

一：编程题

现有一组共计 N 个固定的集合 (N 为万量级), 每个集合有个从 0 开始递增的集合 ID, 每个集合包含 $1 \sim M$ 个 TERM (M 为 $0 \sim 100$ 的量级), 希望设计一个程序能够持续对外服务, 输入是一个 TERM 数组, 输出其中任意一个集合 ID (如果该 TERM 数组包含该集合的所有 TERM), 如果找不到输出 -1。要求:

- 1, 时间复杂度最优, 能够在短时间内对大量输入逐个输出
- 2, 实现具体的代码 (可以是伪代码), 其中常用的数据结构可以采用标准库。
- 3, 给出时间复杂度和空间复杂度。

TERM 组合集合的文件格式举例:

TERM_1 空格 TERM_2

TERM_1 空格 TERM_3

TERM_1 空格 TERM_3 TERM_4

输入的为 TERM 数组 (说明: TERM 为一个词, 可能是中文, 固定字符串表示)

二：算法题

你现在有一个文件, 文件中顺序存有 N 个记录, R_1, R_2, \dots, R_N , 这些记录不是有序的, 但是你知道一个整数 M , 这些记录满足 $R_1 < R_2 < \dots < R_M$ 以及 $R_{M+1} < R_{M+2} < \dots < R_N$.

- 1, 设计一个算法或编写一个程序, 将文件中的记录排序为 $R_1' < R_2', \dots, R_N'$, 算法或程序读取文件的次数为 $O(N)$, 不限内存使用,
- 2, 设计一个算法或编写一个程序, 将文件中的记录排序为 $R_1' < R_2' < \dots < R_N'$, 算法或程序读写文件的次数为 $O(N)$, 空间复杂度为 $O(1)$, (亦即, 你使用的内存大小和 M, N 均无关。)

三：设计题

- 1, 以下是一个简单的 Hello world 程序

```
# include<stdio.h>

Int main()
{
Printf( "hello world\n" );
Return 0;
}
```

编译生成 hello 后, 运行 ./hello;

会先 fork 一个子进程, 然后调用 execve 转载可执行程序 hello, 在调用 fork11 时采用了一种

叫做 COW(copy on write, 写时复制的策略), 这种思想不仅在内核中而且在应用程序中被广泛地采用。请描述下 COW 的思想, 以及它的实现。

2 请描述以下 fork clone 和 fork 的区别。

3 运行 ldd hello 可以得到如下结果

Linux-gate.so.1 (0xb7f4f000)

Lib.so.6 /lib/tls/i686/cmov/libc.so.6(0xb7dee000)

/lib/ld-linux.so.2(0xb7f50000)

其中 libc.so.6 是动态链接库, ld-linux.so.2 是动态链接库加载器, 请简要描述下动态链接库的加载过程和优点。

四: 只记下来一题 (似乎有 3 道小题, 任选一题做)

现在需要对 2000 台机器升级某个软件? 已经有这个软件的最新代码, 1: 你会选择用什么工具自动升级该软件? 请给出具体步骤或方法?

2: 为了便于后期的运维, 如果让你设计一套软件部署方案, 你会怎么设计?

关于我们



中国最先进的大学毕业生求职互动平台

更多海量资料下载, 敬请访问 www.dajie.com