



1 编程:

用 C 语言实现一个 `revert` 函数，它的功能是将输入的字符串在原串上倒序后返回。

2 编程:

用 C 语言实现函数 `void * memmove(void *dest,const void *src,size_t n)`。`memmove` 函数的功能是拷贝 `src` 所指的内存内容前 `n` 个字节到 `dest` 所指的地址上。

3 英文拼写纠错:

在用户输入英文单词时，经常发生错误，我们需要对其进行纠错。假设（1）请描述你解决这个问题的思路；

- (2) 请给出主要的处理流程，算法，以及算法的复杂度；
- (3) 请描述可能的改进(改进的方向如效果，性能等等，这是一个开放问题)。

4 寻找热门查询：

搜索引擎会通过日志文件把用户每次检索使用的所有检索串都记录下来，每个查询串的长度为 1-255 字节。假设目前有一千万个记录，这些查询串的重复度比较高，虽然总数是 1 千万，但如果除去重复后，不超过 3 百万个。一个查询串的重复度越高，说明查询它的用户越多，也就是越热门。请你统计最热门的 10 个查询串，要求使用的内存不能超过 1G。

- (1) 请描述你解决这个问题的思路；
- (2) 请给出主要的处理流程，算法，以及算法的复杂度。

5 集合合并：

给定一个字符串的集合，格式如：{aaa bbb ccc}，{bbb ddd}，{eee fff}，{ggg}，{ddd hhh} 要求将其中交集不为空的集合合并，要求合并完成后的集合之间无交集，例如上例应输出 {aaa bbb ccc ddd hhh}，{eee fff}，{ggg}

- (1) 请描述你解决这个问题的思路；
- (2) 请给出主要的处理流程，算法，以及算法的复杂度
- (3) 请描述可能的改进(改进的方向如效果，性能等等，这是一个开放问题)。

||||||||||||||||||||||

1 题

```
char *revert(char * str)
{
```

```
int n=strlen(str);
int i=0;
char c;
for(i=0;i {
c=str;
str=str[n-i];
str[n-i]=c;
}
return str;
}

///////////
```

2 题

```
void * memmove(void *dest,const void *src,size_t n)
{
assert((dest!=0)&&(src!=0));
char * temp=(char * )dest;
char * ss=(char * )src;
int i=0;
for(;i {
    *temp =*ss ;
}
return temp;
}

///////////
```

3 题

(1)思路：字典以字母键树组织，在用户输入同时匹配

(2) 流程：

每输入一个字母：

沿字典树向下一层，

a) 若可以顺利下行，则继续至结束，给出结果；

b)若该处不能匹配，纠错处理，给出拼写建议,继续至 a)；

算法:

1.在字典中查找单词

字典采用 27 叉树组织,每个节点对应一个字母,查找就是一个字母

一个字母匹配.算法时间就是单词的长度 k.

2.纠错算法

情况:当输入的最后一个字母不能匹配时就提示出错,简化出错处理, 动态提示

可能 处理方法:

(a)当前字母前缺少了一个字母: 搜索树上两层到当前的匹配作为建议;

(b)当前字母拼写错误: 当前字母的键盘相邻作为提示; (只是简单的描述,
可 以有更多的)

根据分析字典特征和用户单词已输入部分选择(a),(b)处理

复杂性分析: 影响算法的效率主要是字典的实现与纠错处理

(a) 字典的实现已有成熟的算法, 改进不大, 也不会成为瓶颈;

(b)纠错策略要简单有效 ,如前述情况, 是线性复杂度;

(3)改进

策略选择最是重要，可以采用统计学习的方法改进。

||||||||||||||||||||||||||||||

4 题

(1)思路: 用哈希做

(2) 首先逐次读入查询串，算哈希值，保存在内存数组中，同时统计频度（注意值与日志项对应关系）my.chinahrlab.com 选出前十的频度，取出对应的日志串，简单不过了。哈希的设计是关键。

||||||||||||||||||||||||||||||||||

5 题

(1) 思路：先将集合按照大小排列后，优先考虑小的集合是否与大的集合有交集。有就合并，如果小集合与所有其他集合都没有交集，则独立。独立的集合在下一轮的比较中不用考虑。这样就可以尽量减少字符串的比较次数。当所有集合都独立的时候，就终止。

