\\DelayedExpansion到0x1或0x0。用户特定设置比机器设置有优先权。命令行命令选项比注册表设置有优先权。

如果延迟环境变量扩充被启用，惊叹号字符可在执行时间，被用来代替一个环境变量的数值。

文件和目录名完成不按默认值启用。你可以用/F:ON或/F:OFF命令选项，为cmd.exe的某个调用而启用或停用文件名完成。你可以在机器上或用户登录会话上启用或停用cmd.exe所有调用的完成，这要通过设置使用Regedit32.exe 的注册表中的一个或两个REG\_DWORD值:

HKEY\_LOCAL\_MACHINE\\Software\\Microsoft\\Command Processor\\CompletionChar

HKEY\_LOCAL\_MACHINE\\Software\\Microsoft\\Command Processor\\PathCompletionChar

和

HKEY\_CURRENT\_USER\\Software\\Microsoft\\Command Processor\\CompletionChar

HKEY\_CURRENT\_USER\\Software\\Microsoft\\Command Processor\\PathCompletionChar

由一个控制字符的十六进制值作为一个特定参数(例如，0x4是Ctrl-D，0x6是 Ctrl-F)，用户特定设置优先于机器设置。命令行命令选项优先于注册表设置。

如果完成是用/F:ON命令选项启用的，两个要使用的控制符是:目录名字完成用Ctrl-D，文件名完成用Ctrl-F。要停用注册表中的某个字符，请用空格(0x20)的数值，因为此字符不是控制字符。

如果键入两个控制字符中的一个，完成会被调用。完成功能将路径字符长带光标的左边，如果没有通配符，将通配符附加到左边，并建立相符的路径列表。然后，显示第一个相符的路径。如果没有相符的路径，则发出嘟嘟声，不影响显示。之后，重复按同一个控制字符会循环显示相符路径的列表。将 Shift 键跟控制字符同时按下，会倒着显示列表。如果对该行进行了任何编辑，并再次按下控制字符，保存的相符路径的列表会被丢弃，新的会被生成。如果在文件和目录名完成之间命令选项，会发生同样现象。两个控制字符之间的唯一区别是文件完成字符符合文件和目录名，而目录完成字符只符合目录名。如果文件完成被用于内置式目录命令(CD、MD或RD)，就会使用目录完成。

将引号用相符路径括起来，完成代码可以正确处理含有空格或其它特殊字符的文件名。同时，如果备份，然后从行内调用文件完成，完成被调用是位于光标右方的文字会被丢弃。

* **操作系统的应用编程接口(API)**

API（Application Programming Interface,应用程序编程接口）是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

1. LINUX系统调用原理-既应用层如何调用内核层函数之软件中断：

ARM Linux系统利用SWI指令来从用户空间进入内核空间。应用层一个进程或线程调用用系统调用进入内核层（SWI陷入），这个时候CPU进入管理模式一个调度周期，同时其它进程或线程还是正常的调度的。如果这个时候内核层由于资源原因导致调用此系统调用的进程或线程（内核层只有进程概念，应用层有进程和线程概念）休眠或阻塞，在这个调度周期时间完之后会保存当前休眠或阻塞状态。当前资源满足后休眠或阻塞结束，等下个调度周斯，当时阻塞或休眠进程继续运行，根据运行结果从SWI陷入的管理模式软中断返回一个结果给调用此系统调用的应用线程或进程，此时应用层线程或进程继续运行根据返回结果。

