竞赛题目候选

**对象：大一、大二、大三、大四、研究生**

**题目一：**通过分析Linux kernel源文件中的作者信息，分类统计贡献者的分布情况**（面向大一）**

**（1）课题要求**

【输入】Linux kernel 5.10.112源码树

可在以下URL或其他镜像站点下载：

https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.10.112.tar.xz

使用tar或winrar解压缩到本地文件系统（如在windows下解压报错，可直接忽略全部错误）

【输出】

1)贡献者统计文件：contribute.csv

csv格式文件

标题行：author(个体)、organize(所属组织)、kloc（源码行数/千行，kilo line of code)

2)记录无法识别的源文件列表的文件：unresolved.txt

普通文本文件

按字母序排序，带有相对路径的文件名，如下例

linux-5.10.112/include/linux/a.out.h

【算法要求】

1)源码统计方法

a)统计.c文件和.h文件

b)剔除空白行

c)以千行为单位，向上取整，如100行取整为1,000行，统计值是1

d)有多个作者的文件，按平均数分别统计到各个作者名下

2)贡献者的识别：

源文件中，作者的相关信息可能以多种方式如出，编程者需自行归纳各种模式，尽量提高识别率及准确度

例1，email地址，这种情况认为作者是'faith'、组织是'redhat.com'

\* Written by Rickard E. (Rik) Faith <faith@redhat.com>

例2, 文本形式，这种情况认为作者'Rusty Russell'，组织是'IBM'

Copyright (C) 2001 Rusty Russell, 2002 Rusty Russell IBM.

a)不能识别作者的，归到上述输出(2)文件中，不进行统计

b)可识别作者，但不能识别组织的，组织均标为'UNKNOWN'

**DCN部分（面向：大二及以上）**

**题目一：**基于CPLD/FPGA的智能风扇控制系统

**（1）题目要求：**

通过CPLD/FPGA系统板实现如下功能：

1、通过I2C接口读取板卡上sensor，获取温度值

2、根据获取的温度值来调整控制风扇转速

3、实时监控风扇的状态，包含在位、转速等参数

4、温度、风扇状态需要通过数码管或者状态LED进行状态显示，若还可通过串口传输到上位PC机则更佳。

**（2）题目说明：**

硬件环境：

淘宝购买温度传感器模块，小风扇，以及500～600左右的FPGA/cpld开发板即可。

1）功能1中的具体接口取决于温度传感器模块

2）功能2要求通过算法计算出风扇转速值（控制算法可以自由发挥，能够实现随着温度升高转速增加的要求即可），并将转速值转化为PWM值后输出到风扇转速控制引脚。举例：若转速最大值为255，则输出为全高电平，若为128则为一半高一半低，若停止则输出全低。

3）功能3要求读取风扇在位信号和风扇转速PWM值，并将PWM值转换为转速值进行显示。

**（3）评价标准：**

1）转速控制算法以及作品完成的完整度为主评分标准

2）完成方案设计，输出设计文档并完成代码编写和仿真为良好

3）实机上板测试演示调速效果（热风吹温度传感器，风扇转速快速响应增加转速）为优秀。

**路由和交换网络技术部分**

**题目一：**网络技术—Pipeline结构的Trie树最长匹配算法

**（1）题目背景：**最长匹配算法（longest prefix matching，LPM）是路由查找的基础算法之一，通常报文进入路由器后获取报文IP，报文IP送入LPM算法模块得出下一跳地址。其中基于Trie树的LPM是最经典的算法。如下图1所示，左上角为路由表信息，Trie树按照二叉树，构建出树形查找结构，黄色填充点为路由实节点，对应存在路由信息（具体原理可网上查看）。

实际上Trie的树形结构是链表存储，传统计算机实现Trie时，将Trie树的节点存储位于同一块硬件内存中，树节点的添加，删除，查找操作都是顺序执行，如下图2。

图1 Trie树算法 图2 CPU Trie存储结构

在查找IP：100时，需要访问RAM0四次，即访问四次后，才能进行下个IP的查找。为了提升查找速率，硬件设计需要将Trie树做成pipeline流水结构。如下图所示：



将Trie树节点散开分布到多级的pipeline流水中，IP查找时每个RAM都仅查找一次，以IP:100为例，RAM访问顺序为：RAM0->RAM1-> RAM2(访问不动作)-> RAM3-> RAM4。在IP：100查找访问完RAM0后，即可启动第二个IP的查找，因此可提升查找速率。

