

实验报告三

题目：下次适应(Next Fit)存储分配算法

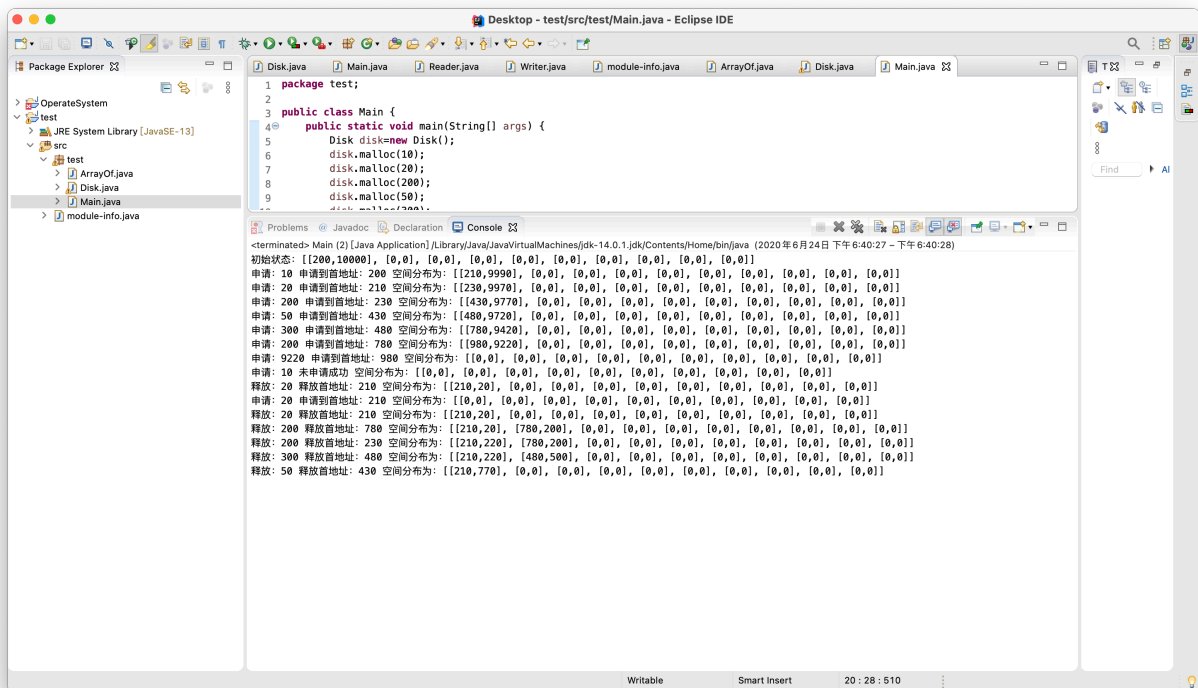
目的：理解各种存储资源分配算法，并用C语言或Java语言实现下次适应算法。

要求：设计存储资源数据结构arrayof(m_size,m_addr)，编写两个函数：(1) malloc(int size)，申请一个长度为size的空闲存储区，返回区域起始地址，不能满足时返回0；(2) mfree(int size, int aa)，size为释放区域大小，aa为起始地址。

说明：(1) 申请时有两种情况：按下次适应找到的空闲区域大小与size相同；空闲区域大于size。(2) 释放时有四种情况：释放区域与前面空闲区相邻，与后面空闲区相邻，与前后空闲区均相邻，与前后空闲区均不相邻。

提示：需要设一个静态变量static p, 指向下次分配时的起始查找位置，注意表的首尾相连。

结果：



代码

```
package test;

public class Main {
    public static void main(String[] args) {
        Disk disk=new Disk();
        disk.malloc(10);
        disk.malloc(20);
        disk.malloc(200);
        disk.malloc(50);
        disk.malloc(300);
        disk.malloc(200);
        disk.malloc(9220);
        disk.malloc(10);
        disk.mfree(20,210);
        disk.malloc(20);
        disk.mfree(20,210);
        disk.mfree(200,780);
        disk.mfree(200,230);
        disk.mfree(300,480);
        disk.mfree(50,430);
    }
}

public class ArrayOf {
    public int m_addr;
    public int m_size;

    @Override
    public String toString() {
        return "[" +
            + m_addr +
            "," + m_size +
            ']';
    }
}

public class Disk {

    int arraySize=10;
    private ArrayOf[] arrayOf=new ArrayOf[arraySize];
    int p;

    Disk(){
        p=0;
        for (int i=0;i<arraySize;i++)
        {
            arrayOf[i]=new ArrayOf();
        }
        arrayOf[0].m_size=10000;
        arrayOf[0].m_addr=200;
        arrayOf[1].m_size=0;
        System.out.println("初始状态: "+outArray());
    }
}
```

```

public int malloc(int size){

    int new_p=p;
    int now_p;
    int address;
    for(now_p=new_p;arrayOf[now_p].m_size!=0&&arrayOf[now_p].m_size<size;now_p++);
    if (arrayOf[now_p].m_size==0)
    {
        for(now_p=0;now_p<new_p&&arrayOf[now_p].m_size<size;now_p++);
    }
    if(arrayOf[now_p].m_size<size)
    {
        System.out.println("申请: "+size+" 未申请成功"+" 空间分布为: "+outArray());
        return 0;
    }
    else {

        address=arrayOf[now_p].m_addr;
        arrayOf[now_p].m_addr+=size;
        arrayOf[now_p].m_size-=size;
        p=now_p;
        if(arrayOf[now_p].m_size==0)
        {
            do {
                now_p++;
                arrayOf[now_p-1].m_size=arrayOf[now_p].m_size;
                arrayOf[now_p-1].m_addr=arrayOf[now_p].m_addr;
            }while (arrayOf[now_p-1].m_size!=0);
        }
        System.out.println("申请: "+size+" 申请到首地址: "+address+" 空间分布为: "+outArray());
        return address;
    }
}

```

