

2021 912真题

数据结构

全是大题，没有选填

1. 从渐进意义上讲 $f(n) = 2^{\log^2 n}$ 与 n^{2020} 哪个增速更快，为什么？
2. $f(n) = 3f\left(\frac{n}{9}\right) + O(1)$ 则从大O记号的角度来讲， $f(n)$ 属于什么复杂度？
3. 中缀表达式与RPN的转换
4. next表与改进的next表填写
5. AVL删除节点发生失衡时（1）我们用g, p, v来表示失衡节点及其相关节点，怎么找到的g, p, v（2）通过connect34修复失衡时，参数有几种可能（3）列举这些可能
6. 对于有向无环图，是否DFS的逆序（总能）（有没有总能我忘了）构成一个拓扑排序，如果是请证明，如果不是给出反例
7. 算法题

```
BstNode{
    BstNode *lc, *rc;
    int size, key
}
count(BstNode * T, int lo, int hi){
    //通过伪代码给出一个算法，
    //计算lo与hi之间所有节点的数目，
    //要求时间复杂度O(h)，空间复杂度O(1)
}
```

(1)写出算法主体（2）给出算法原理并说明正确性（3）证明你的算法符合时间空间复杂度要求

组成原理

判断：

1. 字符的编码中是否蕴含了字符的表示（我没太读明白，大概是这个意思）
2. 忘了

3. 忘了
4. Cache 总容量一定的话，全相连组织方式的命中率不低于组相连方式的命中率。
5. DMA不能独占总线周期

选择：

1. IEEE单精度浮点数表示的范围
 - A.正数多
 - B.负数多
 - C.一样多
 - D.忘了
2. 执行同一条指令，下列哪个用时最短
 - A.主频 2.5GHZ CPI 1
 - B.主频2.8GHZ CPI 1.1
 - C.主频3GHZ CPI 1.5
 - D.主频2GHZ CPI 1.1
3. 关于TLB的说法正确的是
 - A.忘了
 - B.TLB的缺失能在指令cache中找到
 - C.TLB会引发程序错误
 - D.TLB中保存了虚拟地址到物理地址的映射
4. 不可用于解决数据冲突的是
 - A.旁路
 - B.阻塞
 - C.动态预测
 - D.动态调度/编译器调度

填空：

1. short类型变量高字节为01H 低字节为FFH 在大端表示的机器和小端表示的机器中，变量的值分别是多少
2. 32位地址空间，1024个cache块，二路组相连，块内8字节，需要多少位的cache标识和索引
3. 冯诺依曼计算机最大的特点是？

大题：

```
LW R4 0(R5)
ADD R6 R4 R3
SUB R8 R7 R5
```

- (1) 有哪些指令和寄存器存在冲突?
- (2) 不加数据旁路的情况下至少需要阻塞几个周期?
- (3) 加了数据旁路至少需要阻塞几个周期, 并说明每个冲突中旁路数据来源于那个阶段寄存器

操作系统

判断:

1. clock算法可以通过修改和读取页表中的访问位实现
2. 在用了银行家算法时, 系统在不安全状态下会出现死锁
3. 每个进程有且仅有一个PCB
4. 可以通过改变某个寄存器位的方式停止时钟中断
5. 如果系统中有M个进程, 那处于就绪队列中的进程数最多为M-1
6. 如果一个父进程在有許多子进程的状态下推出了, 它的子进程将变为僵尸进程。

填空: 十分类似这个题, 不过具体的操作有差别, 而且改成了填写对应操作后的硬盘状态

某操作系统中建立了一个基于索引节点(index node)结构的文件系统very simple file system, 简称vsfs。

vsfs的用户操作包括(以函数形式表示):

mkdir("str") - 创建一个新目录, 目录名称为"str"
creat("str") - 创建一个空文件, 空文件名称为"str"
fd=open("str"), write(fd), close(fd) - 打开文件"str", 会返回一个整型数fd, 然后对文件写一个buffer, 注意常规文件的最大size是一个data block, 所以第二次写(写文件的语义是在上次写的位置后再写一个data block)会报错(文件大小满了)。或者如果data block也满了, 也会报错。
link("a1", "b1") - 对文件"a1"创建一个硬链接(hard link)文件"b1"
unlink("b1") - 对文件"b1"取消一个硬链接, 如果文件的链接数为0, 则删除此文件
vsfs在硬盘上的布局:

