真题勘误：

一，2003年数据结构第三题：

3．第一步求出散列值

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 40 | 46 | 34 | 63 | 45 | 11 | 57 | 24 | 17 | 33 | 47 | 79 |
| 6 | 12 | 0 | 12 | 11 | 11 | 6 | 7 | 0 | 16 | 13 | 11 |

溢出处理：Hi=（Hi+i）% m，其中Hi表示要插入的元素，i为0,1,2..,m为17。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 34 | 17 | 79 |  |  |  | 40 | 57 | 24 |  |  | 45 | 46 | 63 | 11 | 47 | 33 |

二，2008操作系统第七题

7．（1）1（直接访问）（2）25（指针顺序访问）（3）2（一次索引块，一次读第25块）

三，2017数据结构第一题

C改为D

四，2018年数据结构第一题

5改为s

五，2013年操作系统期末A第八大题

答： 11：10 12：00

11：40 12：20

11：00 11：10

11：10 11：40

六，2017年软工第八题答案代码改成

Public class Emloyee {

Private Double income;

Public Double getIncome(){

Return income;

}

}

Public class Tax {

Private Double taxrate;

Private Employee employee;

Public Double getTaxrate(){

Return taxrate;

}

Public Double getTax() {

employee = new Employee();

Return employee.getIncome()\*taxrate;

}

}

七，2014年操作系统46题答案改为

J1装入主存时间:\_11:10\_\_\_\_，结束时间:\_\_\_12:00\_\_；

八，2003年数据结构第五题代码改为

Type def struct BiNode{

Int Element;

Struct BiNode Leftchild,Rightchild;

}BiNode \*BiNode;

Int getDepth(BiTree \*T){

Int LD,RD;

If(T == Null) return 0;

Else if((!T->Leftchild)&&(!T->Rightchild))

Return 1;

Else{

LD = getDepth(T->Leftchild);

RD = getDepth(T->Reftchild);

Return ((LD>RD)?LD:RD);

}

}

九，2007年操作系统第五题改为

1）FIFO,缺页次数6，缺页中断率为6/13

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 3 | 2 | 1 | 4 | 4 | 5 | 5 | 3 | 4 | 3 | 2 | 1 | 5 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 5 |
| 2 | 2 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 5 | 5 | 1 | 1 |
| 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 2 | 2 | 2 |
| 是否缺页 |  |  |  | ✔ |  | ✔ |  | ✔ |  |  | ✔ | ✔ | ✔ |

2）LRU，缺页次数6，缺页中断率为6/13

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 3 | 2 | 1 | 4 | 4 | 5 | 5 | 3 | 4 | 3 | 2 | 1 | 5 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 5 |
| 2 | 2 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 5 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 1 | 1 |
| 是否缺页 |  |  |  | ✔ |  | ✔ |  | ✔ |  |  | ✔ | ✔ | ✔ |

十，2016年数据结构第一题

（1）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K |  | B | M | D | CI | X |  |  | I | TM |
| 1 |  | 1 | 2 | 1 | 3 | 5 |  |  | 1 | 2 |

1. 总的比较次数为16

十一，2013年选择题第一题选D

十二，2015年选择10选C,8改为D

十三，2014年选择35改为A

十四，2016年软工第四题（4）改为约束（3）（5）改为系统级需求，（1）改为业务需求

十五，2016年操作系统第二题次数改为250，第四题控制块，程序块，数据块，核心块

十六，2016年操作系统第九题（1）改为

1. q=1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RR |  | A | B | C | B | D | B | D | B | D | B | D | B | D | B | D | B | D | D |

十七，2013备用卷选择题33改为B，35改为C，38改为C

十八，2017年操作系统第五题题目有误

5、考虑下面的进程集合：

|  |  |  |
| --- | --- | --- |
| 进程 | 到达时间 | 处理时间 |
| A | 0 | 2 |
| B | 1 | 6 |
| C | 2 | 2 |
| D | 3 | 6 |

如果使用先来先去服务FCFS调度算法，得到的每个单位时间内的进程执行序列表示为

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 算法 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| FCFS | A | A | B | B | B | B | B | B | C | C | D | D | D | D | D | D |

