

准备环境

Any Distribution of Linux, Ubuntu recommended.

下面所有题目，参考资料都给出比较详细的解释。资料不足的地方，请自行查找。

题目1 Git使用

Git是Linux Torvalds写的优秀的开源的分布式的版本控制软件，[Github](#)是一个著名的远程Git服务器，很多优秀的项目都在其上托管，包括[Linux内核](#)在内，本题的目的是让大家早点接触git及Github。

1. 安装git
2. 创建自己的github账号
3. 新建一个仓库，管理此次测试所有解答
4. 可选项：使用SSH无密码推送到Github

参考资料

- [廖雪峰的Git教程](#)

题目2 设备文件及磁盘相关

Linux的原则是：一切都是文件，包括设备。这里，我们将了解Linux是如何管理硬件的，以及磁盘是如何使用的。

回答以下问题：

1. /dev/sdb7代表的含义是？它是主分区还是拓展分区？拓展分区有几个？
2. 磁盘的第一个扇区存放的是什么？它的大小多大？如何划分的？
3. 如何安装有windows和linux双系统，MBR如何使用？又区分两个系统？
4. 设备文件里主要存放什么信息？如果设备插入以后，却没有相应的设备文件，可能问题是什么？
5. /dev/null, /dev/zero是什么？他们一般有什么用处？

参考资料

- [Linux设备文件简介](#)

题目3 Ext2文件系统

文件系统，简单来说，就是操作系统用于明确存储设备或分区上的文件的方法和数据结构；即在存储设备上组织文件的方法。现在，让我们探一探Linux主要的Ext索引式日志文件系统的神秘面纱。

回答以下问题：

1. FAT文件系统碎片如何产生的？Ext2文件系统需要定期执行碎片整理吗？

2. 一个文件的大小小于一个block的大小，是否能够装入下一个文件内容？
3. 如果系统使用2KB的block，而文件系统有10000个小文件，每个文件大小是80B，那么磁盘浪费多少空间？有什么解决方案？
4. 在硬盘上读取文件/etc/passwd，它的inode和block访问顺序是？

参考资料

- [Ext2文件系统](#)

题目4 文件系统的挂载

在Linux下，文件系统挂载是相当方便的，我们不需要像windows那样安装虚拟光驱。也就是说，一个ISO文件，我们可以直接使用命令挂载在一个目录下然后直接访问。下面我们来看一看具体的挂载操作。

解决以下问题：

1. 如果以前目录不为空，但是你却将一个文件挂载到其上，会不会影响这个目录？
2. 将附件的文件挂载，并取出其中文件提交。
3. 将此次测试的解答（不包括此镜像文件），题目（即此文档）制作成一个镜像文件，上传到Github。

参考资料

- man dd,mkfs
- [Linux文件挂载](#)

题目5 Vim编辑器和Shell脚本

Linux下有很多优秀的文本编辑器，比如Emacs和Vim，前者号称可以在里面打游戏，浏览网页，控制你的冰箱等，确实可以做到。但是今天我们使用另一个与它不相上下的文本编辑器，现在开始，逐渐丢弃IDE吧。

解决以下问题：

1. Ubuntu默认是不安装Vim的，请安装。
2. 使用Vim编辑一个hello.c，要求以main函数递归方式打印出“hello,world”。
3. 使用命令行方式编译程序(gcc)，然后执行，并要求写成shell脚本。也就是说，执行这个脚本后，hello.c被编译，并且"hello world"输出。
4. 使用vim写一个脚本ip.sh，查找当前网段内IP用户，重定向到ip.txt文件中。
5. 使用vim写一个脚本user.sh，输出当前系统所有用户及是否可登录。
6. 使用vim写一个脚本replace.sh，它接受两个参数w1,w2,统计一个文件中w1的个数，在哪一行，并且询问是否所有的w1替换成w2，并实现改替换功能。
7. 可选项：修改~/.vimrc，让vim更强大，更易用

参考资料

- [Shell教程](#)
- [/etc/passwd](#)

- `man wc/grep/awk`

题目6 源码编译安装软件

Linux下基本所有软件都是开源的，有些时候，我们Linux发行版包管理找不到相应的软件包时，可以考虑直接去软件官网下载源码进行安装。下面以一个小软件NTP为例：

解决以下问题：

1. 从[官网](#)下载tar包。
2. 解压tar包，生成makefile。
3. 编译，安装。
4. 将ntp运行状态截图(`ps -aux | grep XXX`)。

参考资料

- [Linux如何编译安装源码包软件](#)

题目7 一些常用命令的实现

Linux下很多常用命令/bin,/sbin,/usr/bin,/usr/sbin等都是C语言实现的，很多命令我们自己也可以实现。

解决以下问题：

1. 实现cp命令，即cp.c。
2. 打开一个普通文件显示并支持重定向，即程序接受最多两个参数，第一个为要打开的文件名，第二个为重定向到的文件名(如果存在)，程序为show.c，只需能显示show.c即可，不必支持太大文件。
3. 写一个程序date.c，获得当前时间（包括日期等）。
4. 写一个程序mem.c，打印出当前内存使用信息（提示：使用/proc文件系统）。
5. 写一个程序wc.c，统计一个文件中行数，字符数，单词数。

参考资料

- `man page`
- [linux gcc](#)

题目8 Linux(POSIX)编程

如今单进程的程序很少使用，大部分都是多进程或者多线程的，我们来写一个多线程的程序解决一个问题。我们也尝试写一个守护进程，它是自己在后台运行的类似windows服务的程序。另外，我们看一下Linux下的网络通信。

1. 写一个多进程程序readfile.c，两个子进程分别打开date.c，mem.c并输出，父进程打印子进程PID。
2. 写一个守护线程time.c，要求每隔15s向~/daemon_log.txt每行打印出当前时间，进程总数目，CPU利用率(/proc文件系统)。
3. 利用socket写一个client/server程序，要求两台电脑，一台运行server，一台运行

client, 两者相互发送消息, 即简单的聊天室。

参考资料

- [Linux Deamon编写](#)
- [Linux 多进程](#)
- [Linux socket](#)