(2) 处理流程：

1. 将集合按照大小排序，组成集合合并待处理列表

2. 选择最小的集合，找出与之有交集的集合，如果有，合并之；如果没有，则与其它集合是独立集合，从待处理列表 中删除。

3. 重复直到待处理列表为空

算法： 1。将集合按照大小从小到大排序，组成待处理的集合列表。 2。取出待处理集合列表中最小的集合，对于集合的每个元素，依次在其他集合中搜索是否有此元素存在：

1>若存在，则将此小集合与大集合合并，并根据大小插入对应的位置。转 3。

2>若不存在，则在该集合中取下一个元素。如果无下一个元素，即所有元素都不存在于其他集合。则表明此集合独立，从待处理集合列表中删除。并加入结果集合列表。转 3。

3。如果待处理集合列表不为空，转 2。

如果待处理集合列表为空，成功退出，则结果集合列表就是最终的输出。

算法复杂度分析：

假设集合的个数为 n ，最大的集合元素为 m 排序的时间复杂度可以达到 $n \log(n)$ 然后对于元素在其他集合中查找，最坏情况下为 $(n-1) * m$ 查找一个集合是否与其他集合有交集的最坏情况是 $m * m * (n-1)$ 合并的时间复杂度不会超过查找集合有交集的最坏情况。所以最终最坏时间复杂度为 $O(m * m * n * n)$

需要说明的是：此算法的平均时间复杂度会很低，因为无论是查找还是合并，都是处于最坏情况的概率很小，而且排序后优先用最小集合作为判断是否独立的对象，优先与最大的集合进行比较，这些都最大的回避了最坏情况。

(3)可能的改进：

首先可以实现将每个集合里面的字符串按照字典序进行排列，这样就可以将查找以及合并的效率增高。另外，可能采取恰当的数据结构也可以将查找以及合并等操作的效率得到提高。

.....

1) 此题 10 分

对任意输入的正整数 N ，编写 C 程序求 $N!$ 的尾部连续 0 的个数，并指出计算复杂度。如： $18! = 6402373705728000$ ，尾部连续 0 的个数是 3。 (不用考虑数值超出计算机整数界限的问题)

2) 此题 10 分 编写一个 C 语言函数，要求输入一个 url，输出该 url 是首页、目录页或者其他 url

如下形式叫做首页：

militia.info/
www.apcnc.com.cn/
<http://www.cyjzs.com>www.greena888.com/
www.800cool.net/
<http://hgh-products.my-age.net>/

如下形式叫做目录页：

thursdaythree.net/greenhouses--gas-global-green-house-warming/
<http://www.mw.net.tw/user/tgk5ar1r/profile>/
<http://www.szeasy.com/food/yszt/chunjie>/
www.fuckingjapanese.com/Reality/

请注意：

- a) url 有可能带 http 头也有可能不带
- b) 动态 url (即含有"? "的 url) 的一律不算目录页，如：

www.buddhismcity.net/utility/mailit.php?l=/activity/details/3135/
www.buddhismcity.net/utility/mailit.php?l=/activity/details/2449/

另：如果你会 linux，请用 linux 下的 grep 命令实现第 2 题的功能（附加 5 分）。

3) 此题 40 分

如果必须从网页中区分出一部分"重要网页" (例如在 10 亿中选 8 亿)，比其他网页更值得展现给用户，请提出一种方案。

4) 此题 40 分

假设有 10 亿网页已经被我们存下来，并提供如下信息：网页全文（即网页的源码）、全文长度、网页正文（即网页中提取的主体文字）、正文长度，以及其他网页提取物等，现在希望去掉其中的重复网页，请提出可行的方案，计算出每个网页对应的重复度，你可以自己对网页重复下定义，也可以提出需要哪些更多的网页提取物来实现更好的去重复方案。

.....