参照FCFS调度算法给出的执行序列的写法，试写出如下三个调度算法那的执行序列，即在表格中填入每个单位时间内执行的进程代号。

（1）时间片轮转调度算法（RR，q=1，即时间片长度为1）；（2分）

（2）时间片轮转调度算法（RR，q=4，即时间片长度为4）；（2分）

（3）多级反馈队列算法（Fback，q=1，即各个多级反馈队列的时间片长度均为1）。（2分）

5. q=1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RR |  | A | B | A | C | B | D | C | B | D | B | D | B | D | B | D | D |

q=4

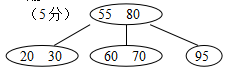
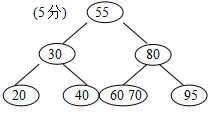
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RR |  | A | A | B | B | B | B | C | C | D | D | D | D | B | B | D | D |

Feedback

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RR |  | A | B | C | D | A | B | C | D | B | D | B | D | B | D | B | D |

十九，2013年计网最后一题改为A：26，B：28，C：26，D：28，答案均不唯一

二十，2005年数据结构第二题的第三小题改为：



二十一，2013年第四十三题改为

有理数类：

public class Rational {

    public int i;

    public int j;

    public Rational(){

    i=1;

    j=1;

    };

    public Rational(int I,int J)

    {

        i=I;

        j=J;

    }

    public void setRational(int I,int J)

    {

                i=I;

                j=J;

    }

    public double getI(){return i;}

    public double getJ(){return j;}

    public Rational sum(Rational b){

    Rational su=new Rational();

    su.setRational((int)(i\*b.getJ()+j\*b.getI()),(int)(b.getJ()\*j));

    return su;

    }

    public Rational subtraction(Rational b){

    Rational sub=new Rational();

    sub.setRational((int)(i\*b.getJ()-j\*b.getI()),(int)(j\*b.getJ()));

    return sub;

    }

    public Rational multiplication(Rational b){

    Rational mul=new Rational();

    mul.setRational((int)(i\*b.getI()),(int)(j\*b.getJ()));

    return mul;

    }

    public Rational division(Rational b){

        Rational div=new Rational();

        div.setRational((int)(i\*b.getJ()),(int)(j\*getI()));

        return div;

    }

    public String toString()

    {

       return i+"/"+j;

    }

}

有理数类的测试：

public class RationalTest {

    public static void main(String[]args)

    {

        Scanner in=new Scanner(System.in);

        System.out.println("please enter two  number");

        Rational a=new Rational(in.nextInt(),in.nextInt());

        System.out.println("please enter two number");

        Rational b=new Rational(in.nextInt(),in.nextInt());

        System.out.println("Rational a:"+a);

        System.out.println("Rational b:"+b);

        System.out.println("a+b:"+ a.sum(b).toString());

        System.out.println("a-b:"+a.subtraction(b).toString());

        System.out.println("a\*b:"+a.multiplication(b).toString());

        System.out.println("a/b:"+a.division(b).toString());

    }

}

2015年操作系统大题第一题改为

1.直接块容量=12×1024B/1024=12KB

一次间接容量=256\*1024B/1024=256KB

二次间接容量=256\*256\*1024B/1024=65536KB

三次间接容量=256\*256\*256\*1024B/1024=1677216KB

28MB=28672KB

28672-12-256=28404KB

28404\*1024B/1024=28404块

所以占二级28404块，一级256块，直接12块

2005年真题数据结构部分解答2扩充4叉树，需要将图中每行按照大小顺序排

2006年数据结构部分解答（1）a下面字母依次添加1 1 2 1 3 5 1 2

（3）原图中重复的1改为3，答案中53改为52

2008年解答（3）第一个图数字改为70

2018年数据结构部分大题最后一题与2008操作相同

2017年软工第九题去重复的题严谨一点暂且可以改为

![C:\Users\yanghaidong\Documents\Tencent Files\1539007875\Image\Group\PWG_3$C]W2OEW}QG{]M](WO.png](data:image/png;base64,)

2017年软工第九题

九、面向对象

Public interface process() {}

Public class addCourse() implements process();

Public class removeCourse() implements process();

