零基础学 Linux

1） 通过 U 盘安装 Linux，有条件的情况下尝试刻光盘安装。
学习和了解 Linux 的安装过程，发现问题寻找答案的能力。

2 天以内完成　优秀

补充练习：Windows 下安装虚拟机，虚拟机上安装 linux

2） 在机器上挂一个空硬盘，机器重启后依然可见硬盘。**（选学）**
在一块原始硬盘上做出 5 块空间相同的小盘。
用这个 5 个小盘，做 raid0，raid1，raid5，raid6。
完成模拟一块坏盘，raid 恢复的过程
mdadm stop raid 后，重新恢复 raid
往 raid 里面增加新盘

5 天以内完成优秀

3） 学习常用命令 cp, ls, mv,pwd, cat, crontab, cut, df, grep, head, history, ifconfig, kill, locate,ln, mkdir, netstat, nohup,ping, ps, rm, rsync, sort, tail, telnet, touch, uptime,wc 等等，了解和记录每个命令的基本功能

2 天以内完成优秀

4） 压缩数据命令强化学习，tar、7za、zip、bz2 等

自动生成一个 1GB 随机纯文本文件，内容是中文或者英文。
采用以上命令压缩，比较压缩率，压缩和解压时间，完成一个表格。

1 天以内完成优秀

5） 排序命令强化学习

自动生成多列文件 1G，第一列是 5 位随机字符串，第二列是 2 位数值，第三列是 5 位数值
1）按照字符串顺序排序整个文件。
2）首先按第一列排序，第一列字符串相同的情况下，按照第二列数值排序，第二列数值相同的情况下按第三列排序。
3）考虑如何并行排序这个 1G 文件，加快速度（切分文件，分别压缩，然后再归并）

2 天以内完成优秀

6） 安装 http 服务

并在服务目录创建 readme 文件，里面写上 hello world. 通过浏览器可以访问到自己开放的这个 http 服务，即打开 http://127.0.0.1/readme 看到这个服务，80 是默认端口，可以不用写成 http://127.0.0.1:80/readme

http 服务配置端口从 80 端口修改到 8080 端口，并且实现关机重启后服务自动启动

7） 安装 mysql 数据库服务，并进行常见的运维（导出，备份，还原）操作

1）并实现创建一个数据库（例如叫 testdb），并在数据库内创建一个表(score_table)，表内包含 id（整形），name（字符串），score（整形）这么几个字段，并且自己 insert 一些记录。

2）能够把创建的表 dump 出来，保存为 score_table.sql 文件。

3）把数据库里的 score_table 表删除了，用这个 score_table.sql 能恢复出原表出来。

4）学习如果将 score_table 表中，score（分数）在 90 分以上的同学记录导出来。

下面的学习需要有网络环境，网络环境可以这样得到

1） 创建两个 linux 虚拟机，互相可以通信

2）没有装虚拟机的同学，我们可以给一个远程机器，这样和自己实验的机器形成一个网络环境

8） 学习 iptables 服务**（选学）**

安装 iptables 服务，并完成 iptables 几个简单功能

1） 假定之前安装的 http 服务开在 8080 端口，在 iptables 里面增加一个配置，使得原来访问 readme 能看到 hello world, 配置后不可打开（相当于封了 8080 端口）

2） 把 httpd 服务配置在 80 端口上启动服务。netstat -anp | grep httpd 可以看到这个服务在 80 端口上服务。 然后做一个 nat 转发，将 8080 端口的请求转发到 80 端口上，这样打开 http://127.0.0.1:80/readme 和 http://127.0.0.1:8080/readme 看到的文件是一样的。

3）如果有两台 linux（两个 IP，可以是内网 IP，只要能互通就行），可以在一台机器（甲）上启动 httpd 服务（80 端口），然后把对另一台机器(乙)的 iptables 配置成 80 端口转发到甲的 80 端口上。这样是访问乙的 80 端口，和打开甲的 80 端口看到的内容一样的。

4）如果有三台 linux，可以做另外一个实验，甲，乙，丙。甲乙上开 httpd 服务，都启动在 80 端口。丙上用 iptables 把 80 端口的请求 50% nat 到甲，50% nat 到 乙上，实现负载分流。当然也可以在 2 台 linux 上做实验，其中在甲机器上启动两个 httpd 实例，一个启动在 80 端口，一个启动在 8080 端口。然后乙机器 50%的流量打给甲的 80 端口，50%的流量打给乙的 7070 端口。

