

## 选择题

1.已经获得除CPU以外的所有所需资源的进程处于( )状态

- A 就绪状态
- B 阻塞状态
- C 运行状态
- D 活动状态

A

进程的五状态模型：

运行态：该进程正在执行。

就绪态：进程已经做好了准备，只要有机会就开始执行。

阻塞态（等待态）：进程在某些事情发生前不能执行，等待阻塞进程的事件完成。

新建态：刚刚创建的进程，操作系统还没有把它加入到可执行进程组中，通常是进程控制块已经创建但是还没有加载到内存中的进程。

退出态：操作系统从可执行进程组中释放出的进程，或由于自身或某种原因停止运行。

2.某二叉树的中序遍历序列为32145，后序遍历序列为32145，则前序遍历序列为

- A 54123
- B 32154
- C 32541
- D 54321

A

二叉树的中序遍历序列为 32145，后序遍历序列为32145，可知该树只有左子树结点，没有右子树结点，5 为根结点。

中序遍历序列与后序遍历序列相同，说明该树只有左子树没有右子树，因此该树有 5 层，从顶向下依次为54123。

具体分析过程也可以参考下[北京联发科嵌入式软件工程师笔试题目解析](#)

3.若已知一个栈的入栈顺序是1,2,3...,n,其输出序列为P1,P2,P3,....Pn,若P1是n,则Pi= ( ) ?

- A i
- B n-i+1
- C 不确定

D n-i

B

栈的排列遵循先进后（即后进先出）出的原则

因为P1是n，是出栈的第一个数字，说明在n之前进栈的数字都没有出栈。所以这个顺序是确定的。

还可以知道，最后出栈的一定是数字1，也就是Pn。代入这个式子，是正确的。

4 (多选题).下面协议中属于应用层协议的是 ( )

A ICMP、ARP

B FTP、TELNET

C HTTP、SNMP

D SMTP、POP3

BD

1、物理层：以太网、调制解调器、电力线通信(PLC)、SONET/SDH、G.709、光导纤维、同轴电缆、双绞线等。

2、数据链路层：Wi-Fi(IEEE 802.11)、WiMAX(IEEE 802.16)、ATM、DTM、令牌环、以太网、FDDI、帧中继、GPRS、EVDO、HSPA、HDLC、PPP、L2TP、PPTP、ISDN、STP、CSMA/CD等。

3、网络层协议：IP IPv4、IPv6、ICMP、ICMPv6、IGMP、IS-IS、IPsec、ARP、RARP、RIP等。

4、传输层协议：TCP、UDP、TLS、DCCP、SCTP、RSVP、OSPF等。

5、应用层协议：DHCP、DNS、FTP、Gopher、HTTP、IMAP4、IRC、NNTP、XMPP、POP3、SIP、SMTP、SNMP、SSH、TELNET、RPC、RTCP、RTP、RTSP、SDP、SOAP、GTP、STUN、NTP、SSDP、BGP等。

5.下列程序段的时间复杂度是 ( )

```
int fact(int n){
    if(n<=1){
        return 1;
    }
    return n*fact(n-1);
}
```

A  $O(\log_2 n)$ B  $O(n \log_2 n)$ C  $O(n)$ D  $O(n^*n)$

C

当 $n \leq 1$ 时执行return 1这一个语句，每次返回上一层都执行 $n * \text{fact}(n-1)$ 这一个语句，共执行 $n-1$ 次。因此共执行基本语句 $n$ 次，时间复杂度为 $O(n)$

6.下列排序算法中最好情况和最坏情况的时间复杂度相同的是？（ ）

- A 堆排序
- B 快速排序
- C 冒泡排序
- D 归并排序

A C D

堆排序在最好和最坏情况下的时间复杂度均为 $O(n \log n)$

快速排序最好和最坏情况下的时间复杂度分别为 $O(n \log n)$ 和 $O(n^2)$

冒泡排序排序在最好和最坏情况下的时间复杂度均为 $O(n^2)$

归并排序在最好和最坏情况下的时间复杂度均为 $O(n \log n)$

7.将两个各有 $n$ 个元素的有序表归并成一个有序表，最少的比较次数是？（ ）

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

A

归并排序是将两个或两个以上的有序子表合并成一个新的有序表。在归并排序中，核心步骤是将相邻的两个有序序列归并为一个有序序列。

题目中告诉我们，有两个各有 $n$ 个元素的有序序列，要将这两个序列归并成一个有序序列，其方法是依次从小到大取每个序列中的元素进行比较，将较小的放进一个新的序列中，直到取完一个有序序列中的所有元素。再把另一个序列中剩下的元素放进新序列的后面即可。

最好的情况是一个有序序列中的最小元素大于另一个有序序列中的所有元素，这样只需要比较 $n$ 次。

8.将递归算法转换为非递归算法通常需要使用（ ）

- A 栈
- B 队列
- C 队尾
- D 广义表

D

9.在MySQL中，`productname regexp '[1-3]xiaomi'`的含义是（ ）

- A productname 匹配“xiaomi重复1次或5次”的字符串
- B productname 匹配“xiaomi字符串前一个字符为1或3”的字符串
- C productname 匹配“xiaomi重复1到3次”的字符串
- D productname 匹配“xiaomi字符串前一个字符为1到3”的字符串