好久没来了。

发生了一些事情，其间的心情已不是几行文字所能表述的了。

终于明白有些事情，并不是自己努力就一定能圆满的；有些事情，是我控制不了的。

唉，不提也罢！

说说今天去百度笔试的经历吧

部门：[百度](#)搜索应用技术部。

地点：海淀南路银科大厦（海淀图书城西临）**18** 层。

时间：**2005/6/15 10: 00-11: 20 am**

九点从实验室出发，725 到知春路，转 735，到海淀桥下车，9: 50 到达百度。在 725 的车上碰到一男士索要手机号，说自己认识信息产业部的部长杨泽民先生，以居高临下的姿态把手机号给了他—！

在百度前台见到了一直帮我安排笔试的杨韫敏 jj，不是想象中的 HR 形象，而是一副干练的女 IT 的样子[跳跃的灵魂](#)很快，给我找了一间小会议室，只有一张桌子，两把椅子，还帮我开了灯，关门，走人，我开始看题。冷汗

也开始流。翻了一下三页纸的笔试题，只有很少的传说中的 Linux 题目，其他的全是 C、数据结构、算法编程的题。第一反应：走人！但又觉得对不起陈 jj，关键的是我已经在笔试题上写了姓名和学校了，sign，总的为自己的名字和学校负责吧，他们是无辜的。如此斗争良久，决定坚持下来。

题目大致是这样的：

第一部分选择题：有几道网络相关的题目，巨简单，比如第一题是 TCP、RIP、IP、FTP 中哪个协议是传输层的……。有一道 linux 的 chown 使用题目。其他的全是数据结构的题目！什么链，表，码的，不知所云跳跃的灵魂唉，我可以没有学过数据结构的人呐！真残忍！这一部分迅速猜完！

第二部分简答题：

1、在 linux 中如何编译 C 程序，使之成为可执行文件？如何调试？

答案：

1)检查程序中.h 文件所在的目录，将其加入系统 PATH 中；

2)执行 C 编译：#gcc [源文件名] -o [目标文件名]

执行 C++ 编译：#g++ [源文件名] -o [目标文件名]

3)改变目标文件为可执行文件：#chmod +x [目标文件名]

4)如需将多个可执行文件连续执行，可生成批处理文件：

#vi [批处理文件名]

可执行文件 1

可执行文件 2

.....

最后将该批处理文件属性该位可执行。

调试：在编译时使用-g 参数，就可以使用 gdb 进行调试。

2、写出内存分配和释放的函数，并指出区别。

答案：

C 语言的标准内存分配函数： malloc, calloc, realloc, free 等。

malloc 与 calloc 的区别为 1 块与 n 块的区别：

malloc 调用形式为(类型*)malloc(size): 在内存的动态存储区中分配一块长度为“size”字节的连续区域，返回该区域的首地址。

calloc 调用形式为(类型*)calloc(n, size): 在内存的动态存储区中分配 n 块长度为“size”字节的连续区域，返回首地址。

realloc 调用形式为(类型*)realloc(*ptr, size): 将 ptr 内存大小增大到 size。

free 的调用形式为 free(void*ptr): 释放 ptr 所指向的一块内存空间。

C++ 中为 new/delete 函数。

3、写出 socket 函数，并指出其功能。

socket(): 建立 socket 通信描述符；

bind(): 将套接字和机器上的一定的端口关联；

connect(): 连接到远程主机；

listen(): 使套接字做好连接的准备，规定等待服务请求队列的长度；

accept(): 接受连接，一旦有客户端发出连接，accept 返回客户地址

信息和一个新的 **sock**;

有了这个新的 **sock**, 双方就可以开始收发数据:

send() 和 **recv()**: 用于流式套接字或者数据套接字的通讯;

sendto() 和 **recvfrom()**: 用于无连接的数据报套接字;

close(): 关闭套接字;

shutdown(): 选择性的关闭套接字, 可以只允许某一方向的通讯关闭;

getpeername(): 返回流式套接字时对端 **peer** 信息;

gethostname(): 返回程序所运行的机器的主机名字;

gethostbyname(): 返回本机 IP;

