IT314- SoftwareEngineering LAB-7

Name: Sujal Manavadariya

StudentID:202201084

LabGroup:1

|| . Code Debugging and Program Inspection of theJAVA files.

1 .Armstrong

A . ProgramInspection

1. Thereisoneerrorintheprogram,relatedtothecomputationofthe remainder,andithasbeenidentifiedandcorrected.

2. ThemosteffectivecategoryofprograminspectionforthiscodeisCategory C: Computation Errors, as the error pertains to the computation of the remainder, a type of computation error.

3. Program inspection does not identify debugging-related errors. It does not detect issues such as breakpoints or runtime errors like logic errors.

4. The program inspection technique is valuable for identifying and rectifying issues related to code structureandcomputationerrors.

B.Debugging

1. Thereisoneerrorintheprogramrelatedtothecomputationofthe remainder,aspreviously identified.
2. To fix this error, one should set a breakpoint at the point where the remainder is computed to ensure it's calculated correctly. Step through the code to observe the values of variables and expressions during execution.
3. Thecorrectedexecutablecodeisasfollows:

```
//ArmstrongNumber class
Armstrong {
publicstaticvoidmain(Stringargs[]){
        intnum=Integer.parseInt(args[0]);
        intn=num;//usedtocheckatthelasttime int check =
        0, remainder;
        while(num>0){
                remainder=num%10;
                check=check+(int)Math.pow(remainder,3);
                num=num/10;
        }

        if(check==n)
                        System.out.println(n+"isanArmstrongNumber");
        else
                System.out.println(n+"isnotanArmstrongNumber");
        }
}
```

2 GCDandLCM

A. ProgramInspection

1. Therearetwoerrorsintheprogram:

2. Error1:Inthegcdfunction,thewhileloopconditionshouldbewhile(a%b !=0)insteadofwhile(a%b==0)tocalculatetheGCDcorrectly.

3. Error2:Inthelcmfunction,thereisalogicerror.Thelogicusedtocalculate LCM is incorrect and will result in an infinite loop.

4. For this code, the most effective category of program inspection is CategoryC:ComputationErrors,asitcontainscomputationerrorsinboththe gcd and lcm functions.

5. Programinspectionisnotabletoidentifyruntimeissuesorlogicalerrors.It can't identify errors like infinite loops.

6. The program inspection technique is worth applying to identify and fix computation-relatedissues.

B. Debugging

1. Therearetwoerrorsintheprogramasmentionedabove.
2. Tofixtheseerrors:

3. ForError1inthegcdfunction,youneedonebreakpointatthebeginningof thewhilelooptoverifythecorrectexecutionoftheloop.

4. ForError2inthelcmfunction,youwouldneedtoreviewthelogicfor calculatingLCM,asit'salogicalerror.

5. Thecorrectedexecutablecodeisasfollows: import
java.util.Scanner;
public class GCD_LCM
{ staticintgcd(intx,inty){
inta,b;
a=(x>y)?x:y;//aisgreaternumber
b=(x<y)?x:y;//bissmallernumber

while(b!=0){//Fixedthewhileloopcondition int temp = b;
b=a%b;
a=temp;
}
returna;
}
staticintlcm(intx,inty){
return(x*y)/gcd(x,y);//CalculateLCMusingGCD
}
publicstaticvoidmain(Stringargs[]){
Scannerinput=newScanner(System.in);
System.out.println("Enterthetwonumbers:");

int x = input.nextInt();
inty=input.nextInt();
System.out.println("TheGCDoftwonumbersis:"+gcd(x,y));
System.out.println("TheLCMoftwonumbersis:"+lcm(x,y)); input.close();
}
}

3 Knapsack

A. ProgramInspection

1. Thereisoneerrorintheprogram.Itisinthefollowingline:intoption1= opt[n++][w];Thevariablenisincremented,whichisnotintended.Itshouldbe: intoption1=opt[n][w];
2. The category of program inspection that would be most effective for this codeisCategoryC:ComputationErrors,astheidentifiederrorisrelatedto computation within loops.

3. Programinspectionisnotabletoidentifyruntimeerrorsorlogicalerrors thatmightariseduringprogramexecution.

4. The program inspection technique is worth applying to identify and fix computation-relatedissues.

B. Debugging

1. Thereisoneerrorintheprogram,asidentifiedabove.
2. Tofixthiserror,youwouldneedonebreakpointattheline:intoption1= opt[n][w]; to ensure n and w are correctly used without unintended increments.

3. Thecorrectedexecutablecodeisasfollows:

```
publicclassKnapsack{
publicstaticvoidmain(String[]args){
        intN=Integer.parseInt(args[0]);//numberofitems
        intW=Integer.parseInt(args[1]);//maximumweightofknapsack

        int[]profit=newint[N+1];
        int[]weight=newint[N+1];
```

```
//Generaterandominstance,items1..N for (int n
        = 1; n <= N; n++) {
                profit[n] = (int) (Math.random() * 1000);
                weight[n]=(int)(Math.random()*W);
        }
        int[][]opt=newint[N+1][W+1];
        boolean[][]sol=newboolean[N+1][W+1];

        for(intn=1;n<=N;n++){
                for(intw=1;w<=W;w++){
                        intoption1=opt[n-1][w];//Fixedtheincrementhere int option2 =
                        Integer.MIN_VALUE;

                        if(weight[n]<=w)
                                option2=profit[n]+opt[n-1][w-weight[n]];

                        opt[n][w]=Math.max(option1,option2);
                        sol[n][w]=(option2>option1);
                }
        }
        System.out.println("Item"+"\t"+"Profit"+"\t"+"Weight"+"\t"+"Take");

        for(intn=1;n<=N;n++){
                System.out.println(n + "\t" + profit[n] + "\t" + weight[n] + "\t" + take[n]);
                }
        }
}
```

4 MagicNumber

A. ProgramInspection

1. Therearetwoerrorsintheprogram:
2. Error1:Intheinnerwhileloop,theconditionshouldbewhile(sum>0) insteadofwhile(sum==0).
3. Error2:Insidetheinnerwhileloop,therearemissingsemicolonsinthe lines:

s=s*(sum/10);sum=sum%10;

Theyshouldbecorrectedas:

s=s*(sum/10);sum=sum%10;

4. The category of program inspection that would be most effective for this codeisCategoryC:ComputationErrors,asitcontainscomputationerrorsin the while loop.

5. Programinspectionisnotabletoidentifyruntimeissuesorlogicalerrors thatmightariseduringprogramexecution.

6. The program inspection technique is worth applying to identify and fix computation-relatedissues.

B. Debugging

1. Therearetwoerrorsintheprogram,asidentifiedabove.

2. Tofixtheseerrors,youwouldneedonebreakpointatthebeginningofthe inner while loop to verify the execution of the loop. You can also use breakpoints to check the values of num and s during execution.

3. Thecorrectedexecutablecodeisasfollows:

importjava.util.*;

```
public class MagicNumberCheck
{ publicstaticvoidmain(Stringargs[]){
        Scannerob=newScanner(System.in);
        System.out.println("Enterthenumbertobechecked.");

        intn=ob.nextInt();
        intsum=0,num=n;

        while(num>9){
                sum=num; int s
                = 0;

                while(sum>0){//Fixedtheconditionhere
                        s=s*(sum/10);
                        sum=sum%10;//Fixedthemissingsemicolon
                }
                num=s;
        }

        if(num==1){
                System.out.println(n+"isaMagicNumber.");
        }else{
                System.out.println(n+"isnotaMagicNumber.");
                }
        }
}
```

5 MergeSort

A. ProgramInspection

1. Thereareseveralerrorsintheprogram:
2. Error 1: In the mergeSort method, the lines int[] left = leftHalf(array+1); and int[]right=rightHalf(array-1);shouldbecorrected.Itseemslikeanattemptto split the array, but it's not done correctly.

3. Error2:TheleftHalfandrightHalfmethodsareincorrect.Theyshouldreturn the correct halves of the array.

4. Error3:Themergemethodshouldhaveleftandrightarraysasinputs,not left++andright--.

5. The category of program inspection that would be most effective for this code is Category C: Computation Errors, as there are computation-related issues in the code.

6. Program inspection cannot identify runtime issues or logical errors that mightariseduringpro gramexecution.

7. The program inspection technique is worth applying to identify and fix computation-relatedissues.

B. Debugging

1. Therearemultipleerrorsintheprogram,asidentifiedabove.

2. Tofixtheseerrors,youwouldneedtosetbreakpointstoexaminethevalues ofleft,rightandarrayduringexecution.Youcanalsousebreakpointstocheck the values of i1 and i2 inside the merge method.

