Linux 操作系统 大作业 （2017 秋季学期 研究生）

注意事项：

1. 请把作业手写在答题纸上，不要计算机打印后贴在答题纸上
2. 答题纸请联系老师领取，不能自己准备答题纸
3. 禁止相互抄袭
4. 2018 年 1 月 12 日 （周五） 中午 1:30pm-2:00pm(过时不候)，将大作业交到 教 3-803 ， 可以找人代交。只交答题纸，不交本试卷。答题纸写好姓名、班级、学号
5. 任何问题 ， email to yuandong@bupt.edu.cn

1. Linux 基础命令(10 分)

合理利用指令，完成以下任务

（1）系统在 10 分钟后重启，并且通知所有当前登录用户，以使其能够保存工作

（2）统计当前目录下 example 文件夹下包含的子文件夹的个数（不包括子文件夹下的文件夹）

（3）随机选取 file.txt 中的 100 行，以第二列的数值排序，并存储到 sample.txt 中

请写出相关指令。

Make rational utilization of instruction, completing the following tasks：

（1）Reboot the system after ten minutes and inform all users logged on so that they can save their work

（2）Count the number of sub-directories in the *example* directory. Note that the directories in sub-directories should not be included.

（3）Randomly select 100 lines from the file *file.txt*, and save the lines in *sample.txt*

Write the related instructions here.

2. Crontab（10 分）

系统管理员可能每天都需要做一些重复工作，crontab 是一个十分有用的工具，请回答一下问题：

利用 crontab 完成以下任务：

<1>每天在 2:20 a.m. 删除 “/confbackup” 目录下所有的子目录和文件

<2>从上午 8 点到下午 2 点每小时读取/home/user/workspace 目录下的 log 文件中的全部数据加入到/backup 目录下的 bak.txt 文件内

A system administrator may have to do some repetitive work every day, while *crontab* is a very useful tool, you should answer following questions.

Complete the following tasks with the tool of crontab:

<1> Remove all the subdirectories and files in "/confbackup" directory at 2:20 a.m every day

<2> Read the log file on the */home/user/workspace* once an hour from 8:00 a.m. to 2:00 p.m.and add the data to the *bak.txt* in */backup* directory.

3. 简单 shell 编程（5 分）

（1）设计一个 shell 程序，添加一个新用户组（Group）为 students，添加属于这个组的 20 个用户，用户名的形式为 stuxx，其中 xx 从 01 到 20.（usradd）

（2）打印 ASCII 码的字符表，格式为：

```
Decimal  HEX  Character
----------------------
   33     21     !
   34     22     “
        ......
```

(1) Design a shell program to add a new group named students. Add 20 users to the group with the format of stuxx, where xx from 01 to 20

(2) Print the ASCII table with the follow format:

```
Decimal  Hex  Character
------------------------
33       21     !
34       22     “
```

4. Shell 编程（15 分）

编写shell程序处理四个人打牌的随机发牌过程（不包括两个Joker）。要求：

（1）用两个数组表示扑克牌的花色和点数；

（2）至少包含两个子函数，seed_random（生成随机数） deal_cards（发牌）

（3）按人打印每个人手里的13张牌

Program a shell script to deal with the random distribution of cards when four man playing the card, where the two jokers are not included. Hint：

(1) The suits and numbers of the cards should be stored in two arrays.

(2) Two of the sub-functions has to be included, including *seed_random* and *deal_cards*

(3) Print the 13 cards of each person.

5. Shell 编程（15 分）

(1) 有个文件名叫 file.txt，该文件有 100 行，请用至少三种方法取出该文件的第 23 行（比如 sed,tail 等）
(2) 给定一个文件 file.txt,统计该文件内单词的频率并按升序输出
(3) 某程序 启动命令为 ./start xxxx 其中 xxxx 为端口号.现后台运行着三个 start 进程，端口号分别为 20001,20002,20003. 需要编写 shell 程序作为它们的守护进程(某个端口进程退出后，只启动该端口的进程，请注意是后台运行)

(1) The file *file.txt* has 100 lines. Please take out the 23rd line of the files. At least provide 3 different methods.
(2) Count the frequency of the word of a given file file.txt and give an output in ascending order.
(3) The starting command of a program is **./start xxxx**, where **xxxx** is the port number. There are three start programs running at the background, with the port 20001, 20002, 20003. Please write a shell program as their guide-programming.(When the program of some certain port exits, the program will start the program on that port immediately. Doing all these on the background.)

