1：shell基础：

学过linux的都知道有shell这么个概念，shell是介于user与kernel之间的，user的操作或是指令经过shell解释传递给kernel，kernel就能够理解并执行相应的操作，我们为什么要学习shell呢？因为shell很重要，有人说过linux的一切都可以使用命令行操作，那么你所使用的命令行又是什么呢？shell不仅包括命令行操作的命令，也包括图形界面的软件，而且据我看来，你只要是干linux相关的，就要求会写一定的脚本，当然，你说我能用python，我只能呵呵。

查看系统能使用的shell与正在使用的shell：我们可以查看/etc/shells文件，一般操作系统默认的是/bin/bash，你可以直接echo $SHELL，既然使用的是bash那就说明它肯定有优点呗，那它的优点是什么：1：history，2：tab，3：alias，4：shell script，5：jobs，bg，fg，&，6：wildcard。那我们之前的linux有那么多的conmands难道都是shell内置的，of course not，我们可以通过type来查看命令的类型，它可能是alias，可能是外部命令，然后才有可能是来自shell内置的，我们可以通过type查看，如type –a ls。

变量，就是可以变化的量，是用一串字符来替代一些设置或数据，我们比较常见的PATH, SHELL, HOME等，变量的显示与设置：echo，unset，前者输出，后者取消，另外有一点要注意的是export的使用，将变量设置为环境变量，变量一般使用大写字母表示，还有一个子进程，我们使用bash之后，就相当于是打开了一个新的进程，父进程的变量无法使用在子进程中，除非使用export设置，单引号与双引号，env查看环境变量，set查看所有变量，$?，语系变量locale，read （-p），array，declare，变量的删除，替换，alias，history。

命令的执行：1：以相对或绝对的路径执行命令2：通过alias，3：bash内置命令4：通过$PATH，bash的登录欢迎界面/etc/motd，bash环境配置文件，login shell，取得bash时有完整的登录流程，non-login shell，这里比较典型的是图形界面登录，取得bash时不需要完整的登录。Login shell，每个用户登录到bash是都会执行/etc/profile，然后是/etc/inputrc，同时/etc/profile.d/\*.sh都会被执行，这里的LANG.sh会调用/etc/sysconfig/i18n，之后便到了用户~/.bash\_profile🡪~/.bashrc—>/etc/bashrc，假如是non-login的话直接会使用~/.bashrc。

通配符与特殊符号（这里一定要严格区分通配符与正则表达式完全不搭边）。数据流重定向<,<<, >,>>, 2>,2>>；命令执行的依据；&& ||，

ls /tmp/sun && echo “exist” || echo “not exist”，管道命令会处理标准输出，对于错误输出忽略，另外，他把输出结果当做标准输入，关键词搜索grep，基于列的处理cut：last | grep “root” | cut –d “ ” –f 1

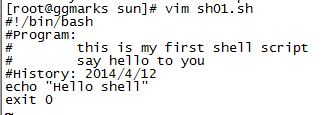
文本统计wc，文本排序sort，文本比较diff:last | cut –d “ ” –f 1 | sore | uniq –c；tee双向重定向 ls / | tee sun.txt | more，字符转化命令：tr，col，join，paste，expand。切割文件split。

2：正则表达式：通过一些特殊字符的排列，用以查找，替换，删除一行或多行文字字符串，是一种字符串处理依据，一种表示法，只要工具程序支持这个表示法，那么该工具程序就可以用来作为正则表达式的字符串处理了，RE分为基础正则表达式，扩展正则表达式，先直接指明语系，特殊符号[:alnum:], [:alpha:]，^, ., [-], [^], sed的使用，a,c,d,I,p，nl /etc/passwd | sed “2,5p”, 扩展正则表达式，egrep，

Grep –v “^$” file | grep –v “^#”，egrep –v “^$|^#” file, last –n 5 | awk ‘{print $1 “\t” $3}’.awk将一行分为若干个字段。

3：shell script：

Shell script就是利用shell功能写“程序”，这个程序主要使用shell命令，正则表达式，重定向，管道等功能达到我们处理的目的，有点类似于我们早期所写的bat文件，学习shell script的意义在于你可以实现自动化管理，配置防火墙之类，简单的数据处理，跨平台等诸多特性。Shell的编写与执行：从上到下从左到右，忽略空格，遇到enter就执行，“#”开头为注释；执行要看是相对路径与绝对路径或直接通过bash的进程来执行，编写第一个shell：



声明所使用后的shell，告知script的用途，历史，声明部分，执行部分，接下来就是你自己的练习了。