第三部分编程题:

1、从文件中读取字符串数据, 反序显示并大小写转换。

2、给定 26 字母表以及对应的密码表, 编程实现加密及解密功能。

第四部分思考题(正是传说中的字典纠错题):

用户在输入英文单词时经常出错, 现对其进行纠错。给定一个正确的英文词典, 考虑纠错实现。1)指出思路。2)流程、算法难易程度及可能的改进策略。

不过陈 jj 没有给我答题纸, 只好拿试题的背面做了答题纸兼草稿纸-__-!
说实话有些题目是很基础的, 就是没背过。不知怎么搞得, 巨潦草。实验室参加过笔试的通同学都是憋着劲做了两个多小时才答完, 而我只一个小时就完了, 唉, 正好说明肚子里只有别人一半的东西~~看着潦草而不着边际的答

题，决定在最后给陈 jj 写段话，大意就是感谢她帮我挽回了一次笔试的机会，但我的表现很遗憾等等……然后交卷走人~~

没想到交了试卷没让我走，等了大约 30 分钟的样子，有个很深沉的 gg 来看我的卷子跳跃的灵魂-__-!我颤颤的跟他说我很长时间没有接触 C 了，当时脖子都红了，真觉得丢人。gg 看了一下，没有很鄙视的样子，问我有没有带简历。当然木有啦~~答应回来给他发个电子版的，然后赶紧跑人了！

题目大致是这样的：

第一部分选择题：

有几道网络相关的题目，巨简单，比如第一题是 TCP、RIP、IP、FTP 中哪个协议是传输层的……。有一道 linux 的 chown 使用题目。其他的全是数据结构的题目！什么链，表，码的，不知所云。唉，我可以没有学过数据结构的人呐！真残忍！这一部分迅速猜完！

第二部分简答题：

1、在 linux 中如何编译 C 程序，使之成为可执行文件？如何调试？

答案： 1)检查程序中.h 文件所在的目录，将其加入系统 PATH 中；

2)执行 C 编译：#gcc [源文件名] -o [目标文件名]

执行 C++ 编译：#g++ [源文件名] -o [目标文件名]

3)改变目标文件为可执行文件：#chmod +x [目标文件名]

4)如需将多个可执行文件连续执行，可生成批处理文件：

#vi [批处理文件名]

可执行文件 1

可执行文件 2

.....

