

C++方向编程题答案

第八周

day43

题目ID: 797 --数据库连接池

链接: <https://www.nowcoder.com/questionTerminal/05f97d9b29944c018578d98d7f0ce56e>

[题目解析]

1. 服务器后台访问数据库时, 需要先连上数据库, 而为了连上数据库, 需要连接方和数据库服务上各自都要维护链接, 而链接本身的创建维护是有成本的
2. 要减少开销, 这里的解决方案就是将以前的mysql链接复用起来, 从而节省了创建和销毁的时间成本, 该题目就是想让你根据算法, 结合日志, 计算当前的业务需要预先创建多少链接, 来满足正常使用

[解题思路]

使用一个set来做, 如果某个记录是connect, 就加到set中去, 如果是disconnect, 就把set中对应的值删除掉。在加进去的过程中, 不断判断set元素的最大个数, 最终返回这个最大个数即可。

```
#include <iostream>
#include <fstream>
#include <algorithm>
#include <string>
#include <set>
using namespace std;

int main()
{
    int n;
    while (cin >> n)
    {
        set<string> pool;
        string id, con;
        int maxSize = 0;
        for (int i = 0; i < n; ++i)
        {
            cin >> id >> con;
            if (con == "connect") pool.insert(id);
            else if (con == "disconnect") pool.erase(id);
            int size = pool.size();
            maxSize = max(maxSize, size);
        }
        cout << maxSize << endl;
    }
    return 0;
}
```

[题目解析]

1. 该题目需要你了解 `mkdir -p` 命令，该命令可以递归创建目录
2. 但是该题目其实想让我们解决的问题是，给你一组数据（详情见题目），怎么用最少的 `mkdir -p` 创建出我们想要的多个目录

[解题思路]

只要按照题面，将前缀字符串路径和相同路径剔除，只 `mkdir -p` 最长或者唯一路径就可以解决这个问题

```
#include <iostream>
#include <fstream>
#include <algorithm>
#include <string>
#include <vector>
using namespace std;

int main()
{
    int n;
    while (cin >> n)
    {
        vector<string> list(n);
        vector<bool> flag(n, true);
        for (int i = 0; i < n; ++i) cin >> list[i];
        sort(list.begin(), list.end()); //将类似字符串经过排序，放在一起

        for (int i = 0; i < list.size() - 1; ++i)
        {
            // 1、两串相同
            // 2、前串是后串的子串，而且后串后一位是 '/'
            if (list[i] == list[i + 1]) flag[i] = false;
            else if (list[i].size() < list[i + 1].size() && \
                list[i] == list[i + 1].substr(0, list[i].size()) && list[i + 1]
                [list[i].size()] == '/')
                flag[i] = false;
        }

        for (int i = 0; i < list.size(); ++i)
            if (flag[i]) cout << "mkdir -p " << list[i] << endl;
        cout << endl;
    }
    return 0;
}
```