SWI命令格式：SWI{cond} immed\_24

Swi参数传递有两种格式：

1）、指令中24位的立即数指定了用户请求的服务类型，参数通过通用寄存器传递。如：

MOV R0,#34

SWI 12

2）、指令中的24位立即数被忽略，用户请求的服务类型有寄存器R0的只决定，参数通过其他的通用寄存器传递。如：

MOV R0, #12

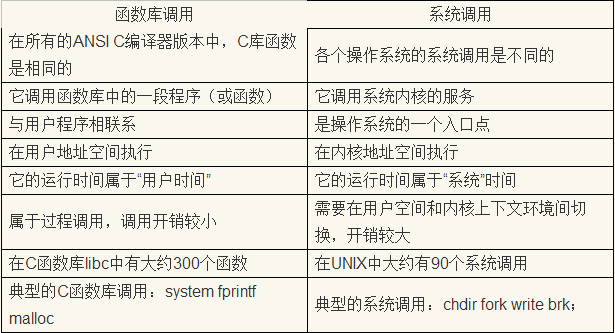
MOV R1, #34

SWI 0

1. Linux下对文件操作有两种方式：系统调用（system call）和库函数调用（Library functions）。

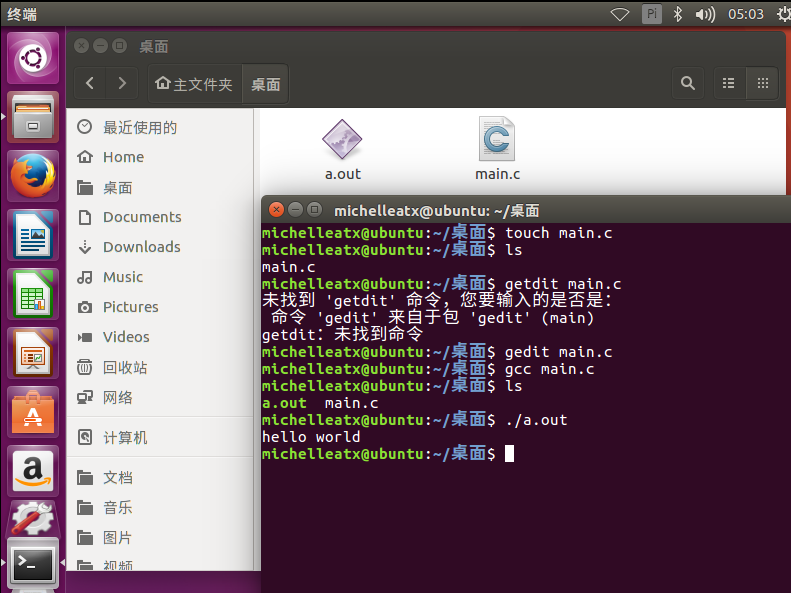
系统调用提供的函数如open, close, read, write, ioctl等，需包含头文件unistd.h。以write为例：其函数原型为 size\_t write(int fd, const void \*buf, size\_t nbytes)，其操作对象为文件描述符或文件句柄fd(file descriptor)，要想写一个文件，必须先以可写权限用open系统调用打开一个文件，获得所打开文件的fd，例如 fd=open(\"/dev/video\", O\_RDWR)。fd是一个整型值，每新打开一个文件，所获得的fd为当前最大fd加1。Linux系统默认分配了3个文件描述符值：0－standard input，1－standard output，2－standard error。系统调用通常用于底层文件访问（low-level file access），例如在驱动程序中对设备文件的直接访问。系统调用是操作系统相关的，因此一般没有跨操作系统的可移植性。系统调用发生在内核空间，因此如果在用户空间的一般应用程序中使用系统调用来进行文件操作，会有用户空间到内核空间切换的开销。事实上，即使在用户空间使用库函数来对文件进行操作，因为文件总是存在于存储介质上，因此不管是读写操作，都是对硬件（存储器）的操作，都必然会引起系统调用。也就是说，库函数对文件的操作实际上是通过系统调用来实现的。例如C库函数fwrite()就是通过write()系统调用来实现的。

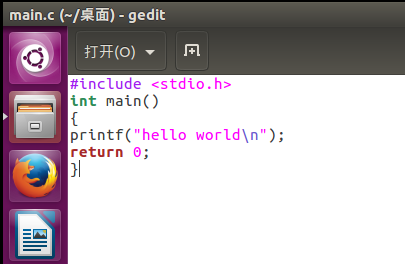
标准C库函数提供的文件操作函数如fopen, fread, fwrite, fclose, fflush, fseek等，需包含头文件stdio.h。以fwrite为例，其函数原型为size\_t fwrite(const void \*buffer, size\_t size, size\_t item\_num, FILE \*pf)，其操作对象为文件指针FILE \*pf，要想写一个文件，必须先以可写权限用fopen函数打开一个文件，获得所打开文件的FILE结构指针pf，例如pf=fopen(\"~/proj/filename\", \"w\")。实际上，由于库函数对文件的操作最终是通过系统调用实现的，因此，每打开一个文件所获得的FILE结构指针都有一个内核空间的文件描述符fd与之对应。同样有相应的预定义的FILE指针：stdin－standard input，stdout－standard output，stderr－standard error。库函数调用通常用于应用程序中对一般文件的访问。库函数调用是系统无关的，因此可移植性好。由于库函数调用是基于C库的，因此也就不可能用于内核空间的驱动程序中对设备的操作。



## 在ubuntu下进行C程序的编辑、编译、运行

1. 通过命令touch命令创建 main.c，通过命令ls查看文件有无创建成功。
2. 借助命令gedit 对main.c进行代码的编写，并及时保存。
3. 借助命令gcc对代码进行编译，没有其他错误提示出现a.out时，表示编译成功。通过命令ls可查看编译main.c文件后所生成的一个可执行文件a.out.
4. 通过命令./a.out运行该文件，得出hello world运行结果。