superblock : 可用inode数量, 可用data block数量
inode bitmap : inode的分配图(基于bitmap)
inodes : inode的存储区域
data bitmap : data block的分配图(基于bitmap)
data : data block的存储区域
vsfs的关键数据结构:

inode数据结构:
inode : 包含3个fields, 用 list 表示
file type: f -> 常规文件: regular file, d -> 目录文件: directory
data block addr of file content: -1 -> file is empty
reference count: file/directory的引用计数, 注意directory的引用计数是指在此目录中的inode的个数
注意: 比如, 刚创建的一个空文件inode: [f a:-1 r:1], 一个有1个硬链接的文件inode: [f a:10 r:2]

数据块内容结构：

一般文件的内容的表示：只是包含单个字符的list,即占一个data block,比如['a'], ['b']

目录内容的表示：多个两元组(name, inode_number)形成的list,比如，根目录[(.,0) (.,0)], 或者包含了一个'f'文件的根目录[(.,0) (.,0) (f,1)] 。

注意：

一个目录的目录项的个数是有限的。 block.maxUsed = 32

data block的个数是有限的,为 fs.numData

inode的个数是有限的,为 fs.numInodes

完整文件系统的例子：

```
fs.ibitmap: inode bitmap 11110000
```

```
fs.inodes: [d a:0 r:5] [f a:1 r:1] [f a:-1 r:1] [d a:2 r:2] [] ...
```

```
fs.dbitmap: data bitmap 11100000
```

```
fs.data: [(.,0) (.,0) (y,1) (z,2) (x,3)] [u] [(.,3) (.,0)] [] ...
```

表明：此文件系统有8个inode空间，8个data blocks。其中,根目录包含5个目录项，“.”，“..”，“y”，“z”，“x”。而“y”是常规文件,并有文件内容,包含一个data block,文件内容为“u”。“z”是一个空的常规文件。“x”是一个目录文件，是空目录。

如果vsfs初始状态为：

```
inode bitmap 10000000
```

```
inodes [d a:0 r:2] [] [] [] [] [] [] []
```

```
data bitmap 10000000
```

```
data [(.,0) (.,0)] [] [] [] [] [] [] []
```

请问接下来的连续6个状态变化的对应用户操作是什么?据此回答以下6个问题

(1)

```
inode bitmap 11000000
```

```
inodes [d a:0 r:3] [d a:1 r:2] [] [] [] [] [] []
```

```
data bitmap 11000000
```

```
data [(.,0) (.,0) (c,1)] [(.,1) (.,0)] [] [] [] [] [] []
```

对应用户操作是 ()

(2)

```
inode bitmap 11100000
```

```
inodes [d a:0 r:3] [d a:1 r:3] [f a:-1 r:1] [] [] [] [] []
```

```
data bitmap 11000000
```

```
data [(.,0) (.,0) (c,1)] [(.,1) (.,0) (h,2)] [] [] [] [] [] []
```

对应用户操作是 ()

```
mkdir("/c/h");
```

```
unlink("/c/h");
```

```
creat("/c/h");
```

```
fd=open("/c/h"), write(fd), close(fd)
```

