

# 操作系统实验 1 熟悉 Linux 系统

计算机 1202 张艺瀚  
学号:20123852

June 23, 2015

## 1 题目

熟悉 Linux 系统的基本操作和开发环境

## 2 目的

熟悉和掌握 Linux 基本命令，熟悉 Linux 编程环境，为以后的实验做好准备。

## 3 要求

1. 熟练掌握 Linux 基本文件命令。
2. 掌握 Linux 编辑程序、对源代码进行编译、连接、运行及调试的过程。
3. 认真做好预习，书写预习报告。
4. 实验完成后要认真总结、完成实验报告。

## 4 常用 Linux 命令及含义

见表 1

## 5 在 Linux 环境下编制、调试源程序的过程

在 Ubuntu 12.04 LTS 下编制 C++ 程序。有源文件 `thread.cpp`, `pcb.h`, `pcb.cpp`, `group_tree.h`, `group_tree.cpp`。它们的 include 关系如下（见代码清单 1-5）：

命令	用法
创建子目录命令 <code>mkdir</code>	<code>mkdir subdir</code>
切换路径命令 <code>cd</code>	<code>cd subdir</code>
显示当前工作路径命令 <code>pwd</code>	<code>pwd</code>
拷贝文件命令 <code>cp</code>	<code>cp sfile dfile</code>
删除文件命令 <code>rm</code>	<code>rm filename</code>
删除目录命令 <code>rmdir</code>	<code>rmdir sdir</code> (子目录为空目录)
显示文件或把 <code>file1</code> 与 <code>file2</code> 的连接 <code>cat</code>	显示文件内容 <code>cat filename</code> 把 <code>file1</code> 与 <code>file2</code> 的连接 <code>cat file1 file2 &gt; file3</code>
改变文件或目录的访问权限 <code>chmod -rwxr-x--x</code>	<code>chmod +/- 数值 filename</code>
改变文件或目录的拥有者 <code>chown</code>	<code>chown filename username</code>
列表显示文件名或目录名 <code>ls</code>	<code>ls</code> 或 <code>ls -l</code>
查找文件或目录 <code>find</code>	<code>find</code> 文件或目录
显示进程或状态 <code>ps</code>	<code>ps</code>
杀进程命令 <code>kill</code>	<code>kill pid</code>
编译	<code>gcc -o 目标文件源文件 (.c)</code> <code>g++ -o 目标文件源文件 (.cpp)</code>
运行	<code>./目标文件</code>
保存运行结果	<code>./目标文件 &gt; 文件名</code>
显示运行结果	<code>cat 文件名</code>
移动相对路径下的文件到绝对路径下	<code>mv 路径/文件路径/文件</code>
在当前目录下改名	<code>mv 文件名新名称</code>
编译	<code>make</code>
安装编译好的源码包	<code>make install</code>
安装包	<code>sudo apt-get install package</code>
删除包	<code>sudo apt-get remove package</code>
更新源	<code>sudo apt-get update</code>
更新已安装的包	<code>sudo apt-get upgrade</code>
升级系统	<code>sudo apt-get dist-upgrade</code>
使用 <code>dselect</code> 升级	<code>sudo apt-get dselect-upgrade</code>
安装相关的编译环境	<code>sudo apt-get build-dep package</code>
下载该包的源代码	<code>apt-get source package</code>
检查是否有损坏的依赖	<code>sudo apt-get check</code>
清理所有软件缓存	<code>sudo apt-get clean</code>

Table 1: 常用 Linux 命令及含义

```
1 // thread.cpp
2 #include "pcb.h"
3 #include "group_tree.h"
```

Listing 1: thread.cpp 的 include 关系

```
1 // pcb.h
2 #ifndef PCB_H
3 #define PCB_H
4 #endif // PCB_H
```

Listing 2: pcb.h 的 include 关系

```
1 // pcb.cpp
2 #include "pcb.h"
```

Listing 3: pcb.cpp 的 include 关系

```
1 // group_tree.h
2 #ifndef GROUP_TREE_H
3 #define GROUP_TREE_H
4 #include "pcb.h"
5 #endif // GROUP_TREE_H
```

Listing 4: group\_tree.h 的 include 关系

```
1 // group_tree.cpp
2 #include "group_tree.h"
```

Listing 5: group\_tree.cpp 的 include 关系

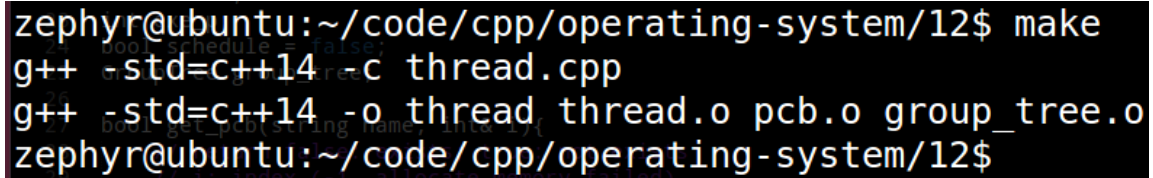
编写 makefile 如下:

```
1 thread: thread.o pcb.o group_tree.o
2   g++ -std=c++14 -o thread thread.o pcb.o group_tree.o
3 thread.o: thread.cpp pcb.h group_tree.h
4   g++ -std=c++14 -c thread.cpp
5 pcb.o: pcb.cpp pcb.h
```

```
6 g++ -std=c++14 -c pcb.cpp
7 group_tree.o: group_tree.cpp group_tree.h pcb.h
8 g++ -std=c++14 -c group_tree.cpp
```

Listing 6: makefile 代码清单

在终端中 `cd` 到当前目录下，运行 `make` 编译源代码，显示如下信息：

A terminal window with a black background and white text. The prompt is 'zephyr@ubuntu:~/code/cpp/operating-system/12\$'. The user enters 'make'. The output shows 'g++ -std=c++14 -c thread.cpp' followed by 'g++ -std=c++14 -o thread thread.o pcb.o group\_tree.o'. The prompt returns to 'zephyr@ubuntu:~/code/cpp/operating-system/12\$'.

```
zephyr@ubuntu:~/code/cpp/operating-system/12$ make
g++ -std=c++14 -c thread.cpp
g++ -std=c++14 -o thread thread.o pcb.o group_tree.o
zephyr@ubuntu:~/code/cpp/operating-system/12$
```

Figure 1: 编译通过效果图

说明编译通过。输入 `./thread` 运行可执行文件。