**五、实验小结（心得体会、遇到的问题及解决方式、未解决/需进一步研讨的问题或建议新实验方法等）**

学到的知识点：

1. ls命令用来列出目录的内容，它的输出格式可通过选项来控制。没有任何选项的ls命令将按照字母顺序列出所有非隐藏文件，显示的列数以正好适合窗口的大小为准。最常用的一组选项是-la。这里a表示要列出所有的文件；l表示以长 格式列出，它使输出结果成为一个详细的长列表。
2. **如今用 KDE 的人为什么没有用 Gnome 的人多？**

自己查阅了一些相关资料得出：KDE 和 GNOME 的差距并不大。加上预装 GNOME 的发行版多，KDE 本来就不占优势。默认 KDE 桌面的只有一个不怎么流行的 openSUSE 和另一个被 Ubuntu 视为异教徒的 Kubuntu 而已。默认 Xfce 桌面的好像没有。相比之下 GNOME 那么多发行版作为默认桌面，份额仍不及 KDE 和 Xfce ，也就证明了 GNOME 3 的失败。另外因为早期 redhat 用的是 gnome， redhat 是 gnome 背后最大的支持，近几年，ubuntu 用的也是 gnome，虽然说现在改用 unity 了。之所以 redhat 用 gnome，纯粹是当初 qt 协议问题，就算改 gpl 依然很多人不买账，后来 nokia 买下后改成 lgpl，跟 gtk 一样了其他方面讲，各个方面，kde 都要比 gnome 优秀kde 本身的恶名有2个原因，1个是 kde 4 代，开源项目是把用户当测试人员用的，最早的 4.0 bug 非常的多，文件管理器就算开在那里不动也可能会崩溃掉，plasma 崩溃后就一片黑色的桌面背景，导致用户一片愤怒，虽然说后续连着几个大版本的更新，稳定性好很多，但是很多人还是留有 4.0 那个时候的印象。还有就是KDE使用的QT是用C++作为主要开发语言的，而GTK+是C。C vs C++已经足以成为很多开发者做出选择的理由了，尽管QT的文档那么优秀。

1. 了解LINUX系统后的一些总结：

linux内核中所有的系统调用函数都用sys\_开头。函数定义中的FASTCALL宏，用于通知编译器，使用寄存器来传递参数。如果上面两个宏都没有，则使用默认传参规则，前4个参数通过R0~R3寄存器传递，其余更多的参数通过栈传递。

进程状态:

task\_struct中的state描述进程的当前状态。进程的状态一共有5种，而进程必然处于其中一种状态：

1）TASK\_RUNNING（运行）——进程是可执行的，它或者正在执行，或者在运行队列中等待执行。这是进程在用户空间中执行唯一可能的状态；也可以应用到内核空间中正在执行的进程。

2）TASK\_INTERRUPTIBLE（可中断）——进程正在睡眠（也就是说它被阻塞）等待某些条件的达成。一旦这些条件达成，内核就会把进程状态设置为运行，处于此状态的进程也会因为接收到信号而提前被唤醒并投入运行。

3）TASK\_UNINTERRUPTIBLE（不可中断）——除了不会因为接收到信号而被唤醒从而投入运行外，这个状态与可打断状态相同。这个状态通常在进程必须在等待时不受干扰或等待事件很快就会发生时出现。由于处于此状态的任务对信号不作响应，所以较之可中断状态，使用得较少。

4）TASK\_ZOMBIE（僵死）——该进程已经结束了，但是其父进程还没有调用wait4()系统调用。为了父进程能够获知它的消息，子进程的进程描述符仍然被保留着。一旦父进程调用了wait4()，进程描述符就会被释放。

TASK\_STOPPED（停止）——进程停止执行，进程没有投入运行也不能投入运行。通常这种状态发生在接收到SIGSTOP，SIGTSTP，SIGTTIN，SIGTTOU等信号的时候。此外，在调试期间接收到任何信号，都会使进程进入这种状态。

需要调整进程的状态，最好使用set\_task\_state(task, state)函数，在必要的时候，它会设置内存屏障来强制其他处理器作重新排序（SMP）。

在Linux系统中，所有的进程都是PID为1的init进程的后代。

进程调度：

调度是一个平衡的过程。一方面，它要保证各个运行的进程能够最大限度的使用CPU(即尽量少的切换进程，进程切换过多，CPU的时间会浪费在切换上)；另一方面，保证各个进程能公平的使用CPU(即防止一个进程长时间独占CPU的情况)。

总体来说，由于之前对Linux系统的不熟悉，本次实验查资料获得知识颇为丰富，涉猎广泛，选取了一些纳入实验报告，期间有很多已在自己电脑上运行过一些小实例，但仍存在一些疑惑，对命令调用还不够熟悉等。