3. Thecorrectedexecutablecodeisasfollows:

```
importjava.util.*;
publicclassMergeSort{
publicstaticvoidmain(String[]args){
        int[]list={14,32,67,76,23,41,58,85};
        System.out.println("before: " + Arrays.toString(list));

        mergeSort(list);
        System.out.println("after:"+Arrays.toString(list));
}
public static void mergeSort(int[] array)
        { if(array.length>1){
                int[]left=leftHalf(array);
                int[]right=rightHalf(array);

        mergeSort(left); mergeSort(right);
        merge(array,left,right);
        }
}
publicstaticint[]leftHalf(int[]array){ intsize1=a
        rray.length/2;
        int[]left=newint[size1];

        for (int i = 0; i < size1; i++)
                { left[i]=array[i];
        }
returnleft;
}
```

```
publicstaticint[]rightHalf(int[]array){ intsize1=ar
        ray.length/2;
        int size2 = array.length - size1;
        int[]right=newint[size2];

        for(inti=0;i<size2;i++){ right[i]=array[i+s
                ize1];
        }
  returnright;
}
publicstaticvoidmerge(int[]result,int[]left,int[]right){ inti1=0;
        inti2=0;

for(inti=0;i<result.length;i++){
        if (i2 >= right.length || (i1 < left.length && left[i1] <= right[i2])) { result[i]=left[i1];
                i1++;
        }else{
                result[i]=right[i2]; i2++;
                }
        }
    }
}
```

6 MultiplyMatrices

A. ProgramInspection

1. Thereareseveralerrorsintheprogram:
2. Error1:Inthenestedloopsformatrixmultiplication,theloopindicesshould start from 0, not -1.

3. Error 2: The error message when the matrix dimensions are incompatible should print ”Matrices with entered orders can’t be multiplied with each other,”not”Matriceswithenteredorderscan’tbemultipliedwitheachother.”

4. The category of program inspection that would be most effective for this code is Category C: Computation Errors, as there are computation-related issues in the code.

5. Program inspection cannot identify runtime issues or logical errors that mightariseduringprogramexecution.

6. The program inspection technique is worth applying to identify and fix computation-relatedissues.

B. Debugging

1. Therearemultipleerrorsintheprogram,asidentifiedabove.
2. Tofixtheseerrors,youwouldneedtosetbreakpointstoexaminethevalues of c, d, k, and sum during execution. You should pay particular attention to the nested loops where the matrix multiplication occurs.

3. Thecorrectedexecutablecodeisasfollows:

```
import java.util.Scanner;
classMatrixMultiplication{
publicstaticvoidmain(Stringargs[]){ intm,n,p,q,
        sum=0,c,d,k;
        Scannerin=newScanner(System.in);

        System.out.println("Enterthenumberofrowsandcolumnsofthefirst matrix");

        m=in.nextInt();
        n=in.nextInt();
        intfirst[][]=newint[m][n];
        System.out.println("Entertheelementsofthefirstmatrix");

        for(c=0;c<m;c++)
                for(d=0;d<n;d++)
                        first[c][d]=in.nextInt();

        System.out.println("Enterthenumberofrowsandcolumnsofthe second matrix");
                p=in.nextInt();
                q=in.nextInt();

                if(n!=p)
                        System.out.println("Matriceswithenteredorderscan'tbe multiplied with
                each other.");
                else{
                        intsecond[][]=newint[p][q];
                        intmultiply[][]=newint[m][q];
```

```
                System.out.println("Entertheelementsofthesecondmatrix");

        for(c=0;c<p;c++)
            for(d=0;d<q;d++)
                second[c][d]=in.nextInt();

                for(c=0;c<m;c++){
                    for(d=0;d<q;d++){
                            for(k=0;k<p;k++){
                               sum=sum+first[c][k]*second[k][d];
                            }
                    multiply[c][d]=sum; sum = 0;
                    }
                }

        System.out.println("Productofenteredmatrices:-");

        for(c=0;c<m;c++){
                   for(d=0;d<q;d++)
                System.out.print(multiply[c][d]+"\t");

              System.out.print("\n");
                    }
            }
        }
}
```

7 QuadraticProbing

A. ProgramInspection

1. Therearemultipleerrorsintheprogram:
2. Error1:Theinsertmethodhasatypointhelinei+=(i+h/h–)

3. Error2:Intheremovemethod,thereisalogicerrorinthelooptorehash keys.Itshouldbei=(i+h*h++)

4. Error3:Inthegetmethod,thereisalogicerrorinthelooptofindthekey.It shouldbei=(i+h*h++)

5. The category of program inspection that would be most effective for this codeisCategoryA:SyntaxErrorsandCategoryB:SemanticErrors,asthere are both syntax errors and semantic issues in the code.

6. Theprograminspectiontechniqueisworthapplyingtoidentifyandfix theseerrors,butitmaynotidentifylogicalerrorsthataffecttheprogram's behaviour.

B. Debugging

1. Therearethreeerrorsintheprogram,asidentifiedabove.
2. Tofixtheseerrors,youwouldneedtosetbreakpointsandstepthroughthe codewhileexaminingvariableslikei,h,tmp1,andtmp2.Youshouldpay attention to the logic of the insert, remove and get methods.

3. Thecorrectedexecutablecodeisasfollows: import java.util.Scanner;

```
class QuadraticProbingHashTable
        { privateintcurrentSize,maxSize;
```

```
        privateString[]keys;
        privateString[]vals;

        public QuadraticProbingHashTable(int capacity) { currentSize = 0;
                maxSize=capacity;
                keys=newString[maxSize];
                vals=newString[maxSize];
        }
publicvoidmakeEmpty(){ currentSize = 0;
        keys=newString[maxSize];
        vals=aString[maxSize];
}
publicintgetSize(){
        returncurrentSize;
}
publicbooleanisFull(){
        returncurrentSize==maxSize;
}
publicbooleanisEmpty(){ return
        getSize()==0;
}
publicbooleancontains(Stringkey){ returnget(key)
        !=null;
}
privateinthash(Stringkey){
        returnkey.hashCode()%maxSize;
}
```

```
publicvoidinsert(Stringkey,Stringval){ int tmp =
        hash(key);
        inti=tmp,h=1;

        do{
                if(keys[i]==null){
                        keys[i] = key;
                        vals[i]=val;
                        currentSize++;
        return;
        }

        if(keys[i].equals(key)){ vals[i]=
                val;
        return;
        }

                i+=(h*h++)%maxSize;
        }while(i!=tmp);
}

publicStringget(Stringkey){ inti=has
        h(key),h=1;

        while(keys[i]!=null){
                if (keys[i].equals(key)) return
                        vals[i];
i=(i+h*h++)%maxSize;
}
```

```
        returnnull;
}
publicvoidremove(Stringkey){ if
        (!contains(key))
                return;

        inti=hash(key),h=1;

        while(!key.equals(keys[i]))
                i=(i+h*h++)%maxSize;

        keys[i]=vals[i]=null;
        for(i=(i+h*h++)%maxSize;keys[i]!=null;i=(i+h*h++)%maxSize)
                {
                Stringtmp1=keys[i],tmp2=vals[i];
                keys[i]=vals[i]=null;
                currentSize--;
                insert(tmp1,tmp2);
                }
currentSize--;
}
public void printHashTable()
        { System.out.println("\nHashTable:");

        for (int i = 0; i < maxSize; i++) if
                (keys[i] != null)
                        System.out.println(keys[i]+""+vals[i]);
System.out.println();
}
```

```
}
publicclassQuadraticProbingHashTableTest{ public static
void main(String[] args) {
        Scannerscan=newScanner(System.in);
        System.out.println("HashTableTest\n\n");
        System.println("Entersize");

        QuadraticProbingHashTable qpht = new
        QuadraticProbingHashTable(scan.nextInt());

        charch;

        do{
        System.out.println("\nHashTableOperations\n");
        System.out.println("1.insert"); System.out.println("2.
        remove"); System.out.println("3.get");
        System.out.println("4. clear"); System.out.println("5.size");
        intchoice=scan.nextInt(); switch
(choice) {
        case1:
                System.out.println("Enterkeyandvalue");
                qpht.insert(scan.next(),scan.next()); Break;
        case2:
                System.out.println("Enterkey");
                qpht.remove(scan.next()); Break;
```

```
            case3:
                System.out.println("Enterkey");
                System.out.println("Value="+qpht.get(scan.next())); Break;
            case4:
                qpht.makeEmpty();
                System.out.println("HashTableCleared\n"); Break;
            case5:
                System.out.println("Size = " + qpht.getSize()); Break;
            default:
                System.out.println("WrongEntry\n"); break;
            }
            qpht.printHashTable();
            System.out.println("\nDoyouwanttocontinue(Typeyorn)\n");
            ch=scan.next().charAt(0);

            }while(ch=='Y'||ch=='y');
        }
}
```

8 SortingArray

A. ProgramInspection

1. Errorsidentified:

2. Error1:Theclassname”AscendingOrder”containsanextraspaceandan underscore. The class name should be corrected to ”AscendingOrder.”