最后将该批处理文件属性该位可执行。

调试：在编译时使用-g 参数，就可以使用 gdb 进行调试。

2、写出内存分配和释放的函数，并指出区别。

答案：

C 语言的标准内存分配函数： malloc, calloc, realloc, free 等。

malloc 与 calloc 的区别为 1 块与 n 块的区别：

malloc 调用形式为(类型*)malloc(size): 在内存的动态存储区中分配一块长度为“size”字节的连续区域，返回该区域的首地址。

calloc 调用形式为(类型*)calloc(n, size): 在内存的动态存储区中分配 n 块长度为“size”字节的连续区域，返回首地址。

realloc 调用形式为(类型*)realloc(*ptr, size): 将 ptr 内存大小增大到 size。

free 的调用形式为 free(void*ptr): 释放 ptr 所指向的一块内存空间。

C++ 中为 new/delete 函数。

3、写出 socket 函数，并指出其功能。

socket(): 建立 socket 通信描述符；

bind(): 将套接字和机器上的一定的端口关联；

connect(): 连接到远程主机；

listen(): 使套接字做好连接的准备，规定等待服务请求队列的长度；

accept(): 接受连接，一旦有客户端发出连接，**accept** 返回客户地址信息和一个新的**sock**；

有了这个新的**sock**，双方就可以开始收发数据：

send()和**recv()**: 用于流式套接字或者数据套接字的通讯；

sendto()和**recvfrom()**: 用于无连接的数据报套接字；

close(): 关闭套接字；

shutdown(): 选择性的关闭套接字，可以只允许某一方向的通讯关闭；

getpeername(): 返回流式套接字时对端**peer** 信息；

gethostname(): 返回程序所运行的机器的主机名字；

gethostbyname(): 返回本机 IP；

第三部分编程题：

1、从文件中读取字符串数据，反序显示并大小写转换。

2、给定 26 字母表以及对应的密码表，编程实现加密及解密功能。

第四部分思考题(正是传说中的字典纠错题)：

用户在输入英文单词时经常出错，现对其进行纠错。给定一个正确的英文词典，考虑纠错实现。**1)**指出思路。**2)**流程、算法难易程度及可能的改进策略。

一道算法题目答案

```
int Replace(Stringtype &S, Stringtype T, Stringtype V); // 将串 S 中所有子串 T  
替换为 V，并返回置换次数
```

```

{
for(n=0,i=1;i <=Strlen(S)-Strlen(T)+1;i++) //注意 i 的取值范围
if(!StrCompare(SubString(S,i,Strlen(T)),T)) //找到了与 T 匹配的子串
{ //分别把 T 的前面和后面部分保存为 head 和 tail
StrAssign(head,SubString(S,1,i-1));
StrAssign(tail,SubString(S,i+Strlen(T),Strlen(S)-i-Strlen(T)+1));
StrAssign(S,Concat(head,V));
StrAssign(S,Concat(S,tail)); //把 head,V,tail 连接为新串
i+=Strlen(V); //当前指针跳到插入串以后
n++;
}//if
return n;
}//Replace

```

分析:`i+=Strlen(V);`这一句是必需的,也是容易忽略的.如省掉这一句,则在某些情况下,会引起不希望的后果,虽然在大多数情况下没有影响.请思考:设
`S='place', T='ace', V='face'`,则省掉 `i+=Strlen(V);`运行时会出现什么结果?
(无限递归 `face`)

百度 2005 年的笔试题

1. 实现 `void delete_char(char * str, char ch);`

把 `str` 中所有的 `ch` 删掉

2. 把字符串 `S` 中所有 `A` 子串换成 `B`,这个没给函数原型

3. 搜索引擎的日志要记录所有查询串,有一千万条查询,不重复的不超过三百

万

要统计最热门的 10 条查询串. 内存<1G. 字符串长 0-255

(1) 主要解决思路 //具体用词和原题不大一样

(2) 算法及其复杂度分析

4. 有字典,设计一个英文拼写纠正算法 (1) 思想 (2) 算法及复杂度 (3) 改进

5. $\{ \text{aaa}, \text{bb}, \text{ccc}, \text{dd} \}$, $\{ \text{bbb}, \text{ff} \}$, $\{ \text{gg} \}$ 等一些字符串的集合

要求把交集不为空的集合并起来,如上例会得到 $\{ \text{aaa}, \text{bb}, \text{ccc}, \text{dd}, \text{ff} \}$, $\{ \text{gg} \}$

(1) 思想 (2) 算法及复杂度 (3) 改进

.....

一、选择题: 15 分 共 10 题

1. 一个含有 n 个顶点和 e 条边的简单无向图, 在其邻接矩阵存储结构中共有
_____个零元素。

- A. e B. $2e$ C. $n^2 - e$ D. $n^2 - 2e$

2. _____是面向对象程序设计语言中的一种机制。这种机制实现了方法的定义
与具体的对象无关, 而对方法的调用则可以关联于具体的对象。

- A. 继承 (Inheritance) B. 模板 (Template)
C. 对象的自身引用 (Self-Reference) D. 动态绑定 (Dynamic Binding)

3. 应用层 DNS 协议主要用于实现 网络服务功能.

- A. IP 地址到网络设备名字的映射 B. IP 地址到网络硬件地址的映射
C. 网络设备名字到 IP 地址的映射 D. 网络硬件地址到 IP 地址的映射

4. linux 默认情况下, 一个进程最多能打开多少文件?

