实验设计者：20144464 翟佳羽

实验题目：银行家算法

实验目的：通过模拟银行家算法来了解操作系统解决死锁的方法

实现内容：

通过JS和html模拟了银行家算法的过程，通过输入进程数来确定要执行的进程数量，

再手动输入每个资源的Max，Allocation，Need，Available的值。之后通过安全性算法来检测当前是否安全安全性算法为

function result() {

var aNeedValue = aNeedVal;

var iAvailableValue = iAvailableVal;

var aAllocationValue = aAllocationVal;

var iFlag = -1;

console.log(aNeedValue.length);

if(aNeedValue.length == 0) {

alert('程序安全');

alert("安全序列为"+aSaveQueue);

addRequest();

}else{

for(var i=0;i<aNeedValue.length;i++) {

if(iAvailableValue>(aNeedValue[i]-1)) {

console.log(iAvailableValue + "available");

console.log(aNeedValue[i] + 'need');

iFlag = i;

aNeedValue.splice(i,1);

aSaveQueue.push(aProcess[i]);

aProcess.splice(i,1);

break;

}

}

if(iFlag!= -1) {

console.log(aAllocationValue[iFlag] + "allocation");

iAvailableVal = parseInt(iAvailableValue)+ parseInt(aAllocationValue[iFlag]);

console.log(iAvailableValue + "available");

result();

}else {

alert("系统不安全");

}

}

如果系统安全则输入Request的值，进行运算后在进行安全性检查算法

不安全则弹出提示框“系统不安全”。

实验结果：

基本实现了银行家算法，完成了安全性算法的模拟，并且测试成功。

实验心得：

这个课题对于我来说还是比较困难的，在第一次入手时基本毫无思绪，通过向同学询问和上网查找一些资料我已经基本了解了算法的原理，并成功通过JavaScript实现了算法。但是其中仍有不足，由于当初偏离了课题的意思，只考虑了安全性算法而未考虑剩余部分，导致后续添加算法的时候出现了很大的混乱。未能继续完成后续的算法，这是我下次实验应当注意的地方。