1.[单选题]

同机房不同机器之间使用ping命令得到平均rtt与下面哪个值最接近

50 ms

5 ms

0.5 ms

0.05 ms

2. [单选题]

64位系统下有 `char \*\* p[4];` 请问 `sizeof p` 的值是多少

8

16

32

64

3. 单选题]

下面哪个方法不能直接支持负载均衡

DNS

NIC Bonding broadcast model

LVS

nginx upstream

4. [单选题]

关于下面代码的说法正确的是

extern "C" {

void foo(int) { }

}

"C" 代表c语言

这段代码应于在c++语言代码中

这段代码告诉c++调用者这是一段C代码

用nm查看，foo实际被命名为类似风格 `\_\_Z4fooi`

5. [单选题]

有定义语句`int \*ptr [4];` 下面等价的定义语句是

int \*\* ptr;

int（ \* ptr) [4];

int ptr[4];

int \* (ptr [4]);

6. [单选题]

下面关于http说法正确的是

是有状态的协议

是基于udp的无状态协议

是超文本传输协议

HTTP/1.1 不支持建立持久连接

7. [单选题]

下列哪种数据结构常用来存储数据库索引？

B-Tree

AVL-Tree

HashTable

RB-Tree

8. [单选题]

IPv6地址由多少位二进制数组成？

16

32

64

128

9. [不定项选择题]

以下关于字节序描述正确的是

网络字节序是 Little endian

网络字节序是 Big endian

intel x86系统是 Little endian

htonl是将无符号长整型从本机转换为网络字节序

10. [不定项选择题]

以下关于性能指标描述正确的是

L1 cache reference : 20 ns

L2 cache reference : 50 ns

Main memory reference : 100 ns

Mutex lock/unlock : 250,000 ns

11. [不定项选择题]

关于下面代码，哪些描述是错误的

std::vector<char> v{'1', '2', '3', '4'};

// A

v.push\_back('5');

// B

\*v.end() == '4'

v.capacity() 一定大于 v.size()

如果A位置v.size()与v.capacity()分别是4,4；那么B位置是5,5

如果A位置v.size()与v.capacity()分别是4,4；A位置&v[0]等于B位置的&v[0]

12. [不定项选择题]

关于 volatile 的描述哪些错误的

能保证原子性（Atomicity）

能保证顺序性（Ordering）

能保证 volatile x=1 的原子性

能用来进行多线程同步

13. [不定项选择题]

下面哪些算法可用于“求一个N位整数x的二进制表示中1的个数”

HAKMEM169

popcnt assembly

Shell sort

mt19937

14. [不定项选择题]

在RTT为40ms的两台服务器之间通过tcp传输磁盘上的文本文件，如果传输过程中对数据进行压缩，下面哪些是正确的

cpu开销会增加

文件传送完成的时间开销会减少

单个tcp包大小会有明显变化

带宽开销会减少

15. [不定项选择题]

下面关于socket选项的哪些说法正确

SO\_KEEPALIVE : 发送周期性保活报文以维持连接

SO\_LINGER : 不管是否有数据需要发送，都延迟关闭

TCP\_NODELAY : 打开 Nagle 算法

SO\_REUSEADDR : 强制使用处于 TIME\_WAIT 状态的连接占用的 socket 地址

16. [不定项选择题]

下面关于stl的数据结构的哪些说法正确

std::map 底层是由rb-tree实现

std::set 底层是由hashtable实现

std::unordered\_map 底层是由rb-tree实现

std::multiset 用rb-tree的 insert\_equal 来实现 insert

17. [编程题] 非递归实现的binary\_search

时间限制：1秒

空间限制：32768K

给出一个有n个从小到大排序的且无重复的int序列,在数组中查找target并返回位置,如果不存在返回-1.(本题会人工改卷,请使用非递归实现的binary\_search实现)

输入描述:

输入包含两行:

第一行有两个整数n(1 ≤ n ≤ 100000000),表示数组数字个数n;target,即查找的目标值。 第二行为n个整数,范围均在32位整数,以空格分隔,保证输入数据合法

输出描述:

如果存在,输出target出现的下标,否则输出-1

输入例子1:

7 0 0 1 2 3 4 5 6

输出例子1:

0

18. [问答题]

题目描述

有代码

union {

uint32\_t u32;

uint8\_t arr[4];

} x;

x.u32 = 12345;

printf("%p, 0x%02x\n", &x.arr[0], x.arr[0]);

printf("%p, 0x%02x\n", &x.arr[1], x.arr[1]);

printf("%p, 0x%02x\n", &x.arr[2], x.arr[2]);

printf("%p, 0x%02x\n", &x.arr[3], x.arr[3]);