- A. 64 B. 128 C. 512 D. 1024

5. 下面结构体

```
struct s1 {
```

```
char ch, *ptr;
union {
    short a, b;
    unsigned int c:2, d:1;
}
struct s1 *next;
};
```

的大小是_____:

- A. 12 字节 B. 16 字节 C. 20 字节 D. 24 字节

6.任何一个基于"比较"的内部排序的算法，若对 6 个元素进行排序，则在最坏情况下所需的比较次数至少为_____。

- A. 10 B. 11 C. 21 D. 36

7.以下不是进程间通讯的是_____

- A 共享内存 B 信号量 C 线程局部存储 D 消息队列

8.下面程序，求 count 的值

```
int func(x)
{
    int count= 0;
    x=9999;
    while(x)
    {
        Count++;
        x = x&(x-1);
    }
    return count;
}
```

- A 8; B 10; C 5; D 11

9.使用 malloc 系统调用分配的内存是在_____上分配的?

- A 栈; B bss; C 物理内存; D 堆

10. 最坏情况下，合并两个大小为 n 的已排序数组所需要的比较次数_____

- A. $2n$ B. $2n-1$ C. $2n+1$ D. $2n-2$

二、简答题：20 分，共 3 题

1. (5 分) 下面这段代码是把中英文混合字符串(汉字用两个字节表示，特点是第一个字节的最高位为 1)中的大写字母转化为小写字母，请找出其中的 bug，注意各种异常情况。

```
for (char *piterator = szWord; *piterator != 0; piterator++)
{
    if (*piterator & 0x80 != 0)
    {
        piterator++;
    }
    else if (*piterator >= 'A' && *piterator <= 'Z')
        • piterator += 32;
}
```

2. (5 分) 对给定的上亿条无序的 url，请按照 domain、site 以及 path 分别排序，并请指出排序过程中可能会遇到的哪些问题？如何提高效率？

例如：<http://www.baidu.com/path/about.html>, domain、site 以及 path 的定义分别如下：

Domain:baidu.com

Site:www.baidu.com

Path: www.baidu.com/path

3. (10 分) 某型 CPU 的一级数据缓存大小为 16K 字节，cache 块大小为 64 字节；二级缓存大小为 256K 字节，cache 块大小为 4K 字节，采用二路组相联。经测试，下面两段代码运行时效率差别很大，请分析哪段代码更好，以及可能的原因。

为了进一步提高效率，你还可以采取什么办法？

A 段代码

```
int matrix[1023][15];
const char *str = "this is a str";
int i, j, tmp, sum = 0;

tmp = strlen(str);
for(i = 0; i < 1023; i++) {
    for(j = 0; j < 15; j++) {
        sum += matrix[i][j] + tmp;
    }
}
```

B 段代码

```
int matrix[1025][17];
const char *str = "this is a str";
int i, j, sum = 0;

for(i = 0; i < 17; i++) {
    for(j = 0; j < 1025; j++) {
        sum += matrix[j][i] + strlen(str);
    }
}
```

三、编程题：30 分 共 1 题

注意：要求尽可能提供完整代码，如果可以编译运行酌情加分。

1. 内存中有一个长数组，条目数为 10 万，数组单元为结构体 struct array，
sizeof(struct array) 为 512 字节。结构有一 int 型成员变量 weight。现需要取得按 weight 值从大到小排序的前 500 个数组单元，请实现算法，要求效率尽可能高。

四、设计题：35 分 共 1 题

注意：请尽可能详细描述你的数据结构、系统架构、设计思路等，建议多写一些伪代码或者流程说明。

1. 请设计一个字典。以字符串为索引，存储用户定义的定长结构。要求有增、删、查、改的功能。已经给定一个函数，可以由字符串映射到一个签名，每个签名由两个 `unsigned int` 类型组成。假设每一个字符串能够对应唯一的一个签名，完全没有重复（或者重复的概率可以忽略），并且签名分布足够均匀。

请描述你的数据结构？内存如何申请？增、删、查、改的功能如何实现？如果操作很频繁，该如何优化？

.....