3. Error2:Thefirstnestedforloophasanincorrectloopconditionfor(inti=0; i¿=n;i++);„whichshouldbemodifiedtofor(inti=0;i¡n;i++).

4. Error3:Thereisanextrasemicolon(;)afterthefirstnestedforloop,which should be removed.

5. The most effective category of program inspection would be Category A: SyntaxErrorsandCategoryB:SemanticErrors,astherearebothsyntaxerrors and semantic issues in the code.

6. Programinspectionalonecanidentifyandfixsyntaxerrorsandsome semantic issues. However, it may not detect logic errors that affect the program’sbehavior.

7. The program inspection technique is worth applying to fix the syntax and semanticerrors,butdebuggingisrequiredtoaddresslogicerrors.

B. **Debugging**

1. Therearetwoerrorsintheprogramasidentifiedabove.

2. To fix these errors, you need to set breakpoints and step through the code. Youshouldfocusontheclassname,theloopconditions,andtheunnecessary semicolon.

3. Thecorrectedexecutablecodeisasfollows:

```
import java.util.Scanner;
publicclassAscendingOrder{
publicstaticvoidmain(String[]args){ int n,
        temp;
        Scanners=newScanner(System.in);
System.out.print("Enterthenumberofelementsyouwantinthearray:"); n=s.nextInt();
        inta[]=newint[n];System.out.println("Enterallltheelem
        ents:");

        for (int i = 0; i < n; i++)
                { a[i]=s.nextInt();
        }
        for(inti=0;i<n;i++){
                for (int j = i + 1; j < n; j++)
                        { if(a[i]>a[j]){
                                temp=a[i];
                                a[i]=a[j];a[j]=te
                                mp;
                        }
                }
        }
        System.out.print("Ascending Order: ");
        for(inti=0;i<n-1;i++){
                System.out.print(a[i]+",");
}
        System.out.print(a[n-1]);
        }
}
```

9 StackImplementation

A. ProgramInspection

1. Errorsidentified:
2. Error1: Thepushmethodhasadecrementoperationonthetopvariable (top–)insteadofanincrementoperation.Itshouldbecorrectedtotop++to push values correctly.

3. Error2:Thedisplaymethodhasanincorrectloopconditioninfor(inti=0;i¿ top;i++).Theloopconditionshouldbefor(inti=0;i¡=top;i++)tocorrectly display the elements.

4. Error3:ThepopmethodismissingintheStackMethodsclass.Itshouldbe added to provide a complete stack implementation.

5. The most effective category of program inspection would be Category A: SyntaxErrors,astherearesyntaxerrorsinthecode.Inaddition,CategoryB: Semantic Errors can help identify logic and functionality issues.

6. The program inspection technique is worth applying to identify and fix syntaxerrors,butadditionalinspectionisneededtoensurethelogicand functionality are correct.

B. Debugging

1. Therearethreeerrorsintheprogram,asidentifiedabove.
2. Tofixtheseerrors,youwouldneedtosetbreakpointsandstepthroughthe code, focusing on the push, pop, and display methods. Correct the push and display methods and add the missing pop method to provide a complete stack implementation.

3. Thecorrectedexecutablecodeisasfollows:

```
publicclassStackMethods{ private
        int top;
        int size;
        int[]stack;

        publicStackMethods(intarraySize){ size=arra
        ySize;
        stack=newint[size]; top=-
        1;
}

publicvoidpush(intvalue){ if(top
        ==size-1){
                System.out.println("Stackisfull,can'tpushavalue");
        }else{
                top++;
                stack[top]=value;
        }
}
      publicvoidpop(){
        if(!isEmpty()){
                  top--;
        }else{
                System.out.println("Can'tpop...stackisempty");
}
}
public boolean isEmpty()
        { return top == -1;
}
```

```
publicvoiddisplay(){
        for(inti=0;i<=top;i++){ System.out.print(stack[i] + " ");
        }
        System.out.println();
    }
}
```

10 TowerofHanoi

A. ProgramInspection

1. Errorsidentified:
2. Error1:InthelinedoTowers(topN++,inter–,from+1,to+1),thereareerrorsin the increment and decrement operators. It should be corrected to doTowers(topN-1,inter,from,to).

3. The most effective category of program inspection would be Category B: SemanticErrorsbecausetheerrorsinthecodearerelatedtologicand function.

4. The program inspection technique is worth applying to identify and fix semanticerrorsinthecode.

B. Debugging

1. Thereisoneerrorintheprogram,asidentifiedabove.
2. Tofixthiserror,youneedtoreplacetheline:

doTowers(topN ++, inter--, from+1, to+1);

3. with the correct version:

doTowers(topN-1,inter,from,to);

4. Thecorrectedexecutablecodeisasfollows: public

```
class MainClass {
public static void main(String[] args) { int
        nDisks = 3;doTowers(nDisks,'A','B','C');
}
publicstaticvoiddoTowers(inttopN,charfrom,charinter,charto){ if(topN==1){
                System.out.println("Disk1from"+from+"to"+to);
        }else{
                doTowers(topN-1,from,to,inter);
                System.out.println("Disk " + topN + " from " + from + " to " + to); doTowers(topN - 1,
                inter, from, to);
                }
        }
}
```

| . Program Inspection/DebuggingforLong-codefromGitHub

Wearegiventhefollowingchecklistandwehavetoffind allthepossibleerrorsaccordingly,

1. DatareferencingErrors
2. DatadeclarationErrors
3. ComputationErrors
4. ComparisonErrors
5. ControlFlowerrors
6. Interfaceerrors
7. Input/OutputErrors
8. OtherChecks

FirstcodeisaNodeJSffilefrom

https://github.com/Medium/medium-sdk-nodejs/blob/master/test/mediumClient_test.js

Thefirsthalfofthecodeisprovidedhere::

```
varmedium=require("../")
varnock=require("nock")
varqs=require('querystring')
varshould=require("should") var url =
require('url')


describe('MediumClient-constructor',function(){

    it('shouldthrowaMediumErrorwhenoptionsareundefined',function(done){ (function(){newmedium.MediumCli
        ent()}).should.throw(medium.MediumError) done()
    })

    it('shouldthrowaMediumErrorwhenoptionsareempty',function(done){ (function(){newmedium.MediumClient
        ({})}).should.throw(medium.MediumError) done()
    })

    it('shouldthrowaMediumErrorwhenonlyclientIdisprovided',function(done){ (function(){newmedium.Medium
        Client({clientId:'xxx'})
}).should.throw(medium.MediumError) done()
    })

    it('shouldthrowaMediumErrorwhenonlyclientSecretisprovided',function(done){ (function(){newmedium.MediumClient({clie
        ntSecret:'yyy'})
}).should.throw(medium.MediumError) done()
    })

    it('shouldsucceedwhenbothclientIdandclientSecretareprovided',function(done){ varclient=newmedium.MediumClient({
        clientId:'xxx',clientSecret:'yyy'})
        done()
    })
})


describe('MediumClient-methods',function(){
```

```
var clientId =
'xxx'varclientSecret='yyy' var
client

beforeEach(function(){
    client=newmedium.MediumClient({clientId:clientId,clientSecret:clientSecret}) nock.disableNetConnect()
})

afterEach(function ()
    { nock.enableNetConnect(); delete
    client
})

describe('#setAccessToken',function(){

    it('setstheaccesstoken',function(done){ var token = "new token"
        client.setAccessToken(token)
        client._accessToken.should.be.String().and.equal(token) done()
    })
})

describe('#getAuthorizationUrl',function(){

    it('returnsavalidURLforfetching',function(done){ var state = "state"
        varredirectUrl="https://example.com/callback"
        varscope=[medium.Scope.BASIC_PROFILE,medium.Scope.LIST_PUBLICATIONS, medium.Scope.PUBLISH_POST]
        varauthUrlStr=client.getAuthorizationUrl(state,redirectUrl,scope)
        varauthUrl=url.parse(authUrlStr,true) authUrl.protocol.should.equal('https:')
        authUrl.hostname.should.equal('medium.com')
        authUrl.pathname.should.equal('/m/oauth/authorize') authUrl.query.should.deepEqual({
            client_id: clientId, scope:
            scope.join(','), response_type:
            'code', state: state,
            redirect_uri:redirectUrl
        })
        done()
    })
})
```

```
describe('#exchangeAuthorizationCode',function(){

    it('makesarequestforauthorization_codeandsetstheaccesstokenfromresponse', function (done) {
        varcode='12345'
        vargrantType='authorization_code'
        varredirectUrl='https://example.com/callback'

            varrequestBody=qs.stringify({ code: code,
            client_id: clientId,
            client_secret:clientSecret, grant_type:
            grantType, redirect_uri: redirectUrl
        })
        //theresponsemighthaveotherparameters.thistestonlyconsiderstheonescalled

        //intheMediumNodeSDKdocumentation var
        accessToken = 'abcdef'
out
        varrefreshToken='ghijkl' var
        responseBody = {
            access_token:accessToken,
            refresh_token:refreshToken
        }
        varrequest=nock('https://api.medium.com/',{ 'Content-
                Type':'application/x-www-form-urlencoded'
            })
            .post('/v1/tokens',requestBody)
            .reply(201,responseBody)

        client.exchangeAuthorizationCode(code,redirectUrl,function(err,data){ if (err) throw err
            data.access_token.should.equal(accessToken)
            data.refresh_token.should.equal(refreshToken) done()
        })
        request.done()
    })
})

describe('#exchangeRefreshToken',function(){

    it('makesarequestforauthorization_codeandsetstheaccesstokenfromresponse', function (done) {
        varrefreshToken='fedcba'
        varaccessToken='lkjihg'
```

```
        varrequestBody=qs.stringify({ refresh_toke
            n: refreshToken, client_id: clientId,
            client_secret: clientSecret, grant_type:
            'refresh_token'
        })
        //theresponsemighthaveotherparameters.thistestonlyconsiderstheonescalled
out
        //intheMediumNodeSDKdocumentation var
        responseBody = {
            access_token:accessToken,
            refresh_token:refreshToken
        }
        var request = nock('https://api.medium.com/', { 'Content-
                Type':'application/x-www-form-urlencoded'
            })
            .post('/v1/tokens',requestBody)
            .reply(201,responseBody)

        client.exchangeRefreshToken(refreshToken,function(err,data){ if (err) throw err
            data.access_token.should.equal(accessToken)
            data.refresh_token.should.equal(refreshToken)
            done()
        })
        request.done()
    })
  })
```

