



## 2015网易游戏校园招聘笔试题游戏插件研发岗

### 一. 单项选择题

1. tcp三次握手创建连接，双方交互的报文中SYN和ACK的序列是什么样的（）

- ☐ A SYN, SYN+ACK, ACK
- ☐ B SYN, ACK, SYN, ACK
- ☐ C SYN+ACK, ACK, SYN
- ☐ D SYN, SYN, ACK

2. 函数参数使用的空间是在（）中申请的，malloc或new是在（）中申请空间的？

- ☐ A 堆，栈
- ☐ B 栈，堆
- ☐ C 栈，栈
- ☐ D 堆，堆

3. 有B+Tree/Hash\_Map/STL Map三种数据结构。对于内存中数据，查找性能较好的数据结构是（），对于磁盘中数据，查找性能较好的数据结构是（）。

- ☐ A Hash\_Map/B+Tree
- ☐ B STL\_Map/B+Tree
- ☐ C STL\_Map/Hash\_Map
- ☐ D B+Tree/Hash\_Map

4. 由源代码生成可执行文件需要经过预编译，编译，汇编，链接等阶段，错误：unresolved external symbol BeginScene属于()阶段错误。

- ☐ A 预编译
- ☐ B 编译
- ☐ C 汇编
- ☐ D 链接

### 二. 多选选择题

5. 下面属于进程间通信的有？

- ☐ A 管道
- ☐ B 消息队列
- ☐ C 内存共享
- ☐ D 套接字

6. 下面关于ISO网络参考模型分层及每一层功能描述错误的有？

- ☐ A 物理层，在此层将数据分帧，并处理流控制



- B 数据链路层，为物理层提供连接，以便透明的传送比特流
- C 网络层，本层通过寻址来建立两个节点之间的连接，为源端的运输层送来的分组，选择合适的路由和交换节点
- D 运输层，常规数据递送 – 面向连接或无连接
- E 会话层，在两个节点之间建立端连接。
- F 表示层，主要用于处理两个通信系统中交换信息的表示方式。

7. 用命令（）可以查看mysql数据库中user表的表结构？

- A desc user;
- B show create table user;
- C show columns for user;
- D describe user;

### 三. 填空题

8.

```
char *p1;int64 *p2;
p1=(char *)0x8000000;
p2=(int64 *)0x8000000;
char *a=p1+2
int64_t *b=p2+2
```

那么a=(),b=()

9. 有一个数组（53,83,18,59,38,35），依次将其存储在hash表中，其中哈希函数为 $h(k)=k\%7$ ,如采用线性探测（每次向后查找1位）的方式解决冲突，则该hash表上查找38,35,53访问hash表的表项次数分别为(),(),()。

10. 32位系统上

```
char c1[]={'a','b','\0','d','e'};
char c2[]="hello";
```

sizeof(c1),strlen(c1),sizeof(c2),strlen(c2)值分别是()()()()。

### 四. 问答题

11.

使用C/C++语言写一个函数，实现字符串的反转，要求不能用任何系统函数，且时间复杂度最小。

函数原型是：char \*reverse\_str(char \*str)

12. 在SQL中，一个表的定义如下：

```
CREATE TABLE t_account(
    account varchar(100),
    account_type TINYTEXT,
```



```
PRIMARY KEY (account),  
};
```

account为账号，account\_type为该账号的类型，写出一个sql，统计账号数累计超过5000个账号类型，并显示对应的账号数，即结果中每行是（账号类型，账号数）

13. http状态码中，()表示访问成功，()表示坏请求，()表示服务不可用

14. 现有/home/script/check.sh脚本，要求每周一到周五14点内每三分钟运行一次，相应的crontab配置是()

15. 请找出下面用于拷贝内存的代码中的逻辑错误，并修正。

```
void memcpy(const char* src,char* dest){  
    int len=strlen(src);  
    dest=(char*)malloc(len);  
    char* d=dest;  
    char* s=src;  
    while(len--!=0){  
        *d=*s;  
        d++;  
        s++;  
    }  
}
```

16.

有pqueue.h如下

```
#ifndef HEADER_PQUEUE_H  
#define HEADER_PQUEUE_H  
typedef struct_pqueue{  
    pitem *items;  
    int count;  
}pqueue_s;  
typedef struct_pqueue *pqueue;  
typedef struct_pitem{  
    unsigned char priority[8];  
    void *data;  
    struct_pitem *next;  
}pitem;  
typedef struct_pitem *piterator;  
pitem *pitem_new(unsigned char *prio64be,void *data);  
void pitem_free(pitem *item);  
  
pqueue pqueue_new(void);  
void pqueue_free(pqueue pq);  
pitem *pqueue_insert(pqueue pq,pitem *item);  
pitem *pqueue_peek(pqueue pq);
```



```
pitem *pqueue_pop(pqueue pq);
pitem *pqueue_find(pqueue pq,unsigned char *prio64be);
pitem *pqueue_iterator(pqueue pq);
pitem *pqueue_next(piterator *iter);
int pqueue_size(pqueue pq);
#endif /*! HEADER_PQUEUE_H */
```

pq\_test.c如下:

```
#include<stdlib.h>
#include<string.h>
#include"pqueue.h"
/*remember to change expected.txt if you change there values*/
unsigned char prio1[8]="supercal";
unsigned char prio2[8]="ifragili";
unsigned char prio3[8]="sticexpi";
static void
pqueue_print(pqueue pq)
{
    pitem *iter,*item;
    iter=pqueue_iterator(pq);
    for(item=pqueue_next(&iter);item!=NULL;
        item=pqueue_next(&iter)){
        printf("item\t%02x%02x%02x%02x%02x%02x%02x%02x\n",
            item->priority[0],item->priority[1],
            item->priority[2],item->priority[3],
            item->priority[4],item->priority[5],
            item->priority[6],item->priority[7],
        )
    }
}
int main(void)
{
    pitem *item;
    pqueue pq;
    pq=pqueue_new();
    item=pitem_new(prio3,NULL);
    pqueue_insert(pq,item);

    item=pitem_new(prio1,NULL);
    pqueue_insert(pq,item);

    item=pitem_new(prio2,NULL);
    pqueue_insert(pq,item);
    item=pqueue_find(pq,prio1);
    fprintf(stderr,"found %p\n",item->priority);
    item=pqueue_find(pq,prio2);
    fprintf(stderr,"found %p\n",item->priority);

    item=pqueue_find(pq,prio3);
    fprintf(stderr,"found %p\n",item->priority);

    pqueue_print(pq);
```



```
    for(item=pqueue_pop(pq);item!=NULL;item=pqueue_pop(pq))  
        pitem_free(item);  
  
    pqueue_free(pq);  
    return 0;  
}
```

pq\_test.sh如下:

```
#!/bin/sh  
set -e  
./pq_test | cmp $srcdir/pq_expected.txt-
```

pq\_expected.txt如下:

```
item 6966726167696c69  
item 7374696365787069  
item 737570657263616c
```

- 1.根据测试代码描述pqueue的工作原理。
- 2.请实现 pitem \*pqueue\_insert(pqueue pq,pitem \*item);



技术QQ群: 379386529



微博: <http://www.weibo.com/nowcoder>



微信

登录牛客网, 参与以上题目讨论, 查看更多笔试面试题