9） 学习 NFS 服务**（选学）**

两个 Linux 机器（甲和乙），网络可以互通，将甲的硬盘远程 mount 到乙的机器上。并测试下这种远程硬盘的读写速度和本地硬盘读写速度的差异，找一个度量硬盘读写速度的工具，并对比这种差异，比如 dd 命令创建一个文件，在本地硬盘创建一个 1G 文件的时间和在远程硬盘（但 mount 在本地）上创建的时间的差距。

10） 学习 awk 命令，简单的字符串处理。awk 中需要了解的内容包括。

- BEGIN、END 的含义
- NR 和 FNR 处理两个文件
- 数组的理解（常用于计数器）

学习并理解这个命令：
假定有两个文件，分别是 account 和 cdr。例如张三的卡号是 000001，消费了 2 次，一次 10 元，一次 20 元。现在需要用 awk 命令得到张三和李四消费了多少元。

```
#cat account
张三|000001
李四|000002
#cat cdr
000001|10
000001|20
000002|30
000002|15
```

理解下面命令：

```
awk -F \| 'NR==FNR{a[$2]=$1;next}{sum[a[$1]]+=$2;}END{ for (var in sum) { print var"\t"sum[var];} }' count cdr
```

- awk 的内置函数 rand，gsub，substr，index，length，substr
- awk 里面用 system 函数执行一个脚本，system 也是 awk 里面一个内置函数，但是比较重要，这使得 awk 可以启动其他任何脚本命令的能力

awk 命令非常强大，可以很方便的做很多统计性工作，开发和运维都需要大量使用。

11） Linux 系统配置的学习

之前做 iptables 的 nat，大家应该接触过需要在 /etc/sysctl.conf 下配置：net.ipv4.ip_forward=1 才可能激活 nat 的功能。我们需要熟悉和了解一系列 linux 下的配置。以下是最常见的一些配置，大家上网找资料学习下，并掌握。

- 配置 linux 文件最大打开数（ulimit -n 可以看到当前系统默认值）

- 学习交换区的概念
  1） cat /proc/sys/vm/swappiness 了解下这里配置的含义

2） 内存不够的时候，不能增加内存需要增加自定义交换区，如何配置？
用 swapon，swapoff 增加和取消自定义交换区

3） 用 free 命令考察交换区使用情况

- DNS 的配置
在/etc/resolv.conf 中配置 dns。将本机的 DNS 解析使用 114.114.114.114，如何配置？配置好以后，怎么整明自己解析的 baidu.com 是通过 114.114.114.114 解析出来的，了解下 dig 命令。看看 dig baidu.com 返回什么？

- Yum 源的配置
通常机器默认 yum 源都很慢，往往在国外，尝试配置网易 yum 源，或者阿里云 yum 源。对比下载安装的速度差异。

- cat /etc/passwd
查看当前机器的所有用户，有时候判断机器是否被黑了，或者出现了不明登录用户就要用这个命令。了解 nologin 的含义。几个系统默认账号的含义。

- cat ~/.ssh/known_hosts
这个里面存放了什么？ /root/.ssh/id_rsa.pub 这里面存放的是什么？ 如果本机（比如叫机器 A）上没有，可以通过什么方式创建。有了 id_rsa.pub 后，怎么把这个东西存放到其他机器上（比如叫机器 B），使得本机（机器 A）可以 ssh 到那台机器 B 而不需要密码。

12） 学会使用 github

- 每个人在 https://github.com/ 上注册账号，上传一些自己的代码或者文档。学会 checkin checkout,clone,fetch,pull 等命令。

- 理解分支
什么是主分支，一个人的项目和多个人合作的项目会有很大区别，一个人的项目搞搞主分支就行了，但是多人合作的项目会有很大区别，特别是同一个部分可能有两个人一起开发。

- 理解 Tag
一个大的项目可能有多人合作开发，肯定会在一个合适的时候，大家都开发结束了，这个时候需要有项目管理人员做一个 Tag（往往一个 Tag 对应一个版本号），然后测试人员就取这个 tag 下的代码，其他人可以继续在主分支提交代码。测试人员测试 OK 后，运维人员就需要从这个 tag 上取代码下来，到线上环境去跑。

因此如果不做开发，分支用到的机会很少，但是 Tag 用到的机会几乎是百分之百的。