1. DataReferencingErrors

- Nonefound.

2. Data DeclarationErrors

- Nonefound.

3. ComputationErrors

- Nonefound.

4. ComparisonErrors

- The assertion client._accessToken.should.be.String().and.equal(token) in the setAccessTokentestischeckingif_accessTokenisastringbeforeensuringits equality. The order of assertions could lead to an unhandled error if _accessTokenisnotdefinedorisnotastring.

5. ControlFlowErrors

- Nonefound.

6. InterfaceErrors

- Themethodclient.setAccessToken(token)iscalledinthesetAccessToken test, but if setAccessToken is not implemented correctly, it could lead to unexpected behavior.

7. Input/OutputErrors

- The request.done() line in both exchangeAuthorizationCode and exchangeRefreshTokentestsisincorrectlyplaced;itshouldbecalledafterthe request is executed, not directly after the request declaration. This may lead to premature invocation of done() in the context of network requests.

Nexthalfofthecodeishere::

```
describe('#getUser',function(){
    it('getstheinformationfromexpectedURLandreturnscontentsofdataenvelope', function (done) {
        varresponse={data:'responsedata'}

        varrequest=nock('https://api.medium.com')
            .get('/v1/me')
            .reply(200,response)

        client.getUser(function(err,data){ if (err) throw err
            data.should.deepEqual(response['data']) done()
```

```
        })
        request.done()
    })
})

describe('#getPublicationsForUser',function(){

    it('throwsaMediumErrorwhennouserIDisprovided',function(done){
        (function(){client.getPublicationsForUser({})}).should.throw(medium.MediumError) done()
    })

    it('makesaproperGETrequesttothemediumAPIandreturnscontentsofdataenvelope
whenvalidoptionsareprovided',function(done){
        varuserId='123456'
        varresponse={data:'responsedata'}

        varrequest=nock('https://api.medium.com/')
            .get('/v1/users/'+userId+'/publications')
            .reply(200,response)

        client.getPublicationsForUser({userId:userId},function(err,data){ if (err) throw err
            data.should.deepEqual(response['data']) done()
        })
        request.done()
    })
})

describe('#getContributorsForPublication',function(){

    it('throwsaMediumErrorwhennopublicationIDisprovided',function(done){ (function ()
        { client.getContributorsForPublication({})
}).should.throw(medium.MediumError) done()
    })

    it('makesaproperGETrequesttothemediumAPIandreturnscontentsofdataenvelope', function (done) {
        varoptions={publicationId:'abcdef'}
        varresponse={data:'responsedata'}
        varrequest=nock('https://api.medium.com/')
            .get('/v1/publications/'+options.publicationId+'/contributors')
            .reply(200, response) client.getContributorsForPublication(options,function(err,data){
```

```
            if (err) throw err
            data.should.deepEqual(response['data']) done()
        })
        request.done()
    })
})

describe('#createPost',function(){

    it('makesaproperPOSTrequesttotheMediumAPIandreturnscontentsofdataenvelope', function (done) {
        var options =
            { userId:'123456',
            title: 'new post title',
            content:'<h1>NewPost!</h1>',
            contentFormat: 'html',
            tags:['js','unittests'],
            canonicalUrl:'http://example.com/new-post',publishedAt:'2004-02-
            12T15:19:21+00:00',
            publishStatus: 'draft', license:'all-rights-
            reserved'
        }
        varresponse={data:'responsedata'}
        varrequest=nock('https://api.medium.com/')
            .post('/v1/users/'+options.userId+'/posts',{ title: options.title,
                    content: options.content,
                    contentFormat:options.contentFormat, tags:
                    options.tags,
                    canonicalUrl:options.canonicalUrl,
                    publishedAt:options.publishedAt,
                    publishStatus:options.publishStatus, license:
                    options.license
            })
            .reply(200,response)

        client.createPost(options,function(err,data){ if (err) throw err
            data.should.deepEqual(response['data'])
            done()
        })
        request.done()
    })
})

describe('#createPostInPublication',function(){
```

```
    it('shouldthrowanerrorwhennopublicationIDisprovided',function(done){ (function(){client.createPostInPublication({})}).s
        hould.throw(medium.MediumError) done()
    })

    it('makesaproperPOSTrequesttotheMediumAPIandreturnscontentsofdataenvelope', function (done) {
        var options =
            { publicationId:'abcdef',
            title:'newposttitle',
            content:'<h1>NewPost!</h1>',
            contentFormat: 'html',
            tags:['js','unittests'],
            canonicalUrl:'http://example.com/new-post',publishedAt: '2004-
            02-12T15:19:21+00:00',
            publishStatus: 'draft', license:'all-rights-
            reserved'
        }
        varresponse={data:'responsedata'}
        varrequest=nock('https://api.medium.com/')
            .post('/v1/publications/'+options.publicationId+'/posts',{ title: options.title,
                    content: options.content,
                    contentFormat:options.contentFormat, tags:
                    options.tags,
                    canonicalUrl: options.canonicalUrl, publishedAt:
                    options.publishedAt, publishStatus:options.publishStatus,
                    license: options.license
            })
            .reply(200,response)

        client.createPostInPublication(options,function(err,data){ if (err) throw err
            data.should.deepEqual(response['data']) done()
        })
        request.done()
    })
  })
})
```

Hereareetheidentifiederrorsclassifiedaccordingtoyourcategoriesinthe provided code:

1. DataReferencingErrors
- Nonefound.

2. Data DeclarationErrors
- Nonefound.

3. ComputationErrors
- Nonefound.

4. ComparisonErrors
- In the tests for getUser, getPublicationsForUser, getContributorsForPublication, createPost, and createPostInPublication, the assertion data.should.deepEqual(response['data']) assumes that response['data'] contains the expected output structure. If response.data is not defined correctly or is different in structure, this could lead to a comparison error.

5. ControlFlowErrors
- Nonefound.

6. InterfaceErrors
- Nonefound.
7. Input/OutputErrors
- Therequest.done()callineachtestshouldbeexecutedaftertherequestis processed (after the callback), rather than immediately after the request declaration. This could lead to improper handling of request expectations.