**（2）题目要求：**系统中有RAM0-63块内存，深度为8K，内存中每一行的数据结构自定义。实现IPV4 Trie树Pipeline结构的最长路由查表算法，包括路由表的添加、删除、查找操作，CPU模拟硬件流水结构和查找。注意Trie树节点布置的均衡性，流水线中的级数必须按顺序进行（RAM0->RAM1-> RAM2-> RAM3-> RAM4），但中间可跳跃（RAM0->RAM1-> RAM2(访问不动作)-> RAM3-> RAM4）。

输入：随机生成或以样本txt的格式给出路由表。格式为：

27 1 0 187 160

27 1 0 187 192

27 1 0 187 224

27 1 0 188 0

27 1 0 188 32

27 1 0 188 64

27 1 0 188 96

其中：27 1 0 187 160代表路由长度为27，IP前缀为：1.0.187.160

**（3）评判标准：**1、装表容量2、RAM操作次数3、RAM资源使用均衡

**题目二：**网络技术—TopK流量检测

**（1）背景说明：**网络环境中通常伴随着复杂的流量，有些流量巨大，通常称为大象流。网络处理设备中往往对大象流的处理比较复杂。如果能提前且快速的识别出大象流，针对大象流做相应的处理操作，能简化相关的处理流程，增强系统稳定性。如下图所示：



通常称Id0为大象流。

**（2）题目要求：**系统中有ID0-16383个ID流量，尽可能精确且实时地检测出流量排名前10（Top10）的ID号，计算复杂度低，计算量小，内存消耗低。避免为每个ID开设流量统计的方式进行设计。每输入一个报文，都需要实时快速更新前10名。

输入：随机或特定的流量模型输入。

格式为：Id0 Id0 Id1 Id2 Id3 Id0 Id0

**（3）评判标准：**1、计算量2、内存消耗3、检测精度

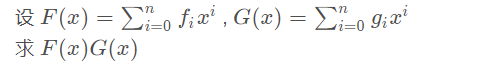
**移动通信技术部分（高年级包括研究生）**

**题目一：**低运算量高性能FFT设计

**（1）题目说明：**

评价移动通信好坏除了通信性能外，还有一个很重要的指标就是复杂度。由于复杂度的原因，很多逼近香农定理的优秀算法都无法得到实际的应用。FFT（Fast Fourier Transformation）是[离散](https://so.csdn.net/so/search?q=%E7%A6%BB%E6%95%A3&spm=1001.2101.3001.7020" \t "_blank)傅氏变换（DFT）的快速算法。即为快速傅氏变换。它是根据离散傅氏变换的奇、偶、虚、实等特性，对离散傅立叶变换的算法进行改进获得的。它是通信系统基本的时频域变化处理，是关键的一环。

FFT（[快速傅里叶变换](https://so.csdn.net/so/search?q=%E5%BF%AB%E9%80%9F%E5%82%85%E9%87%8C%E5%8F%B6%E5%8F%98%E6%8D%A2&spm=1001.2101.3001.7020" \t "_blank)）是DFT（离散傅里叶变换）的优化，用于加速多项式乘法。即：



**（2）题目要求：**

1. 实现1024、2048、4096点FFT/IFFT；
2. 复杂度(复乘数)低于1024\*log2(1024)、2048\*log2(2048)、4096\*log2(4096)
3. 在模型仿真上达到通信性能要求；在ARM v8架构平台验证通过；

**（3）评判方法:**

|  |  |  |
| --- | --- | --- |
| 评分项目 | 评分标准 | 权重 |
| 设计思路 | 创新性；数学推导完备性； | 30% |
| 设计性能 | 通信性能； | 30% |
| 模型资源 | 计算资源；处理延时； | 40% |

**题目二：**低运算量高性能DFT设计

**（1）题目说明：**

评价移动通信好坏除了通信性能外，还有一个很重要的指标就是复杂度。由于复杂度的原因，很多逼近香农定理的优秀算法都无法得到实际的应用。其中关于DFT，在工程应用中，得益于数字技术的应用，绝大多数傅里叶变换的应用都是采用离散傅里叶变换（DFT），更确切的说，是它的快速算法FFT。在5G通信系统中，由于dft-s-ofdm地峰均比等优势，在实际系统中DFT还是得到了应用，因此如何解决DFT复杂度问题，就变得尤其有意义。

**（2）题目要求：**

1. 实现的（k1,k2,k3为非负整数，）DFT/IDFT，使其复杂度(复乘数)低于N\*N
2. 在模型仿真上达到通信性能要求；在ARM v8架构平台验证通过

**（3）评判方法:**

|  |  |  |
| --- | --- | --- |
| 评分项目 | 评分标准 | 权重 |
| 设计思路 | 创新性；数学推导完备性； | 30% |
| 设计性能 | 通信性能； | 30% |
| 模型资源 | 计算资源；处理延时； | 40% |