**<操作系统原理>课外实践2指导说明：**

实践名称： 熟悉Ubuntu 编程环境

实践目标：

1、了解Ubuntu系统基本操作方法，学会独立使用该系统。

2．熟悉Ubuntu下如何编辑、编译和运行一个C语言程序。

3．学会利用gcc、gdb编译、调试C程序。

实践内容: 1、装系统

1. 熟悉unbuan 环境
2. 运行一个C程序

**学习内容**

**一、登陆Linux**

开机，选择 进入，再选择Ubuntu登陆窗口，输入用户名： students ，输入密码： 111111 ，进入Ubuntu图形桌面环境。

**二、熟悉Ubuntu图形桌面环境**

桌面包含上方的菜单栏和下方的任务栏。菜单栏包含“应用程序菜单”、“位置”、“系统”。

通过主菜单可访问应用程序。

(1) “应用程序”菜单

“应用程序”菜单包含“办公”、“附件”、“互联网”、“图形”、“游戏”等。

“办公”包含了文字处理及电子表格等应用程序。

“附件”下包含了“搜索工具”、“计算器”、“文本编辑器”和“终端”等应用程序。

(2)“位置”菜单

“位置”菜单包含“主文件夹”、“桌面”等信息。

(3)“系统”菜单

“系统”菜单包含“首选项”和“系统管理”等信息。

(4)启动终端模拟器

GNOME终端模拟器用一个窗口来模拟字符终端的行为。终端常常被称为命令行或者 shell，Linux 中绝大部分工作都可以用命令行完成。要启动一个终端，可以选择 ***应用程序 → 附件 → 终端。***

**三、目录和文件系统**

Linux 和 Unix 文件系统被组织成一个有层次的树形结构。文件系统的最上层是 /，或称为 根目录。在 Unix 和 Linux 的设计理念中，一切皆为文件——包括硬盘、分区和可插拔介质。这就意味着所有其它文件和目录（包括其它硬盘和分区）都位于根目录中。 例如：/home/student/sy1.c 给出了正确的完整路径，它指向 sy1.c 文件，而该文件位于 student 目录下，该目录又位于 home 目录，最後，home 目录又位于根(/) 目录下。 在根 (/) 目录下，有一组重要的系统目录，在大部分 Linux 发行版里都通用。直接位于根 (/) 目录下的常见目录列表如下：

* /bin - 重要的二进制 (binary) 应用程序
* /boot - 启动 (boot) 配置文件
* /dev - 设备 (device) 文件
* /etc - 配置文件、启动脚本等 (etc)
* /home - 本地用户主 (home) 目录
* /lib - 系统库 (libraries) 文件
* /lost+found - 在根 (/) 目录下提供一个遗失+查找(lost+found) 系统
* /media - 挂载可移动介质 (media)，诸如 CD、数码相机等
* /mnt - 挂载 (mounted) 文件系统
* /opt - 提供一个供可选的 (optional) 应用程序安装目录
* /proc - 特殊的动态目录，用以维护系统信息和状态，包括当前运行中进程 (processes) 信息。
* /root - root (root) 用户主文件夹，读作“slash-root”
* /sbin - 重要的系统二进制 (system binaries) 文件
* /sys - 系统 (system) 文件
* /tmp - 临时(temporary)文件
* /usr - 包含绝大部分所有用户(users)都能访问的应用程序和文件
* /var - 经常变化的(variable)文件，诸如日志或数据库等

**四、文件权限**

Linux 系统上的所有文件都有权限，以允许或者阻止其它用户查阅、修改或者执行。超级用户“root”则可以访问系统上的任意文件。每个文件都有访问限制、用户限制以及隶属于某个用户/组(owner/group)。 所有文件皆有如下三组权限加以保护，按重要性依次排列：