2ndCodeisasampleofsmallOperatingsystem

https://github.com/nuta/operating-system-in-1000-lines/tree/main

ThereareSubpartstoitsoIplacedalloftheminrandomorder…

```
#include"common.h"

void*memset(void*buf,charc,size_tn){ uint8_t *p = (uint8_t *)
        buf;
        while(n--)
                *p++=c; return
        buf;
}

void*memcpy(void*dst,constvoid*src,size_tn){ uint8_t *d = (uint8_t *) dst;
        constuint8_t*s=(constuint8_t*)src; while (n--)
                *d++=*s++; return
        dst;
}

char*strcpy(char*dst,constchar*src){ char *d = dst;
        while(*src)
                *d++=*src++;
        *d = '\0';
        returndst;
}

intstrcmp(constchar*s1,constchar*s2){ while (*s1 && *s2) {
                if(*s1!=*s2) break;
```

```
            s1++;
            s2++;
        }

        return*(unsignedchar*)s1-*(unsignedchar*)s2;
}

voidputchar(charch);

voidprintf(constchar*fmt,...){ va_list vargs;
        va_start(vargs,fmt); while
        (*fmt) {
            if(*fmt=='%'){ fmt++;
                switch(*fmt){ case '\0':
                                putchar('%'); goto
                                end;
                        case'%':
                                putchar('%'); break;
                        case's':{
                                constchar*s=va_arg(vargs,constchar*); while (*s) {
                                        putchar(*s); s++;
                                }
                                break;
                        }
                        case'd':{
                                intvalue=va_arg(vargs,int); if (value < 0) {
                                        putchar('-'); value=-
                                        value;
```

```
                        }

                        intdivisor=1;
                        while(value/divisor>9) divisor *= 10;

                        while(divisor>0){
                                putchar('0'+value/divisor); value %= divisor;
                                divisor/=10;
                        }

                        break;
                }
                case'x':{
                        intvalue=va_arg(vargs,int); for (int i = 7; i >=
                        0; i--) {
                                intnibble=(value>>(i*4))&0xf;
                                putchar("0123456789abcdef"[nibble]);
                        }
                }
        }
    }else{
            putchar(*fmt);
    }

    fmt++;
  }

end:
    va_end(vargs);
}
```

1. DataReferencingErrors

- Nonefound.

2. Data DeclarationErrors

- Intheprintffunction,theva_listvargsisdeclaredbutnotproperlyhandled. Ifva_end(vargs)iscalledwithoutacorrespondingva_start(vargs,fmt),itcould lead to undefined behavior, although this isn't directly indicated here since va_start is correctly used before va_end.

3. ComputationErrors

- Nonefound.

4. ComparisonErrors

- Nonefound.

5. ControlFlowErrors

- Intheprintffunction,thegotoend;statementinsidetheswitchblockcan create confusion. Although it is not an error, using goto can lead to less readable code and should be avoided if possible.

6. InterfaceErrors

- The putchar function is declared but not defined in the provided code. This could lead to linker errors if putchar is called without a definition available.
- Thefunctionprintfusesvariousformats(%d,%x,%s),butthereisnoerror handling for unsupported formats, which could lead to unpredictable behavior if an unsupported format specifier is encountered.

7. Input/OutputErrors

- Intheprintffunction,thereisnocheckforanullpointerintheconstchar*s =va_arg(vargs,constchar*);lineforthestringformatspecifier(%s).Ifanull pointer is passed, it could lead to dereferencing a null pointer and cause a segmentation fault.

```
#include"kernel.h"
#include"common.h"

extern char kernel_base[]; extern char
stack_top[];externcharbss[],bss_end[];
externcharfree_ram[],free_ram_end[];
externchar_binary_shell_bin_start[],_binary_shell_bin_size[];

structprocessprocs[PROCS_MAX]; struct
process *current_proc; struct process
*idle_proc;

paddr_talloc_pages(uint32_tn){
        staticpaddr_tnext_paddr=(paddr_t)free_ram; paddr_t paddr =
        next_paddr;
        next_paddr+=n*PAGE_SIZE;

        if(next_paddr>(paddr_t)free_ram_end) PANIC("out of
                memory");

        memset((void*)paddr,0,n*PAGE_SIZE); return paddr;
}

voidmap_page(uint32_t*table1,uint32_tvaddr,paddr_tpaddr,uint32_tflags){ if(!is_aligned(vaddr,PAGE_SIZE))
                PANIC("unalignedvaddr%x",vaddr);

        if(!is_aligned(paddr,PAGE_SIZE))
                PANIC("unalignedpaddr%x",paddr);

        uint32_tvpn1=(vaddr>>22)&0x3ff;
        if((table1[vpn1]&PAGE_V)==0){
                uint32_tpt_paddr=alloc_pages(1);
                table1[vpn1]=((pt_paddr/PAGE_SIZE)<<10)|PAGE_V;
        }
```

```
        uint32_tvpn0=(vaddr>>12)&0x3ff;
        uint32_t*table0=(uint32_t*)((table1[vpn1]>>10)*PAGE_SIZE);
        table0[vpn0]=((paddr/PAGE_SIZE)<<10)|flags|PAGE_V;
}

structsbiretsbi_call(longarg0,longarg1,longarg2,longarg3,longarg4, longarg5,longfid,longeid){
        registerlonga0asm("a0")=arg0;
        registerlonga1asm("a1")=arg1;
        registerlonga2asm("a2")=arg2;
        registerlonga3asm("a3")=arg3;
        registerlonga4asm("a4")=arg4;
        registerlonga5asm("a5")=arg5;
        registerlonga6asm("a6")=fid;
        registerlonga7asm("a7")=eid;

        asmvolatile("ecall"
                                        :"=r"(a0),"=r"(a1)
                                        :"r"(a0),"r"(a1),"r"(a2),"r"(a3),"r"(a4),"r"(a5), "r"(a6), "r"(a7)
                                        :"memory");
        return(structsbiret){.error=a0,.value=a1};
}

structvirtio_virtq*blk_request_vq; struct
virtio_blk_req *blk_req; paddr_t blk_req_paddr;
unsignedblk_capacity;

uint32_tvirtio_reg_read32(unsignedoffset){
        return*((volatileuint32_t*)(VIRTIO_BLK_PADDR+offset));
}

uint64_tvirtio_reg_read64(unsignedoffset){
        return*((volatileuint64_t*)(VIRTIO_BLK_PADDR+offset));
}

voidvirtio_reg_write32(unsignedoffset,uint32_tvalue){
        *((volatileuint32_t*)(VIRTIO_BLK_PADDR+offset))=value;
}

voidvirtio_reg_fetch_and_or32(unsignedoffset,uint32_tvalue){ virtio_reg_write32(offset,virtio_reg_read
        32(offset)|value);
}

boolvirtq_is_busy(structvirtio_virtq*vq){ returnvq
        ->last_used_index!=*vq->used_index;
```

```
}

voidvirtq_kick(structvirtio_virtq*vq,intdesc_index){
        vq->avail.ring[vq->avail.index%VIRTQ_ENTRY_NUM]=desc_index; vq->avail.index++;
        sync_synchronize(); virtio_reg_write32(VIRTIO_REG_QUEUE_NOTIFY,vq
        ->queue_index); vq->last_used_index++;
}

structvirtio_virtq*virtq_init(unsignedindex){
        paddr_tvirtq_paddr=alloc_pages(align_up(sizeof(structvirtio_virtq),PAGE_SIZE)/ PAGE_SIZE);
        structvirtio_virtq*vq=(structvirtio_virtq*)virtq_paddr; vq->queue_index = index;
        vq->used_index = (volatile uint16_t *) &vq->used.index;
        virtio_reg_write32(VIRTIO_REG_QUEUE_SEL, index);
        virtio_reg_write32(VIRTIO_REG_QUEUE_NUM,VIRTQ_ENTRY_NUM);
        virtio_reg_write32(VIRTIO_REG_QUEUE_ALIGN, 0);
        virtio_reg_write32(VIRTIO_REG_QUEUE_PFN, virtq_paddr); return vq;
}

voidvirtio_blk_init(void){
        if(virtio_reg_read32(VIRTIO_REG_MAGIC)!=0x74726976) PANIC("virtio: invalid magic
                value");
        if(virtio_reg_read32(VIRTIO_REG_VERSION)!=1) PANIC("virtio: invalid
                version");
        if(virtio_reg_read32(VIRTIO_REG_DEVICE_ID)!=VIRTIO_DEVICE_BLK) PANIC("virtio: invalid device id");

        virtio_reg_write32(VIRTIO_REG_DEVICE_STATUS, 0);
        virtio_reg_fetch_and_or32(VIRTIO_REG_DEVICE_STATUS,VIRTIO_STATUS_ACK);
        virtio_reg_fetch_and_or32(VIRTIO_REG_DEVICE_STATUS, VIRTIO_STATUS_DRIVER);
        virtio_reg_fetch_and_or32(VIRTIO_REG_DEVICE_STATUS,VIRTIO_STATUS_FEAT_OK); blk_request_vq =
        virtq_init(0); virtio_reg_write32(VIRTIO_REG_DEVICE_STATUS,VIRTIO_STATUS_DRIVER_OK);

        blk_capacity=virtio_reg_read64(VIRTIO_REG_DEVICE_CONFIG+0)*SECTOR_SIZE; printf("virtio-blk:
        capacity is %d bytes\n", blk_capacity);

        blk_req_paddr=alloc_pages(align_up(sizeof(*blk_req),PAGE_SIZE)/PAGE_SIZE); blk_req = (struct
        virtio_blk_req *) blk_req_paddr;
}
```

1. DataReferencingErrors
   - Noneidentified.

2. Data DeclarationErrors
   - Noneidentified.

3. ComputationErrors
   - Noneidentified.

4. ComparisonErrors
   - Noneidentified.

5. ControlFlowErrors
   - Nocheckforsuccessfulallocationinvirtq_init()afteralloc_pages().This could lead to dereferencing a NULL pointer.