D

10.同个进程的不同线程以下不能被共享的是？（ ）

- A 全局变量
- B 堆
- C 文件句柄
- D 栈

B

线程共享的进程环境包括：

进程代码段、进程的公共资源（如全局变量，利用这些共享的数据，线程很容易的实现相互之间的通信）、进程打开的文件描述符、消息队列、信号的处理程序、进程的当前目录、进程用户ID、进程组ID

线程独占资源：

线程ID、寄存器组的值、用户栈、内核栈（在一个进程的线程共享堆区（heap））、错误返回码、线程的信号屏蔽码、线程的优先级

## 专项选择题

1.下列Java函数的执行结果是什么（ ）

```
static boolean foo(char c)
{
    System.out.print(c);
    return true;
}
public static void main(string[] args){
    int i = 0;
    for(foo('B');foo('A')&&(i<2);foo('C'))
    {
        i++;
        foo('D');
    }
}
```

A BADCBDCB

B BACDBACD

C BADCADCA

D 运行时抛出异常

C

1.其实foo('B');就是初始化条件，只会执行一次，所以第一个打印的肯定是B。

2.因为i=0;循环条件是i<2（由此可知，循环i等于2的时候就会停止循环），所以0<2满足条件，接着会输出A。然后执行i++;i就变成1了，在输出D，在最后输出C。一次循环后的结果是：BADC。

3.第二次循环的开始是foo('B');是初始条件，所以不会执行。直接从foo('A')开始，输出A，然后i为1，且小于2，此时循环体内再次执行i++；i的值为2了，再次输出D，最后输出C。第二次循环输出：ADC。

4.然后循环再次执行for(foo('B');foo('A')&&(i<2);foo('C'))，直接输出A。i的值在第二轮循环后的值变成了2，2<2不成立，终止循环，输出A。

故输出结果是：BADCADCA。

2.下列有关软链接表述正确的是？（ ）

A 不可以对不存在的文件创建软链接

B 不能对目录创建软链接

C 和普通文件没有什么不同，inode都指向同一个文件在硬盘中的区块

D 保存了其代表的文件的绝对路径是另一种文件。在硬盘上有独立的区块，访问时替代自身路径

C

A：错。后半句说的是硬链接。硬链接是共同拥有同一个inode,不过每个链接名不同，暂时理解成不同的文件名却指向同一文件。一个文件每加一个硬链接linkcount加1。

B：错。可以对目录创建软链接。如下图所示。

D:错。可以对不存在的文件创建软链接，如下图所示。

3.选项中那一行代码可以替换//add code here 而不产生编译错误（ ）

```
public abstract class MyClass{
    public int testInt = 5;
    //addcode here
    public void method(){
```

```

    }
}

```

- A public abstract void another Method(){}
- B testInt = testInt \* 5
- C public int method();
- D public abstract void another Method(int a)

D

A:该项方法有abstract修饰，所以是抽象方法，由于抽象方法不能有方法体，所以A项错误

B:类体中只能定义变量和方法，不能有其他语句，所以B项错误

C:选项中的方法和类中的方法重复，所以会发生编译异常，所以C项错误

#### 4.有关Java静态初始化块说法不正确的是？（ ）

- A 用户可以控制何时执行静态初始化块
- B 无法直接用静态初始化块
- C 在创建第一个实例前，将自动调用静态初始化块来初始化
- D 静态初始化块没有访问修饰符和参数

A

JAVA的初始化顺序：

父类的静态成员初始化>父类的静态代码块>子类的静态成员初始化>子类的静态代码块>父类的代码块>父类的构造方法>子类的代码块>子类的构造方法

#### 5(多选题)以下分别对变量a给出定义，正确的有（ ）

- A 一个有10个指针的数组，该指针指向同一个整型数:int \*a[10];
- B 一个指向10个整型数组的指针:int (\*a)[10];
- C 一个指向函数的指针，该函数有一个整型数并返回一个整型数:int \*a(int);
- D 一个有10个指针的数组，该指针指向一个函数，该函数有一个整型参数并返回一个整型数：int (\*a[10])(int);

ABD

C改为：int (\*a)(int)

指针数组：首先是一个数组，数组里面的元素都是指针；（存储指针的数组）

数组指针：首先是一个指针，指针指向一个一维数组；（指向数组的指针）

函数指针：一定要理解，回调中经常使用函数指针；

指针函数：就是一个普通函数，只是返回值是指针形式；

6(多选题)下列叙述正确的是 ( )

- A 指针可以为空，引用不能为空。
- B 不存在指向空值的引用，但是存在指向空值的指针
- C 引用必须被初始化，但是指针不必
- D 指针初始化后不能被改变，引用可以改变所指对象