的输出为：

0x22cac0, 0x39

0x22cac1, 0x30

0x22cac2, 0x00

0x22cac3, 0x00

请写出下面代码的输出结果

printf("htole32(x.u32) = 0x%08x\n", htole32(x.u32));

printf("htobe32(x.u32) = 0x%08x\n", htobe32(x.u32));

19. [问答题]

题目描述

linux 的 list \_x005F的实现有一些宏定义

1

2

3

4

5

6

7

8

9

10

11

12

#define LIST\_HEAD\_INIT(name) { &(name), &(name) }

#define LIST\_HEAD(name) \

struct list\_head name = LIST\_HEAD\_INIT(name)

#define INIT\_LIST\_HEAD(ptr) do { \

(ptr)->next = (ptr); (ptr)->prev = (ptr); \

} while (0)

#define list\_entry(ptr, type, member) \

( (type \*)( (char \*)(ptr)-(unsigned long)(&( (type \*) 0

)->member) ) )

LIST\_HEAD(foo)

宏展开后结果是什么 ？

有下列代码，

#pragma pack(push)

#pragma pack(n)

struct Node {

uint16\_t d0;

uint8\_t d1;

uint16\_t d2;

struct list\_head list;

} ;

#pragma pack(pop)

请写出 `#pragma pack(n)` 中，当 n =1, 2, 8 时 `OFFSETOF(Node, list)` 在64位系统的结果

20. [问答题]

题目描述

liunx 使用 buddy/slab等算法管理内存，请问 - a. 外碎片与内碎片是什么及各自产生的原因 - b. buddy/slab方案对应的碎片是 - c. 简单说明buddy/slab的工作方式

21. [问答题]

题目描述

C++编程过程中你会用哪些编程方法来避免资源泄漏, 说明原因并用具体代码展示

22. [问答题]

题目描述

按顺序写出下面代码的输出 (设 \_\_PRETTY\_FUNCTION\_\_ 在 virtual void fun() 输出为 virtual void Base::fun()，)

class Base {

public:

virtual void fun() { cout << \_\_PRETTY\_FUNCTION\_\_ <<

endl; }

void fun2() { cout << \_\_PRETTY\_FUNCTION\_\_ << endl;

}

virtual ~Base() { cout << \_\_PRETTY\_FUNCTION\_\_ <<

endl; };

};

class Derived : public Base {

public:

void fun() { cout << \_\_PRETTY\_FUNCTION\_\_ << endl;

}

void fun2() { cout << \_\_PRETTY\_FUNCTION\_\_ << endl;

}

~Derived() { cout << \_\_PRETTY\_FUNCTION\_\_ << endl;

};

};

int main() {

Base \*a = new Base;

Base \*b = new Derived;

a->fun();

b->fun();

a->fun2();

b->fun2();

delete a;

delete b;

return 0;

}

23. [问答题]

题目描述

请找出下列代码的memory相关问题

class Base {

public:

Base() { ptr = new char[10240]; }

~Base() {}

virtual void fun() { delete ptr; }

protected:

char \*ptr;

};

class Derived : public Base {

public:

Derived() { ptr = new char[20480]; }

~Derived() {}

virtual void fun() { Base::ptr[10240] = '\0'; }

protected:

char \*ptr;

};

int main() {

Base \*ptr = new Derived;

ptr->fun();

delete ptr;

delete ptr;

return 0;

}

24. [问答题]

题目描述

代码中需要用到正则表达式用于校验XXX.XXX.XXX.XXX格式的IPv4地址，请考虑尽可能多的边界条件，写出正则表达式

参考答案：

DCBBD

CAD

9.BCD

10.CD

11.ABCD

12.ABCD

13.AB

14.ABD

15.AD

16.AD

17. #include<stdio.h>

int a[100000000],n,i,target,l,r;

int main(){

    for(scanf("%d%d",&n,&target),i=0;i<n;i++) scanf("%d",a+i);

    for(l=0,r=n-1;l+1<r;){

        int mid=l+(r-l)/2;

        if(mid<=target) l=mid;

        else r=mid;

    }

    if(a[l]!=target&&a[r]!=target) printf("-1\n");

    else printf("%d\n",a[l]==target?l:r);

}

18.

htole32(x.u32) = 0x00003039

htobe32(x.u32) = 0x39300000

19. struct list\_head foo = { &(foo), &(foo) };

5, 6, 8

20. virtual void Base::fun()

virtual void Derived::fun()

void Base::fun2()

void Base::fun2()

virtual Base::~Base()

virtual Derived::~Derived()

virtual Base::~Base()

21. 1. Invalid write of size 1

2. 10,240 bytes in 1 blocks are definitely lost

3. 20,480 bytes in 1 blocks are definitely lost

4. Invalid free() / delete / delete[] / realloc()

5. delete delete ptr;

22. ^(25[0-5]|2[0-4]\d|[0-1]?\d?\d)(\.(25[0-5]|2[0-4]\d|[0-1]?\d?\d)){3}$