6. InterfaceErrors
   - Noexplicitvalidationforregisteroffsetsinvirtio_reg_read32, virtio_reg_read64, and related functions.

7. Input/OutputErrors
   - Noneidentified.

```
voidread_write_disk(void*buf,unsignedsector,intis_write){ if(sector>=blk_capacity/SECT
    OR_SIZE){
        printf("virtio:triedtoread/writesector=%d,butcapacityis%d\n", sector,blk_capacity/SECTOR_SIZE);
        return;
    }

    blk_req->sector=sector;
    blk_req->type=is_write?VIRTIO_BLK_T_OUT:VIRTIO_BLK_T_IN;

    if(is_write)
        memcpy(blk_req->data,buf,SECTOR_SIZE);
```

```
        structvirtio_virtq*vq=blk_request_vq; vq
        ->descs[0].addr=blk_req_paddr;
        vq->descs[0].len=sizeof(uint32_t)*2+sizeof(uint64_t); vq
        ->descs[0].flags=VIRTQ_DESC_F_NEXT;
        vq->descs[0].next=1;

        vq->descs[1].addr=blk_req_paddr+offsetof(structvirtio_blk_req,data); vq->descs[1].len =
        SECTOR_SIZE;
        vq->descs[1].flags=VIRTQ_DESC_F_NEXT|(is_write?0:VIRTQ_DESC_F_WRITE); vq->descs[1].next = 2;

        vq->descs[2].addr=blk_req_paddr+offsetof(structvirtio_blk_req,status); vq
        ->descs[2].len=sizeof(uint8_t);
        vq->descs[2].flags=VIRTQ_DESC_F_WRITE;

        virtq_kick(vq,0);
        while(virtq_is_busy(vq))
                ;

        if(blk_req->status!=0){
                printf("virtio:warn:failedtoread/writesector=%dstatus=%d\n", sector, blk_req->status);
                return;
        }

        if(!is_write)
                memcpy(buf,blk_req->data,SECTOR_SIZE);
}

structfilefiles[FILES_MAX];
uint8_tdisk[DISK_MAX_SIZE];

intoct2int(char*oct,intlen){ int dec = 0;
        for(inti=0;i<len;i++){
                if(oct[i]<'0'||oct[i]>'7') break;

                dec=dec*8+(oct[i]-'0');
        }
        returndec;
}

void fs_flush(void)
        {memset(disk,0,sizeof(disk)); unsigned
        off = 0;
```

```
    for(intfile_i=0;file_i<FILES_MAX;file_i++){ structfile*file=&files[file_i];
        if(!file->in_use) continue;

        structtar_header*header=(structtar_header*)&disk[off]; memset(header,0,sizeof(*header));
        strcpy(header->name,file->name);
        strcpy(header->mode, "000644");
        strcpy(header->magic, "ustar");
        strcpy(header->version, "00");
        header->type='0';

        intfilesz=file->size;
        for(inti=sizeof(header->size);i>0;i--){ header->size[i-
            1]=(filesz%8)+'0';
            filesz/=8;
        }

        intchecksum=''*sizeof(header->checksum);
        for(unsignedi=0;i<sizeof(structtar_header);i++)
            checksum+=(unsignedchar)disk[off+i];

        for(inti=5;i>=0;i--){
            header->checksum[i]=(checksum%8)+'0'; checksum /= 8;
        }

        memcpy(header->data,file->data,file->size);
        off+=align_up(sizeof(structtar_header)+file->size,SECTOR_SIZE);
    }

    for(unsignedsector=0;sector<sizeof(disk)/SECTOR_SIZE;sector++)
        read_write_disk(&disk[sector*SECTOR_SIZE],sector,true);

    printf("wrote%dbytestodisk\n",sizeof(disk));
}

voidfs_init(void){
    for(unsignedsector=0;sector<sizeof(disk)/SECTOR_SIZE;sector++)
        read_write_disk(&disk[sector*SECTOR_SIZE],sector,false);

    unsignedoff=0;
    for(inti=0;i<FILES_MAX;i++){
        structtar_header*header=(structtar_header*)&disk[off]; if (header->name[0] == '\0')
            break;
```

```
        if(strcmp(header->magic,"ustar")!=0)
                PANIC("invalidtarheader:magic=\"%s\"",header->magic);

        intfilesz=oct2int(header->size,sizeof(header->size)); struct file *file = &files[i];
        file->in_use=true;
        strcpy(file->name, header->name); memcpy(file
        ->data,header->data,filesz); file->size = filesz;
        printf("file:%s,size=%d\n",file->name,file->size);

        off+=align_up(sizeof(structtar_header)+filesz,SECTOR_SIZE);
    }
}
```

1. DataReferencingErrors
   - Thecodereferencesblk_req,blk_capacity,blk_request_vq,and blk_req_paddr without showing their definitions. Make sure these variables are properly initialized and referenced.

2. Data DeclarationErrors
   - Thevariablediskisdeclaredwithuint8_tdisk[DISK_MAX_SIZE];,butthere's no indication of the value assigned to DISK_MAX_SIZE. Ensure it's defined somewhere.
   - Thestructtar_headerisreferencedwithoutadeclarationintheprovided code. Ensure it is defined correctly in your project.

3. ComputationErrors
   - Thecalculationoffileszinfs_flushdoesnotaccountforthepotential overflow when calculating the checksum. Although the tar format specifies a maximumsize,it'sagoodpracticetochecksizestoavoidoverflow.

- In the oct2int function, if the input oct string has more than three characters(whichrepresentavalidoctaldigit),theconversionmightgive unexpected results. Consider adding a limit on len.

4. ComparisonErrors
   - Infs_init,thecheckif(strcmp(header->magic,"ustar")!=0)isvalid,butthe code doesn't handle the case where header->magic could be NULL. Consider adding a NULL check before comparison.

5. ControlFlowErrors
   - The read_write_disk function might enter an infinite loop if the diskrequestisnevercompleted.Ensurethatvirtq_kick(vq,0)andvirtq_is_busy(vq) are implemented correctly to handle this situation.
   - Thefunctionfs_flushwillprintthatithaswrittentothediskregardlessof whether the write was successful. Consider checking for errors in read_write_disk.

6. InterfaceErrors
   - The putchar function must be defined elsewhere, or else there will be linking errors when compiling.
   - Ensure that align_up is properly defined and that its purpose is clear; it seems to be intended for aligning data sizes, but its implementation is not providedhere.

7. Input/OutputErrors
   - Inthefs_flushfunction,whenwritingtothedisk,ifread_write_diskfailsfor anyreason(e.g.,duetoafulldiskorhardwarefailure),theuserisnotnotified. Implement error handling to manage this.