ABC

D：引用初始化以后不能被改变,指针可以改变所指的对象

7.下列关于C++容器描述错误的是？ ( )

- A list类型支持双向顺序访问，在list中任何位置插入删除都很快
- B deque类型支持快速顺序访问，在头尾插入 / 删除速度很快
- C C++标准库map的底层实现为红黑树
- D vector类型在每次调用 pushback时都在栈上开辟新内存

B

deque：双端队列。支持快速随机访问。在头尾位置插入/删除速度很快。

8 (多选题) C++中，下列数据类型的转换，哪个可能会发生信息丢失？

- A int -> double
- B int -> long
- C long -> float
- D int -> char

CD

精度丢失只会发生在从大范围到小范围的转换

32位编译器：

char short int long float double 指针

1 2 4 4 4 8 4

64位编译器：

char short int long float double 指针

1 2 4 8 4 8 8

9.在Java中，要使某个类能被同一个包中的其他类访问，但不能被这个包以外的类访问，可以（ ）

- A 使用 private关键字
- B 让该类不使用任何关键字
- C 使用public关键字
- D 使用protected关键字

B

default和protected的区别是：

前者只要是外部包，就不允许访问。

后者只要是子类就允许访问，即使子类位于外部包。

总结：default拒绝一切包外访问；protected接受包外的子类访问

10（多选题）list和vector的区别有哪些（ ）

- A vector拥有一段连续的内存空间，因此支持随机存取，如果需要高效的随即存取，而不在乎插入和删除的效率，使用 vector.
- B list拥有一段不连续的内存空间，因此支持随机存取，如果需要大量的插入和删除，而不关心随即存取，则应使用list
- C 已知需要存储的元素时，使用list较好
- D 如果需要任意位置插入元素，使用 vector较好

AB

C:已知需要存储的元素时,vector要好

D:如果需要任意位置插入元素，list要好

## 编程题1（字符串筛选）

给定一个字符串，需要去除所有之前曾经出现过的字符，只保留第一次出现的字符。

样例输入:hello,welcome to xiaomi

样例输出:helo,wcmtxia

### 思路

1.  以及“输出的字符串数组” new[ ]；
2. 取old数组中的第一个字符去和new数组中的每一个字符串相比较是否相同，若出现相同，则取old数组的下一个字符再次与new中每一个字符相比较，若都不相同则存入new的数组中；
3. 最后输出数组new；

## 代码

```

#include <stdio.h>
void killsame(char *o, char *n)
{
    int i=0, j, k=0;
    int label;

    while(o[i] != '\0')
    {
        label = 1;
        for(j=0; j<i; j++)
        {
            if (o[i] == n[j])
                label = 0; //一旦相同标志位置0
        }
        if(label) // 不相等
            n[k++]=o[i];
        i++;
    }
    n[k]='\0'; //结尾给\0
    puts(n); //输出
}

int main(void)
{
    printf("Please input a string you want:\n");
    char old[126];
    char new[126];
    scanf("%s",old);
    killsame(old, new);//去重
    return 0;
}

```

## 编程题2 ( 字符串有效判断 )

给定一个只包括"(",),"{","}","[","]"的字符串，判断字符串是否有效

有效字符串需满足：

- 1.左括号必须使用相同类型的右括号闭合
- 2.左括号必须以正确的顺序闭合

注意空字符串可被认为是有效字符串

输入描述：待判断的字符串，多个字符串需换行输入

输出描述：每个字符串的判断结果，多个结果需换行输出

样例输入：

() [] {}

(D)

{[]}

样例输出：

true

false

true

### 思路

栈先入后出特点恰好与本题括号排序特点一致，所以用栈来实现。

当左括号出现的时候入栈，当右括号出现的出栈，如果匹配就继续，不匹配就错误。

当字符串遍历完成之后，栈内仍有字符串就错误。

用一个数组进行和一个记录栈顶值的int进行了栈的模拟，代码很简单，很好理解。

### 代码

```
#include <stdio.h>
bool isValid(char * s){

    int len = strlen(s);
    char stack[3500];
    int top = -1;
    int i = 0;
    for( i=0;i<len;i++)
    {
        if((s[i] == '(')||(s[i] == '{')||(s[i] == '['))
            stack[++top] = s[i];
        else
        {
            if(top<0)//出现了右括号，但数组为空，即没有左括号与之匹配
                return false;
            if((s[i] == ')'))
            {
                if(stack[top]!='(') return false;
                else top--;
            }

            if((s[i] == '}'))
            {
                if(stack[top]!='{') return false;
                else top--;
            }

            if((s[i] == ']'))
```

```
        {
            if(stack[top]!='(') return false;
            else top--;
        }

    }
}
if(top>=0) //数组内仍有左括号·没有右括号与之匹配
    return false;\
//这里一定要有返回值
    return true;
}
int main(void)
{
    printf("Please enter a bunch of brackets:\n");
    char brackets[100];
    scanf("%s",brackets);
    printf(isValid(brackets));
    return 0;
}
```