填空：

1. 修改当前用户登录密码的命令是： passwd 。
2. 查询当前用户登录名的命令是： who 。
3. 查询系统当前日期的命令是： date 。
4. 在shell下快速退出系统的快捷键是： ctrl-d(输入是exit) ，在shell下删除当前输入行的快捷键是： ctrl-u ，中断命令执行的快捷键是： Del(或者ctrl-c) 。
5. 查询c语言printf函数参数的命令是： man printf 。
6. 进入vi后，在命令模式下左移光标的命令是： h ，上移光标的命令是 k 。
7. 在vi命名模式下，在光标左侧插入字符的命令是： i ，在光标右侧插入字符的命令是： a 。
8. 在vi编辑文件完毕后，输入命令保存退出： :wq ，不保存直接退出的命令是： :q 。
9. 在vi中，删除一行的命令是： dd ，复制一行的命令是： yy 。
10. 在vi命令模式下，向文件尾部搜索hello字符串的命令是： ?hello ，向文件头部搜索hello字符串的命令是： /hello 。
11. 在vi命令模式下，取消最近修改的命令是： u ，重复最近一次修改的命令是： .(点) 。
12. 在shell下，回到主目录的命令是： cd 。
13. 在shell下，查看当前工作目录的命令是： pwd 。
14. 在shell下，在当前目录下创建子目录hello的命令是： mkdir hello ，创建一个空文件world的命令是 touch world ，删除空目录hello的命令是： rmdir hello 。
15. ls –al命令执行后，一行头部的第1个字符为d，表示它是 目录文件 ，头部字符为-表示它是 普通文件 ，头部字符是c表示它是 面向字符的特殊（设备）文件，如打印机 ，头部字符是b表示它是 面向块的特殊（设备）文件，如磁盘 ，头部字符是l表示 面向另一文件的键接 。
16. 在屏幕上输出hello文件内容的命令是： cat hello 。
17. 递归删除hello目录的命令是： rm-r hello 。
18. 在vi中，读取hello文件到光标后部的命令是： :r hello 。
19. 在vi中，在保存完当前文件后，编辑另一个文件的命令是： :e 。
20. 在vi中，进入virtual模式的命令是： v ，在virtual模式下拷贝选中内容的命令是 y ，将拷贝的内容粘贴到光标所在位置的命令是： p 。
21. 在vi中，将光标当前位置开始7行的内容拷贝到w寄存器的命令是： w7yy 。
22. 在vi中，拷贝从光标所在位置到行尾的内容到g寄存器的命令是： gy& 。
23. 在vi中，查看h寄存器内容的命令是： hp 。
24. 在vi中，向文件尾部翻半页的命令是： ctrl+d ，向文件头部翻半页的命令是： ctrl+u 。
25. 在vi命令模式下，将光标跳转到第1行的命令是： gg ，跳转到最后一行的命令是： G ，跳转到第10行的命令是： 10G ，跳转到最后一个字符的快捷键是： G$ 。
26. 让vi显示所有行的行号的命令是： :set number ，查看所有vi参数的命令是： :set all 。
27. 在vi中，切换到shell模式的命令是： :shell ，粘贴date命令执行结果的命令是： :r !date 。
28. 在vi中，使用正则表达式搜索所有头部为hello字符串的行是： /\<hello\> ，将所有world字符串替换为hello字符串的命令是： 。