- Inthefs_initfunction,ifthedatareadfromthediskdoesn’tmatchthe expected format or the file size exceeds DISK_MAX_SIZE, it may cause out-of-bounds memory access when populating the file structures.

```
structfile*fs_lookup(constchar*filename){ for(inti=0;i<FILES_MA
    X;i++){
        structfile*file=&files[i];
        if(!strcmp(file->name,filename)) return file;
    }

    returnNULL;
}

voidputchar(charch){
    sbi_call(ch,0,0,0,0,0,0,1/*ConsolePutchar*/);
}

longgetchar(void){
    structsbiretret=sbi_call(0,0,0,0,0,0,0,2); return ret.error;
}

attribute((naked))
attribute((aligned(4))) void
kernel_entry(void) {
    asmvolatile(
        "csrrwsp,sscratch,sp\n" "addi sp, sp, -4 *
        31\n"
        "sw   ra,    4  *0(sp)\n"
        "sw   gp,    4  *1(sp)\n"
        "sw   tp,    4  *2(sp)\n"
        "sw   t0,    4  *3(sp)\n"
        "sw   t1,    4  *4(sp)\n"
        "sw   t2,    4  *5(sp)\n"
        "sw   t3,    4  *6(sp)\n"
        "sw   t4,    4  *7(sp)\n"
        "sw   t5,    4  *8(sp)\n"
        "sw   t6,    4  *9(sp)\n"
        "sw   a0,    4  *10(sp)\n"
        "sw   a1,    4  *11(sp)\n"
        "sw   a2,    4  *12(sp)\n"
        "sw   a3,    4  *13(sp)\n"
        "sw   a4,    4  *14(sp)\n"

        "swa5,4*15(sp)\n"
```

```
        "sw   a6,     4  *16(sp)\n"
        "sw   a7,     4  *17(sp)\n"
        "sw   s0,     4  *18(sp)\n"
        "sw   s1,     4  *19(sp)\n"
        "sw   s2,     4  *20(sp)\n"
        "sw   s3,     4  *21(sp)\n"
        "sw   s4,     4  *22(sp)\n"
        "sw   s5,     4  *23(sp)\n"
        "sw   s6,     4  *24(sp)\n"
        "sw   s7,     4  *25(sp)\n"
        "sw   s8,     4  *26(sp)\n"
        "sw   s9,     4  *27(sp)\n"
        "sw   s10,    4  *28(sp)\n"
        "sw   s11,    4  *29(sp)\n"

        "csrra0,sscratch\n"
        "swa0,4*30(sp)\n"

        "addia0,sp,4*31\n" "csrw
        sscratch, a0\n"

        "mva0,sp\n"
        "callhandle_trap\n"

        "lw   ra,     4  *0(sp)\n"
        "lw   gp,     4  *1(sp)\n"
        "lw   tp,     4  *2(sp)\n"
        "lw   t0,     4  *3(sp)\n"
        "lw   t1,     4  *4(sp)\n"
        "lw   t2,     4  *5(sp)\n"
        "lw   t3,     4  *6(sp)\n"
        "lw   t4,     4  *7(sp)\n"
        "lw   t5,     4  *8(sp)\n"
        "lw   t6,     4  *9(sp)\n"
        "lw   a0,     4  *10(sp)\n"
        "lw   a1,     4  *11(sp)\n"
        "lw   a2,     4  *12(sp)\n"
        "lw   a3,     4  *13(sp)\n"
        "lw   a4,     4  *14(sp)\n"
        "lw   a5,     4  *15(sp)\n"
        "lw   a6,     4  *16(sp)\n"
        "lw   a7,     4  *17(sp)\n"
        "lw   s0,     4  *18(sp)\n"
        "lw   s1,     4  *19(sp)\n"
        "lw   s2,     4  *20(sp)\n"
        "lw   s3,     4  *21(sp)\n"
        "lw   s4,     4  *22(sp)\n"
```

```
            "lws5,4*23(sp)\n"
            "lws6,4*24(sp)\n"
            "lws7,4*25(sp)\n"
            "lws8,4*26(sp)\n"
            "lws9,4*27(sp)\n"
            "lws10,4*28(sp)\n"
            "lws11,4*29(sp)\n"
            "lwsp,4*30(sp)\n" "sret\n"
    );
}

attribute((naked))voiduser_entry(void){
    asmvolatile( "csrwsepc,%[sepc]\n"
            "csrwsstatus,%[sstatus]\n" "sret\n"
            :
            :[sepc]"r"(USER_BASE),
             [sstatus]"r"(SSTATUS_SPIE|SSTATUS_SUM)
    );
}

             attribute((naked))voidswitch_context(uint32_t*prev_sp,
                                                  uint32_t*next_sp){
    asmvolatile(
            "addisp,sp,-13*4\n"
            "sw    ra,     0       *4(sp)\n"
            "sw    s0,     1       *4(sp)\n"
            "sw    s1,     2       *4(sp)\n"
            "sw    s2,     3       *4(sp)\n"
            "sw    s3,     4       *4(sp)\n"
            "sw    s4,     5       *4(sp)\n"
            "sw    s5,     6       *4(sp)\n"
            "sw    s6,     7       *4(sp)\n"
            "sw    s7,     8       *4(sp)\n"
            "sw    s8,     9       *4(sp)\n"
            "sw    s9,     10      *4(sp)\n"
            "sw    s10,    11      *4(sp)\n"
            "sw    s11,    12      *4(sp)\n"
            "sw   sp,(a0)\n"
            "lw   sp,(a1)\n"
            "lw   ra,0*4(sp)\n"
            "lw   s0,1*4(sp)\n"
            "lw   s1,2*4(sp)\n"
            "lw   s2,3*4(sp)\n"
           "lws3,4*4(sp)\n"
```

```
                "lws4,    5    *4(sp)\n"
                "lws5,    6    *4(sp)\n"
                "lws6,    7    *4(sp)\n"
                "lws7,    8    *4(sp)\n"
                "lws8,    9    *4(sp)\n"
                "lws9,    10   *4(sp)\n"
            "lws10,11*4(sp)\n"
            "lws11,12*4(sp)\n"
            "addisp,sp,13*4\n" "ret\n"
    );
}

structprocess*create_process(constvoid*image,size_timage_size){ struct process *proc = NULL;
    inti;
    for(i=0;i<PROCS_MAX;i++){
        if(procs[i].state==PROC_UNUSED){ proc = &procs[i];
            break;
        }
    }

    if(!proc)
        PANIC("nofreeprocessslots");

    uint32_t*sp=(uint32_t*)&proc->stack[sizeof(proc->stack)];



     *--sp   = 0;                      //  s11
     *--sp   = 0;                      //  s10
     *--sp   = 0;                      //  s9
     *--sp   = 0;                      //  s8
     *--sp   = 0;                      //  s7
     *--sp   = 0;                      //  s6
     *--sp   = 0;                      //  s5
     *--sp   = 0;                      //  s4
     *--sp   = 0;                      //  s3
     *--sp   = 0;                      //  s2
     *--sp   = 0;                      //  s1
     *--sp   = 0;                      //  s0

    *--sp=(uint32_t)user_entry;//ra

    uint32_t*page_table=(uint32_t*)alloc_pages(1);

    //Kernelpages.
    for(paddr_tpaddr=(paddr_t)kernel_base;
        paddr<(paddr_t)free_ram_end;paddr+=PAGE_SIZE)
        map_page(page_table,paddr,paddr,PAGE_R|PAGE_W|PAGE_X);
```

```
    //virtio-blk
    map_page(page_table,VIRTIO_BLK_PADDR,VIRTIO_BLK_PADDR,PAGE_R|PAGE_W);

    //Userpages.
    for(uint32_toff=0;off<image_size;off+=PAGE_SIZE){ paddr_t page =
        alloc_pages(1);
        memcpy((void*)page,image+off,PAGE_SIZE); map_page(page_table, USER_BASE
        + off, page,
                    PAGE_U|PAGE_R|PAGE_W|PAGE_X);
    }

    proc->pid=i+1;
    proc->state = PROC_RUNNABLE; proc
    ->sp = (uint32_t) sp; proc
    ->page_table=page_table; return proc;
}
```

1. DataReferencingErrors

- Potential Null Pointer Dereference: The fs_lookup function assumes that files isinitializedandvalid.IffilesisuninitializedorifFILES_MAXissetto0,itmay lead to undefined behavior.

2. Data DeclarationErrors

- MissingStructDefinition:Thestructfileandthefilesarrayarereferencedbut notdefinedintheprovidedcode.Thiscouldleadtocompilationerrorsifthey arenotdeclaredelsewhereintheprogram.

3. ComputationErrors

- Nonefound.

4. ComparisonErrors

- Nonefound.

5. ControlFlowErrors

- UnconditionalExit:ThePANIC("nofreeprocessslots");calldoesnothandle thecasewhereprocisNULLgracefully,potentiallyleadingtoabrupt terminationoftheprogram.Instead,itshouldideallyreturnorcleanup resources.

6. InterfaceErrors

- Nonefound.