(3)

```
inode bitmap 11100000
```

```
inodes [d a:0 r:3] [d a:1 r:4] [f a:-1 r:2] [] [] [] [] []
data bitmap 11000000
data [(.,0) (.,0) (c,1)] [(.,1) (.,0) (h,2) (p,2)] [] [] [] [] []
对应用户操作是 ( )
```

```
mkdir("/c/p");
creat("/c/p");
link("/c/h", "/c/p");
unlink("/c/h");
```

```
(4)
inode bitmap 11100000
inodes [d a:0 r:3] [d a:1 r:3] [f a:-1 r:1] [] [] [] [] []
data bitmap 11000000
data [(.,0) (.,0) (c,1)] [(.,1) (.,0) (p,2)] [] [] [] [] []
对应用户操作是 ( )
```

```
mkdir("/c/h");
unlink("/c/p");
link("/c/h", "/c/p");
unlink("/c/h");
```

```
(5)
inode bitmap 11000000
inodes [d a:0 r:3] [d a:1 r:2] [] [] [] [] [] []
data bitmap 11000000
data [(.,0) (.,0) (c,1)] [(.,1) (.,0)] [] [] [] [] []
对应用户操作是 ( )
```

```
mkdir("/c");
unlink("/c");
creat("/c/p");
unlink("/c/p");
```

```
(6)
inode bitmap 11100000
inodes [d a:0 r:3] [d a:1 r:3] [f a:-1 r:1] [] [] [] [] []
data bitmap 11000000
data [(.,0) (.,0) (c,1)] [(.,1) (.,0) (f,2)] [] [] [] [] []
对应用户操作是 ( )
```

```
mkdir("/c/f");
creat("/c/f");
link("/c", "/c/f");
fd=open("/c/f", write(fd), close(fd))
```

简答:

操作系统中通信可以使用消息队列、信号、信号量、管道，说明下列操作中最适合使用哪种通信机制，并解释为什么

- (1) 输入指令关闭一个进程(是不是关闭进程不太确定)
- (2) linux指令 "cat file | grep exam"
- (3) 第一个进程对一段数据进行写入，另一个进程基于写入的数据对其进行排序

计算机网络

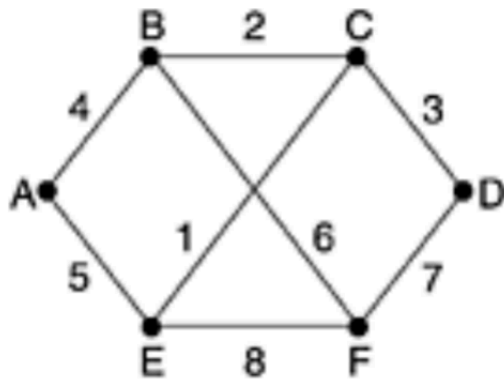
选择:

- 1. 传输层服务访问点是
 - A.IP地址，端口号
 - B.忘了
 - C.MAC地址，IP地址，端口号
 - D.MAC地址，IP地址
- 2. CSMA/CD的链路，数据传输率是 1Gbps ，数据传播速率20 0000km/s 问最小帧长增加50个字节后，最远两节点间距离应该
 - A.增加40m
 - B.减少40m
 - C.增加80m
 - D.减少80m
- 3. 循环冗余码的多项式为 $x^2 + x + 1$ 时下列哪个是对的
 - A.具体数记不清了
 - B.
 - C.
 - D.
- 4. 下列正确的是 I.同一个域名可以拥有多个IP地址 II.同一个IP地址可以拥有多个域名
 - A.I
 - B.II
 - C.都对
 - D.都不对

大题:

1.

- 5.9 考虑图5.13(a)中的子网。该子网使用了距离矢量路由算法，下面矢量刚刚到达路由器C：来自B的矢量为 (5,0,8,12,6,2)；来自D的矢量为 (16,12,6,0,9,10)；来自E的矢量 (7,6,3,9,0,4)。经测量到B、D和E的延迟分别为6,3和5.请问C的新路由表将会怎样？请给出将使用的输出线路以及期望的延迟



(11,6,0,3,5,8)

(B,B,-,D,E,B)

注意路由不对称性，到B、D、E使用自身测算的延迟

和这个题图一样，不过数不一样，求C的路由表

1. 如果c使用链路装调路由算法，则C的发送的链路状态信息是？
2. 依据上图，每个路由器连接一个子网，每个子网内依次有 90, 78, 4, 15, 3, 40, 2（数是编的不过应该差不多）给了一个子网为55.16.22.0/24(也是编的，知道/24就行) 求一个合理的子网划分，写出每个子网的范围 and 对应子网掩码
3. 最后一道大题似乎是和滑动窗口协议相关的（没做完没来得及读题QAQ…写数据结构去了）