经过在线笔试、两轮电话面试，今天上午收到了百度的拒信，我的百度求职算是告一段落了……

从[百度](#)校园招聘开始，我就投了一份简历。在别人都有在线笔试机会的时候，我却没有任何消息。

听说师兄可以给推荐，我就又通过内部推荐的方式投递了一次，这次很快就有消息了——拒信。

那时基本上就放弃了百度。可是大概 20 多天以后，我投递的第一份简历有消息了——通知我在线笔试。经过精心准备，笔试题答得还凑合。过了几天给我来了封邮件告知我笔试通过，会找时间安排电话面试。又过了好几天，我正在剃头的时候接到百度电话，约了第二天下午三点电话面试。心情异常兴奋，回到寝室拼命复习数据结构并收集百度面试题型……临阵磨枪 呵呵。

百度的面试氛围很是轻松，让你很快就觉得是在聊天而不是面试。第一轮主要是在针对我的在线笔试的题目进行提问和分析，主要讲的是做题的思路和改进的方法。面试时间大概有半个小时，觉得应该有下一轮。

果然上个星期五晚上接到了百度技术经理的电话，自称姓刘。在前一天我同学也是这个时候接到百度第二面电话，看来是同一个人，后来的面试内容证实了是同一个人。面试过程大概如下：

- 1、介绍一下项目。
- 2、提了一个问题：上千万条记录，统计出重复记录最多的前 N 条。
- 3、一个概率题：54 张扑克牌，除去两张大小王剩下 52 张扑克牌。问红桃 A 和黑桃 A 同时被一个人拿到的概率是多少？
- 4、多个线程访问共享内存时因该怎么办？
- 5、在写程序遇到问题的时候，通常采用什么调试方法？
- 6、一个 client/server 的协议问题
- 7、剩下就是随便聊聊，比如有缺点、期望工作的性质、职业规划等

总结一下教训：

- 1、介绍项目的时候不能一味的按照事前想好的模板说，应该根据所申请的工作的性质，多说一些和自己申请的工作内内容相近的东西说。我在介绍我的项目的时候，说了很多硬件的东西，而相关的 Linux 下的 C 编程却没有提到多少，一大失败之处。

2、对于他提的第二个问题，当时因为紧张没有想出来，挂了电话以后才有了思路。

3、这个概率题以前碰到过，而且和同学们讨论过，答案很早就知道了。但是遇到面试的时候，不能马上就说出答案，因为这样摆明了告诉人家你以前就见过这道题，这样就失去了作为考题的意义。所以，如果事前知道答案也不要马上说出来，装作考虑中，然后慢慢说出答案。我就是很快就说出了答案，失败！

4、在问项目的时候，他问我代码行大概有多少？我说大概有 5.6K 行左右。在回答第四个问题的时候，我几乎是将书上所讲过的东西背了一遍给他，虽然答案是正确的，但是我估计他一听就听出来是在背书了，所以这也会减分不少。而且百度强调创新，其实就算你不知道答案也可以按照自己的思路说一下的，只要逻辑清晰、合理都会比我背书强……

5、我的回答是有时候用 `gdb`，有时候用输出日志的形式。以我之前给他讲的项目经验是不大可能会涉及这么多的知识的，所以估计他又听出我是在背书了……继续减分

6、后来我发现这个问题其实他不是在考我问题的答案，是考我解决问题的能力和考虑问题的思路。这点是我比较差的地方，没办法……减分
我前面表现那么失败，基本上已经没有什么希望了，后面的谈话已经没有意义了，只不过是礼貌性的结束这次面试了。

上面的总结是我收到拒信以后才总结出来的，还以为能被录取呢……

面试官太和蔼了，而且气氛及其融洽，根本没有任何不好的征兆，面试官好厉害！

至此，我的**百度**求职过程到此告一段落……生活还在继续，工作还得继续努力去找，加油！

百度电话面试题目：

- 1.谈谈你对数据库中索引的理解
- 2.现在普通关系数据库用得数据结构是什么类型的数据结构
- 3.索引的优点和缺点
- 4.**session** 和 **cache** 的区别是什么
- 5.如果有几千个 **session**，怎么提高效率
- 6.**session** 是存储在什么地方，以什么形式存储的。