7. Input/OutputErrors

- BufferOverrunRisk:Theloopthatinitializesthestack(with*--sp=0;) assumesthatthestackhassufficientspace.Ifthesizeofproc->stackisless than expected, it may result in a stack overflow.

```
voidyield(void){
    structprocess*next=idle_proc;for(inti=0;i<PROC
    S_MAX;i++){
        structprocess*proc=&procs[(current_proc->pid+i)%PROCS_MAX]; if(proc
        ->state==PROC_RUNNABLE&&proc->pid>0){
            next=proc; break;
        }
    }

    if(next==current_proc) return;

    structprocess*prev=current_proc; current_proc = next;

    asmvolatile( "sfence.vma\n"
        "csrwsatp,%[satp]\n"
        "sfence.vma\n"
        "csrwsscratch,%[sscratch]\n"
        :
        :[satp]"r"(SATP_SV32|((uint32_t)next->page_table/PAGE_SIZE)), [sscratch]"r"((uint32_t)&next
            ->stack[sizeof(next->stack)])
    );

    switch_context(&prev->sp,&next->sp);
}
```

```
voidhandle_syscall(structtrap_frame*f){ switch (f->a3) {
            case SYS_PUTCHAR:
                putchar(f->a0);
                break;
            caseSYS_GETCHAR:
                while (1) {
                    longch=getchar(); if (ch >=
                    0) {
                        f->a0=ch;
                        break;
                    }

                    yield();
                }
                break;
            caseSYS_EXIT:
                printf("process%dexited\n",current_proc->pid); current_proc
                ->state=PROC_EXITED;
                yield(); PANIC("unreachable");
            caseSYS_READFILE:
            caseSYS_WRITEFILE:{
                constchar*filename=(constchar*)f->a0; char *buf = (char *) f
                ->a1;
                intlen=f->a2;
                structfile*file=fs_lookup(filename); if (!file) {
                    printf("filenotfound:%s\n",filename); f->a0 = -1;
                    break;
                }

                if(len>(int)sizeof(file->data)) len = file->size;

                if(f->a3==SYS_WRITEFILE){
                    memcpy(file->data,buf,len); file->size =
                    len; fs_flush();
                }else{
                    memcpy(buf,file->data,len);
                }

                f->a0=len; break;
```

```
            }
            default:
                PANIC("unexpectedsyscalla3=%x\n",f->a3);
        }
}

voidhandle_trap(structtrap_frame*f){ uint32_tscause=
        READ_CSR(scause);
        uint32_tstval=READ_CSR(stval);
        uint32_tuser_pc=READ_CSR(sepc); if (scause
        == SCAUSE_ECALL) {
            handle_syscall(f); user_pc += 4;
        }else{
            PANIC("unexpectedtrapscause=%x,stval=%x,sepc=%x\n",scause,stval,user_pc);
        }

        WRITE_CSR(sepc,user_pc);
}

voidkernel_main(void){
        memset(bss,0,(size_t)bss_end-(size_t)bss); printf("\n\n");
        WRITE_CSR(stvec,(uint32_t)kernel_entry); virtio_blk_init();
        fs_init();

        idle_proc=create_process(NULL,0); idle_proc
        ->pid = -1; // idle current_proc = idle_proc;

        create_process(_binary_shell_bin_start,(size_t)_binary_shell_bin_size); yield();

        PANIC("switchedtoidleprocess");
}

attribute((section(".text.boot")))
attribute((naked)) void boot(void)
{
        asmvolatile(
            "mvsp,%[stack_top]\n" "j
            kernel_main\n"
            :
            :[stack_top]"r"(stack_top)
        );
}
```

DataReferencingErrors

- Potential Null Pointer Dereference: current_proc could be null if no processeshavebeencreatedorifithasbeenimproperlyinitializedbefore yield() is called.

Data DeclarationErrors

- UninitializedVariables:Variablessuchasidle_procandcurrent_procmaybe used without proper initialization if create_process fails or if there are no processes.

ComputationErrors

- ImproperMemoryAccess:Thecalculationofnext->page_table/PAGE_SIZE could lead to incorrect values if next->page_table is not properly aligned or initialized.

ComparisonErrors

- Unsignedvs.SignedComparison:Comparingproc->pid>0maycause unintended behavior if proc->pid is an unsigned type.

ControlFlowErrors

- InfiniteLoopRisk:Thewhile(1)loopinhandle_syscallforSYS_GETCHARmay leadtoaninfiniteloopifgetchar()neverreturnsavalidcharacter.

InterfaceErrors

- MissingErrorHandlingforSystemCalls:Functionslikefs_lookup,memcpy, and printf may fail silently without error checking or reporting in certain scenarios.

Input/OutputErrors

- DataOverwriteRisk:Inhandle_syscallforSYS_WRITEFILE,iflenisnot properlyvalidated,itmayleadtowritingbeyondtheboundsoffile->data.

```
#include"user.h"

voidmain(void){ while (1) {
prompt:
            printf(">
            ");charcmdline[128];
            for(inti=0;;i++){ charch=getchar
                  (); putchar(ch);
                  if (i == sizeof(cmdline) - 1)
                        { printf("commandlinetoolong\n"); goto prompt;
                  }elseif(ch=='\r'){
                        printf("\n");
                        cmdline[i]='\0'; break;
                  }else{
                        cmdline[i]=ch;
                  }
            }

            if(strcmp(cmdline,"hello")==0) printf("Helloworldfromshell!\n");
            elseif(strcmp(cmdline,"exit")==0) exit();
            elseif(strcmp(cmdline,"readfile")==0){ char buf[128];
                  intlen=readfile("hello.txt",buf,sizeof(buf)); buf[len] = '\0';
                  printf("%s\n",buf);
            }
            elseif(strcmp(cmdline,"writefile")==0) writefile("hello.txt","Hellofromshell!\n",19);
            else
                  printf("unknowncommand:%s\n",cmdline);
      }
}
```

DataReferencingErrors
- Potential Buffer Overflow: The cmdline buffer is not properly null-terminated iftheuserinputsmorethan127characters(sinceonebyteisusedforthenull terminator).

Data DeclarationErrors
- Uninitialized Variable: The variable buf in the readfile command could be uninitialized if the file reading fails before it is populated.

ComputationErrors
- Length Calculation: In the readfile command, the length returned by readfile()isuseddirectlywithoutcheckingifitexceedsthesizeofbuf.Iflenis largerthan128,thiscouldleadtoabufferoverflowwhensettingbuf[len]= '\0';.

ComparisonErrors
- Useofstrcmp:Ifcmdlineisnotproperlynull-terminatedduetobuffer overfloworamissedterminationcase,thebehaviorofstrcmpcanbe undefined.

ControlFlowErrors
- InfiniteLoopRisk:Thewhile(1)loopwillrunindefinitelyunlessacommand that calls exit() is executed. There's no condition to break out of the loop except for exit().

InterfaceErrors
- MissingErrorHandling:Thereturnvalueofreadfileisnotcheckedforerrors. Ifthefiledoesnotexistorreadfails,itcouldleadtoundefinedbehavior.

Input/OutputErrors

- Data Overwrite Risk: In the writefile command, there is no check to ensure thatthedatabeingwrittenislessthanorequaltothelengthofthefilebuffer on the file system.

```
#include"user.h"

externcharstack_top[];

int syscall(int sysno, int arg0, int arg1, int arg2) { register int a0 asm("a0") =
        arg0;
        register int a1 asm("a1") = arg1; register int a2
        asm("a2") = arg2; register int a3 asm("a3") = sysno;

        asm_____volatile("ecall"
                                        :"=r"(a0)
                                        :"r"(a0),"r"(a1),"r"(a2),"r"(a3)
                                        :"memory");

        returna0;
}

voidputchar(charch){
        syscall(SYS_PUTCHAR,ch,0,0);
}

intgetchar(void){
        returnsyscall(SYS_GETCHAR,0,0,0);
}

intreadfile(constchar*filename,char*buf,intlen){
        returnsyscall(SYS_READFILE,(int)filename,(int)buf,len);
}

intwritefile(constchar*filename,constchar*buf,intlen){
```

```
        returnsyscall(SYS_WRITEFILE,(int)filename,(int)buf,len);
}

attribute((noreturn)) void exit(void) { syscall(SYS_EXIT, 0, 0, 0);
        for(;;);
}

attribute((section(".text.start")))
attribute((naked)) void
start(void) {
        asm_____volatile(
                "mv sp, %[stack_top]\n" "call main\n"
                "callexit\n"::[stack_top]"r"(stack_top));
}
```

DataReferencingErrors

- CastingPointerstoIntegers:Thecodecastsconstchar*filenameandchar *buftoint,whichcanleadtodatalossorcorruptiononarchitectureswhere pointersarelargerthanintegers(e.g.,64-bitsystems).

Data DeclarationErrors

- UninitializedVariables:Ifsyscallfailsorreturnsanerrorvalue,thevariables buf and filename may not be handled properly in readfile and writefile functions, which could lead to unexpected behavior.

ComputationErrors

- ReturnValueIgnored:Inreadfileandwritefile,thereturnvaluefromsyscallis notchecked.Ifthesyscallfails(e.g.,filenotfound),thiscouldleadto undefined behavior when using the data later.

ComparisonErrors

- Noapparentcomparisonerrorsexistintheprovidedcode.

ControlFlowErrors

- EndlessLoopinexit:Thefor(;;);loopintheexitfunctionwillcreateaninfinite loop after the syscall call, which could indicate a lack of proper termination or errorhandling.

InterfaceErrors

- NoErrorHandlingforSystemCalls:Thereisnoerrorcheckingforthereturn valuesofsyscallinanyfunction.Forinstance,ifafileoperationfails,theerror is not handled.

Input/OutputErrors

- InvalidMemoryAccess:Ifbufinreadfileorwritefilepointstoaninvalidor unallocatedmemoryaddress,thecodewillattempttoreadfromorwriteto thatmemorylocation,leadingtopotentialcrashesordatacorruption.