2016年软工第六题去掉内聚太高应为内聚不够

2015年操作系统第二题opt算法最后两次替换改为712，712

2014年软工45题

用工厂如下:

public class Department {  
 private List<Employee> employeeList;  
 private simpleFactory factory;  
 public Department(simpleFactory factory){  
 this.factory = factory;  
 }  
 public Employee addEmployee(int employeeType,String name){  
 Employee emp = factory.createEmployee(employeeType,name);  
 emp.setDepartment(this);  
 employeeList.add(emp);  
 update();  
 return emp;  
 }  
}  
public class simpleFactory{  
  
 public Employee createEmployee(int employeeType,String name){  
 Employee emp;  
 switch (employeeType){  
  
 case EMPLOYEE.SALARIED: emp = new SalariedEmployee(name);break;  
 case EMPLOYEE-HOUR: emp= new HourlyEmployee(name); break;  
 case EMPLOYEE.COMMISSION:emp=newComrmissionedEmployee(name); break;  
 default: emp= new Employee(name);  
  
 }  
 return emp;  
 }  
}

2015年软工第八大题方法名统一改为compare（……）

2013操作系统AB卷中管程的相关题目去ppt中找

大题四改为

Clock算法

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 7 | 0 | 1 | 2 | 0 | 1 | 2 | 3 | 2 | 7 |
| 页框0 | 0\* | 0\* | ->0\* | ->0\* | ->0\* | 2\* | 2\* | ->2\* | ->2\* | 3\* | 3\* | 3\* |
| 页框1 | -> | 1\* | 1\* | 1\* | 1\* | ->1 | 0\* | 0\* | 0\* | ->0 | 2\* | 2\* |
| 页框2 |  | -> | 7\* | 7\* | 7\* | 7 | ->7 | 1\* | 1\* | 1 | ->1 | ->7\* |
| 缺页标记 | F | F | F |  |  | F | F | F |  | F | F | F |

缺页次数为\_\_9\_\_次

操作系统样卷中选择题17改为B

大题四改为

Clock算法：缺页次数为 9 次。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 4 | 3 | 2 | 1 | 4 | 3 | 5 | 4 | 3 | 2 | 1 | 5 |
| 页框0 | 4\* | 4\* | →4\* | 1\* | 1\* | →1\* | 5\* | 5\* | 5\* | 5 | →5 | →5\* |
| 页框1 | → | 3\* | 3\* | →3 | 4\* | 4\* | →4 | →4\* | →4\* | 2\* | 2\* | 2\* |
| 页框2 |  | → | 2\* | 2 | →2 | 3\* | 3 | 3 | 3\* | →3 | 1\* | 1\* |
| 缺页标记 | F | F | F | F | F | F | F |  |  | F | F |  |

2013计网期末单选题14改为D，多选题1改为ACD，3改为CE，4改为ABD，5改为AD，6改为BC，8改为AEF，（AB卷对应题）

B卷15改为C

A卷第六题改为

6. A.192.168.20.0/26 A=60+1 26-2=62 192.168.20.0 0 xx xxxx/26

B.192.168.20.128/28 B=6+1 23-2=6 192.168.20.0 1 0 0 1xxx/28

C.192.168.20.160/28 C=12+1 24-2=14 192.168.20.0 1 0 1 xxxx/28

D.192.168.20.64/26 D=31+1 26-2=62 192.168.20.0 1 1 x xxxx/26

B卷对应题作同样修改，并且20改为0

软工B卷第七大题无限不循环数才为无理数,第二题改为：

图也相应的自己作修改

public class Borrower{

Puvlic void borrowBook();

public void notifyReturnBook();

}

class BachlorBo extends Borrower{

int MAX\_FOR BACHLOR=5;

private void borrow BookforBachlor(){

borrowBook();

NotifyReturnBook();

}

}

class MasterBo extends Borrower{

int MAX\_FOR BACHLOR=10;

private void borrow BookforMaster(){

borrowBook();

NotifyReturnBook();

}

}

class TeacherBo extends Borrower{

int MAX\_FOR BACHLOR=20;

private void borrow BookforTeacher(){

borrowBook();

NotifyReturnBook();

}

2013年操作系统A卷258KB改为256KB,B卷一样。