用户(user) ：适用于该文件的所有者

组(group) ：适用于该文件所属的组

其他(other) ：适用于所有其他用户

上述三组权限信息的每一组都包含了实际权限。这些权限信息，连同它们对文件和目录具有的不同含义，概述如下：

读(read) ：允许显示/打开该文件 可以显示目录内容

写(write) ：可以编辑或删除该文件 可以更改目录内容

可执行(execute) ：可执行文件可以作为程序运行 可以进入该目录

要查看和编辑文件或目录的权限，请打开 ***位置*** → ***主文件夹*** 并在文件或者目录上单击右键。然後选择 ***属性***。授权信息就在 ***权限标签页***中，如果您是该文件的所有者，您可以编辑所有的授权等级。

例如：ls –l

普通文件：-rw-￤r--￤r-- 1 bin bin 2208 May 28 1987 /etc/rc

￤ ￤

目录文件：drwx￤r-x￤r-x 2 bin bin 240 Nov 13 1987 dos

￤ ￤

特殊文件：brw-￤r-x￤r-x 3 bin bin 2,7 Jan 12 15:07 /dev/fd0

c-w-￤-w-￤-w- 2 bin bin 6,0,Jan 11 13:37 /dev/Lp

存取权

所 同 其

有 组 他

者 用 用

户

用chmod改变文件或目录的存取权限：

格式：

chmod 谁 操作符 许可权 {文件名（或目录名）}

a + r

g w

- x

o s

u = t

u

g

o

其中，a:代表all,即所有的用户；

g:代表group，即小组用户；

o:代表others,即一般用户；

u:代表 user,即文件的所有者；

+：代表增加许可权；

-：代表删除许可权；

=：代表赋予指定的许可权。

例如：$chmod u+w display 完成对u用户增加对文件display的写权限。

注：上述命令是由文件所有者或超级用户进行修改。

**五、练习常用的Shell命令。（重点）**

当用户登录到字符界面系统或使用终端模拟窗口时，就是在和称为shell的命令解释程序进行通信。当用户在键盘上输入一条命令时，shell程序将对命令进行解释并完成相应的动作。这种动作可能是执行用户的应用程序，或者是调用一个编辑器、GNU/Linux实用程序或其他标准程序，或者是一条错误信息，告诉用户输入了错误的命令。

1．目录操作

mkdir abc 创建一个目录abc

cd abc 将工作目录改变到abc

cd 改变当前目录到主目录

ls 列出当前目录的内容

ls -l 输出当前目录内容的长列表，每个目录或文件占一行

pwd 显示当前目录的全路径

2．文件显示实用程序

cat mx.c 显示mx.c文件内容

more mx.c 分屏显示mx.c内容

tail mx.c 显示文件后几行

cat file1 file2 连接file1 和file2

head filename 显示文件filename的开始10行

wc filename 统计文件filename中的行数、单词数和字符数

od 文件 查看非文本文件

3．文件管理实用程序

cp file1 file2 将文件1复制到文件2

mv file1 file2 将文件重命名为file2

rm filename 删除文件filename

rm -i filename 请求用户确认删除

4．数据操作实用程序

tty 显示当前终端的路径和文件名

who 显示当前登录用户的列表

sort filename 显示文件filename中的行的排序结果

spell filename 检查文件filename中的拼写错误

5．其他实用程序

date 输出系统日期和时间

cal 显示本月的日历。cal 2002 显示2002年的日历

clear 清除终端屏幕

history 显示你以前执行过的命令的列表

man 显示实用程序的有用信息，并提供该实用程序的基本用法

echo 读取参数并把它写到输出

**六、熟悉gcc编译器**

GNU/Linux中通常使用的C编译器是GNU gcc。编译器把源程序编译生成目标代码的任务分为以下4步：

a. 预处理，把预处理命令扫描处理完毕；

b. 编译，把预处理后的结果编译成汇编或者目标模块；

c. 汇编，把编译出来的结果汇编成具体CPU上的目标代码模块；

d. 连接，把多个目标代码模块连接生成一个大的目标模块；

1．使用语法：

　　gcc [ option | filename ]...

