1.用命令完成以下功能（写出对应的命令，要求命令尽可能简单[能一行命令完成的尽量一行命令完成, 不能使用课上教的Shell脚本或者编程方法]）

a)将当前目录下所有文件备份并压缩成tar.gz，如果成功再将备份拷贝到/backup目录下。

tar zcvf backup.tar.gz \* && cp backup.tar.gz /backup/backup.tar.gz

b) 将上述备份压缩后的文件解压到原来的目录。

tar zxvf backup.tar.gz

c) 将当前目录下所有后缀为.html的文件的后缀改为.htm。

rename ‘s/\.html$/\.htm/’ \*.html

d）将当前目录下所有子目录下以a开头的.cpp文件改名为以b开头的.c文件。

find -name "a\*.cpp" -type f -exec rename 's#(.\*/)a([^/]\*).cpp#$1b$2.c#g' {} \;

2.自己查阅Linux相关书籍资料，学习并完成以下作业题：

a) 请简要叙述DOS/Windows中的文本文件，UNIX/Linux中的文本文件以及Mac中的文本文件有何不同之处，并给出至少一种你认为较为实用的转换方案。(需要两两给出转换方案)

三者的不同点主要是文件结尾换行符的不同。

Unix/linux:　　'\n',　　0x0A（换行字符LF）

Windows:　　‘\r\n’,　　0x0D0A回车字符CR，换行字符LF）

Mac OS:　　'\r',　　　0x0D（回车字符CR）

转换方案：

Windows->Linux : sed -i ‘s/\r$//’ new.txt

Linux->Windows : sed -i ‘s/$/\r/’ new.txt

Windows->Mac : cat win.txt | tr -d ‘\n’ > mac.txt

Mac->Windows : cat mac.txt | tr ‘\r’ ‘\n’ | sed 's/$/\r/' > win.txt

Mac->Linux : cat mac.txt | tr ‘\r’ ‘\n’ > win.txt

Linux->Mac : cat linux.txt | tr ‘\n’ ‘\r’ > mac.txt

b) 请查阅Linux系统中/etc/fstab文件各字段功能，并对你自己安装的Linux系统中/etc/fstab的各个字段做出解释。

/etc/fstab是用来存放文件系统的静态信息的文件。位于/etc/目录下，可以用命令less /etc/fstab 来查看，如果要修改的话，则用命令 vi /etc/fstab 来修改。当系统启动的时候，系统会自动地从这个文件读取信息，并且会自动将此文件中指定的文件系统挂载到指定的目录。

Ubuntu 的/etc/fstab文件主要包括6段

1. <file systems> - 要挂载的分区或存储设备.
2. <mount point> - 挂载点，也就是自己找一个或创建一个目录，然后把文件系统<file system> 挂到这个目录下，然后就可以从这个目录中访问要挂载的文件系统。对于 swap 分区，这个域应该填写 none，表示没有挂载点。
3. <type>指定挂载设备或是分区的文件系统类型，支持许多种不同的文件系统：：adfs, befs, cifs, ext3, ext2, ext, iso9660, kafs, minix, msdos, vfat, umsdos, proc, reiserfs, swap, squashfs, nfs, hpfs, ncpfs, ntfs, affs, ufs。
4. <options>挂载时使用的参数，选项：

auto: 开机自动挂载

          noauto: 开机不自动挂载

              defaults: 按照大多数永久文件系统的缺省值设置挂载定义

              ro: 按只读权限挂载

              rw: 按可读可写权限挂载

              user: 任何用户都可以挂载

              user: 同步磁盘与内存中的数据，async 则是异步

1. <dump> dump 工具通过它决定何时作备份. dump 会检查其内容，并用数字来决定是否对这个文件系统进行备份。允许的数字是 0 和 1 。0 表示忽略， 1 则进行备份。大部分的用户是没有安装 dump 的，对他们而言 <dump> 应设为0。
2. <pass> fsck 读取 <pass> 的数值来决定需要检查的文件系统的检查顺序。允许的数字是0, 1, 和2。 根目录应当获得最高的优先权 1, 其它所有需要被检查的设备设置为 2. 0 表示设备不会被 fsck 所检查。

c) 用命令行实现： 从光盘制作一个.iso文件，如果成功将光盘弹出。

cp /dev/cdrom ~/mydisk.iso && eject cdrom

附加题：

假设当前Linux系统的目录/tmp/gsl下的文件结构如下图所示，其中有很多Makefile分散在各个子目录下(但并非每个子目录都存在Makefile，例如block目录下就没有)。

.

|-- INSTALL

|-- Makefile

|-- NEWS

|-- ...

|-- blas

| |-- ChangeLog

| |-- Makefile

| |-- TODO

| |-- blas.c

| |-- ...

|-- block

| |-- ChangeLog

| |-- block.c

|-- bspline

| |-- Makefile

| |-- ...

|-- ...

要求将这些Makefile拷贝到目录/home/usrname/gsl的对应子目录下(假设/home/usrname/gsl下已经存在blas，bspline等子目录)，使得拷贝后/home/usrname/gsl下的文件如下图所示：

.

|-- Makefile

|-- blas

| |-- Makefile

|-- bspline

| |-- Makefile

|-- ...

\*要求：该要求实际上是可以用一行命令完成的，难度较高。同学们实际做作业时，允许使用两到三行命令，但不允许使用任何编程实现(即不允许Shell脚本的循环，分支等控制流程，可以使用管道)。

\*提示：利用find，sed和sh命令。

find /tmp/gsl -name 'Makefile'|awk’{a=$0;sed ‘s/\/tmp\/gsl/\/home\/usrname\/gsl’;system(“cp”a”$0)}’