6.Shell 编程，自动测试模型性能（15 分）

自动测试模型，并返回准确率结果最高的模型，参数如下： 模型的地址在./models/ 下面，模型文件后缀为 model,当然里面也有其他不是模型的文件 solverstate 后缀结尾， 模型的定义文件是 deploy.prototxt， 测试模型的命令是如下 ./bin deploy.prototxt models/xxx.model， 如果不是模型文件的话也要使脚本能够正确继续执行，测试下一个模型;测试时屏幕上会打印许多模型相关的参数的 log，log 中的最后一行会有准确率，比如"accuracy 0.998”。

Title description The script can automatically test the model and return the model with the highest accuracy result The address of the model is ./models/, the suffix of the model file is model; Of course there are other files like solverstate and so on.The model definition file is deploy.prototxt, and the command to test the model is as follows: /bin deploy.prototxt models/xxx.model You should also make sure that the script performs correctly even though it is not a model file; When the model is on test, it will print lots of model-related parameters on the screen, and in the last line shows accuracy, "accuracy 0.998"

7. Makefile 文件的编写（10 分）

Android NDK 是一系列编译工具的集合，可以帮助开发者快速开发 C/C++ 的动态库，并且可以自动将 so 和 java 应用一起打包成 apk 应用。开发人员只需要简单修改 mk 文件，就可以创建相应的.so 库。
查询并阅读 NDK 相关的资料，完成下列任务。

将 jni 文件夹下的 nonfree_init.cpp,sift.cpp, surf.cpp 三个文件编译并链接为动态库 libnonfree.so,供 Android 平台调用。三个文件以静态库的形式依赖于 OpenCV 模块。OpenCV 的 mk 文件位于
/home/user/OpenCV-android-sdk/sdk/native/jni/OpenCV.mk

请编写 makefile 文件 Android.mk 和 Application.mk，并给出编译指令生成动态库 libnonfree.so。编译好的动态库需要支持 c++11 特性，并支持 ARMv5 和 ARMv7 两种指令集。
（提示：以静态库的形式编译 OpenCV 依赖）

Android NDK is a set of compiling tools to help the developers to create the dynamic libraries in C/C++ and can automatically pack the .so and java applications into an APK. Only a slight change of .mk files will enable the developers to create the corresponding library.

Read the correlations for NDK and finish the following job.

Compiling and linking dynamically with 3 files, *nonfree_init.cpp,sift.cpp, surf.cpp* under *jni* folder to spare for Android. The library is named as libnonfree.so. The three files should rely on OpenCV with in a static way. The mk file of OpenCV is in
/home/user/OpenCV-android-sdk/sdk/native/jni/OpenCV.mk

Please write Android.mk and Application.mk and give the compile command to create libnonfree.so. The .so file should support C++11 with ARMv5 and ARMv7.

8. GCC,makefile （20 分）

Makefile 与 C/C++编程：本题考查 Makefile 的编写，C/C++语言基础，以及编程规范性。

编写 C 或 C++程序，完成：输入一个数组，输出该数组的全排列，要求：全排列不能有重复的。
例:【1，1，2】
输出：

[
【1，1，2】,
【1，2，1】,
【2，1，1】
]

要求：
(1) 建立四个文件。 main.cpp，function.cpp，function.h，Makefile；
(2) 用 C/C++语言实现，main.cpp 仅包括 main()函数，将输入的两个字符串用命令行参数传入到 main() 函数中

(3)编写 Makefile， 并使用 G++通过 Makefile 对你写的程序实现编译，连接，形成最终可以执行的文件，加分项：将 function.cpp 编译为动态库文件。

(4) 现有一个外部项目要和该项目整合，已知外部项目依赖 libboost_system.so,libopenblas.so,libglog.so,libopencv_highgui.so,libopencv_imgproc.so,libopencv_core.so,现已将所有的依赖库放到/usr/lib 文件夹下，头文件放在/usr/include 文件夹下，该外部项目还依赖多线程编译 openmp，请将外部项目和该项目代码整合成一个动态库,该外部项目代码在/tmp/src，头文件在/tmp/include 下,外部项目函数名字不会和该项目函数名字冲突

对不满足条件的酌情扣分。（考点：C/C++编程，Makefile 编写）

Write an ANSI C /C++ code. Use a Makefile to build it (including compile and link) into binary file.

The function of this binary file should be: Given a array that might contain duplicates, return all possible unique permutations.

or example,
[1,1,2]
Return

[
  [1,1,2],
     [1,2,1],
  [2,1,1]
]

You should give:

(1) Makefile (hint: four files, main.cpp，function.cpp，function.h, makefile).

(2) All C/C++ source code. main.cpp only passes parameter of input output and use the command line to pass the parameters to the main() .

(3) The whole procedure of how you compile and link C code.
(4) There is an external project to be mixed with the current project. The external project relies on *libboost_system.so,libopenblas.so,libglog.so,libopencv_highgui.so,libopencv_imgproc.so,libopencv_core.so,* all the libraries has been set in the folder /usr/lib, and the head files in /usr/include. The external project also relies on OPENMP for multi-thread processing. Please combine the two projects into a single dynamic library file. The source code for external project is in /tmp/src/ and the head file in /tmp/include. The function name in external project will not be conflict with the existing project.