　　其中 option 为 gcc 使用时的选项，而 filename 为 gcc要处理的文件。

2．GCC选项

    GCC的选项有很多类，这类选项控制着GCC程序的运行，以达到特定的编译目的。

⑴全局选项(OVERALL OPTIONS)

    全局开关用来控制在“GCC功能介绍”中的GCC的4个步骤的运行，在缺省的情况下，这4个步骤都是要执行的，但是当给定一些全局开关后，这些步骤就会在 某一步停止执行，这产生中间结果，例如可能你只是需要中间生成的预处理的结果或者是汇编文件(比如你的目的是为了看某个CPU上的汇编语言怎么写)。

①  –x  language

    对于源文件是用什么语言编写的，可以通过文件名的后缀来标示，也可以用这开关。指定输入文件是什么语言编写的，language 可以是如下的内容

a.  c

b. objective-c

c. c-header

d. c++

e.cpp-output

f.assembler

g.assembler-with-cpp

②–x none

把-x开关都给关掉了。

③  –c

编译成把源文件目标代码，不做连接的动作。

④–S

把源文件编译成汇编代码，不做汇编和连接的动作。

⑤–E

只把源文件进行预处理之后的结果输出来。不做编译，汇编，连接的动作。

**⑥ –o file (常用)**

**指明输出文件名是file。**

⑦–v

把整个编译过程的输出信息都给打印出来。

⑧–pipe

由于gcc的工作分为好几步才完成，所以需要在过程中生成临时文件，使用-pipe就是用管道替换临时文件。

⑵ 语言相关选项(Language Options)

用来处理和语言相关的选项。

①–ansi

    这个开关让GCC编译器把所有的GNU的编译器特性都给关掉，让你的程序可以和ansi标准兼容。

②–include file

    在编译之前，把file包含进去，相当于在所有编译的源文件最前面加入了一个#include <file>语句，

③–C

    同-E参数配合使用。让预处理后的结果，把注释保留，让人能够比较好读它。

⑶连接开关(Linker Options)

    用来控制连接过程的开关选项。

① –llibrary

    连接库文件开关。例如-lugl，则是把程序同libugl.a文件进行连接。

② –lobjc

    这个开关用在面向对象的C语言文件的库文件处理中。

③ –nostartfiles

    在连接的时候不把系统相关的启动代码连接进来。

④ –nostdlib

    在连接的时候不把系统相关的启动文件和系统相关的库连接进来。

⑤–static

    在一些系统上支持动态连接，这个开关则不允许动态连接。

⑥shared

    生成可共享的被其他程序连接的目标模块。

⑷目录相关开关(Directory Options)

    用于定义与目录操作相关的开关。

–Ldir

    搜寻库文件(\*.a)的路径。

**⑸调试开关(Debugging Options)**

**–g**

    把调试开关打开，让编译的目标文件有调试信息。

–V version

用来告诉编译器使用它的多少版本的功能，version参数用来表示版本。

**七、掌握Ubuntu下C程序编辑运行过程（重点）**

**Ubuntu**下编写C程序要经过以下几个步骤：

⑴启动常用的编辑器，键入C源程序代码。

例如，点击***应用程序/附件/文本编辑器***，进入编辑环境，输入C源程序，保存并命名为hello.c

# include <stdio.h>

void main(void)

{

Printf(“Hello world!\n”);

}

⑵编译源程序

点击***应用程序/附件/终端***，进入命令行。用gcc编译器对C源程序进行编译，以生成一个可执行文件。方法：

gcc -o hello.out hello.c ↙

⑶运行可执行文件

·/hello.out ↙

注：命令行中 -o选项表示要求编译器输出可执行文件名为hello.out文件，hello.c是源程序文件。

实践要求：1、完成学习内容，学习；

2、完成实践2报告。

实验报告要求：

1．举例列出常用的shell命令使用方法。学习内容中列举的shell命令都要运行，报告中至少有6个以上命令结果截图；

2．通过实例总结上机调试C语言程序的过程。用gcc 编译程序，写出实验结果，