:%s/"world"/"hello"/g

1. 匹配空行的正则表达式是： ，^\s\*\n

匹配行尾是.的正则表达式是： 。\w+[^(\.)]$

1. 利用cat命令从标准输入创建一个myfile文件的命令是： cat >myfile ，利用cat拷贝myfile成为myfile2文件的命令是： cat myfile > myfile2 ，利用cat将f1，f2合并为f3文件的命令是： cat f1 f2 > f3 ，利用cat将标准输入重定向到myfile3，并以EOF字符串表示输入结束的命令是： cat > myfile3 <<EOF 。
2. 将目录hello递归拷贝成为world目录的命令是： cp -r hello world 。
3. 修改一个文件hello为world文件的命令是： mv hello world 。
4. 查看/etc/passwd文件头部5行的命令是： head -5 /etc/passwd ，查看/etc/passwd文件尾部7行的命令是： tail -7 /etc/passwd 。
5. 利用/etc/passwd统计系统注册用户数目的命令是： wc -l /etc/passwd 。
6. 在用户主目录下的bin目录创建/usr/bin/gcc的符号链接： ln -s /usr/bin/gcc 。
7. 输出/etc/passwd所有用户名的命令： cat /etc/passwd 。
8. 匹配html、xml文件中标记<…>的正则表达式： \<.[^<>]\*\> 。
9. Markdown文件中# （#后跟一个空格）表示一级标题，## 表示二级标题，匹配所有标题的BRE正则表达式： ^#[# ] 。
10. 采用BRE表示匹配cat、cet的字符串： c[ae]t ，连续匹配3次的表达式： \{3\} 。
11. 在vi中删除C++源文件中所有//注释行的命令： grep -v "//" 。

命令组合：

1. 统计/dev目录下所有块设备的个数

ls -l |grep "^-"|wc -l /dev

1. 统计一个目录下所有.h和.c文件的行数，要求排除空行

find . \( -name "\*.h" -or -name "\*.c" \) |wc -l

1. 搜索/usr/include目录中，引用EPOLLIN常数的头文件  
   find /usr/include | xargs grep EPOLLIN

1. 搜索/exam目录下.txt文件，然后将以\r\n结尾的dos文本文件，修改为unix结尾的\n文件
2. 统计一个html文件中标记<…>个数的命令

grep -o '<.\*>' html | wc -l

1. 将一个markdown文件中的所有一级标题替换为二级标题的命令
2. 统计/usr/include/sys目录下的所有目录及子目录个数

ls -lR|grep “^-”|wc -l /usr/include/sys

1. 查找/usr/include/boost目录下的所有头文件，然后将之替换为/usr/local/include/boost路径，输出到header文件
2. 删除hello.h文件中的空行，保存到hello2.h

awk '/./' hello.h > hello2.h

简答：

1. 简述用户登录系统过程。

简述(linux)：

1.开机BIOS自检

2.MBR引导

3.grub引导菜单

4.加载内核kernel

5.启动init进程

6.读取inittab文件，执行rc.sysinit,rc等脚本

7.启动mingetty，进入系统登陆界面

1. 简述/etc/passwd的格式。



从文件中我们可以看到，/etc/passwd中一行记录对应着一个用户，每行记录又被冒号(:)分隔为7个字段，其格式和具体含义如下：   
**用户名:口令:用户标识号:组标识号:注释性描述:主目录:登录Shell**

1. ls –al得到一个文件的权限为drwxrwxr--，说明这个目录的权限。

第一列是文件类别和权限，这列由10个字符组成，第一个字符表明该文件的类型。d为目录。接下来的属性中，每3个字符为一组，

|  |  |
| --- | --- |
| 第2-4个字符 | 代表该文件所有者（user）的权限 |

，

|  |  |
| --- | --- |
| 第5-7个字符 | 代表给文件所有组（group）的权限 |

；

|  |  |
| --- | --- |
| 第8-10个字符 | 代表其他用户（others）拥有的权限 |





综上，此文件为user有读取+写入+执行权限，group有读取+写入+执行权限，

Others有读取权限。

1. tcouh一个文件后，是否为该文件分配了inode？分配了文件block？使用vim编辑后，mv移动到另一个目录下，inode是否发生改变？block是否改变？改变的是什么？ln成另一个文件，另一个文件是否分配了inode？分配了block？ln –s成另要给文件，是否分配了inode？是否分配了block？

<https://blog.csdn.net/dangpu/article/details/40020801>

<https://blog.csdn.net/chuojianpeng9484/article/details/101054052>

tcouh一个文件后，是否为该文件分配了inode？是

分配了文件block？否

1个inode，不分配block

使用vim编辑后，mv移动到另一个目录下，inode是否发生改变？block是否改变？改变的是什么？例如：移动/test5/testfile到/test6/

/test5 inode减少一个， /test6 inode增加一个，/test5 block应该不变，/test6应该减少一个 可能因为文件大小

ln成另一个文件，另一个文件是否分配了inode？分配了block？建立硬链接：inode和block都没有分配

ln –s成另要给文件，是否分配了inode？是否分配了block？建立符号链接：分配1个inode，不分配block