

电子科技大学

计算机专业类课程

实验报告

课程名称：计算机操作系统
学 院：计算机科学与工程学院
专 业：计算机科学与技术
学生姓名：吴思扬
学 号：2013060105016
指导教师：薛瑞尼

日 期： 2016 年 6 月 4 日

电子科技大学 实 验 报 告 实验二

学生姓名：吴思扬

学 号：2013060105016

一、实验项目名称： 银行家算法程序

二、实验原理：

银行家算法程序

* 输入

* `p`：进程数量

* `r`：资源数量

* 各进程的 `max`, `allocation`

* 输出

* 若产生死锁，打印提示：`死锁状态`。

* 否则，给出一种调度顺序。

三、实验设计

```
#include<iostream>
```

```
#include<stdlib.h>
```

```
using namespace std;
```

```
int *Available;//可用资源向量
```

```

int **Max;//最大需求矩阵

int **Allocation;//分配矩阵

int **Need;//需求矩阵

bool *Finish;

int *Work;

void Safe(const int p, const int r)
{
    int * P = (int *)malloc(sizeof(int)*p);
    int p_num = 0;
    bool has_n_w = true;
    while (has_n_w)
    {
        has_n_w = false;
        for (int i = 0; i < p; i++)
        {
            bool n_w = true;
            if (*(Finish + i))
            {
                for (int j = 0; j < r; j++)
                {
                    if (*(Need + i) + j > *(Work + j))
                    {
                        n_w = false;
                        break;
                    }
                }
            }
            if (n_w)
            {
                for (int j = 0; j < r; j++)
                    *(Work + j) = *(Allocation + i) + j)
+ *(Work + j);

                *(Finish + i) = true;
                *(P + p_num) = i + 1;
                p_num++;
                has_n_w = true;
            }
        }
    }
}

bool fin = true;
for (int i = 0; i < p; i++)
{
    if (*(Finish + i))
    {
        fin = false;
        break;
    }
}

```

```

    }
}
if (fin)
{
    for (int i = 0; i < p; i++)
        cout << " P" << *(P + i)<< endl;
}
else
{
    cout << "死锁" << endl;
}
}

void Init()
{
    int P, R;

    cout << "输入进程数量" << endl;
    cin >> P;
    cout << "输入资源数量" << endl;
    cin >> R;

    Available = (int *)malloc(sizeof(int)*P);
    Max = (int **)malloc(sizeof(int)*P);
    for (int i = 0; i < P; i++)
        *(Max + i) = (int *)malloc(sizeof(int)*R);
    Allocation = (int **)malloc(sizeof(int)*P);
    for (int i = 0; i < P; i++)
        *(Allocation + i) = (int *)malloc(sizeof(int)*R);
    Need = (int **)malloc(sizeof(int)*P);
    for (int i = 0; i < P; i++)
        *(Need + i) = (int *)malloc(sizeof(int)*R);
    Finish = (bool *)malloc(sizeof(int)*P);
    Work = (int *)malloc(sizeof(int)*P);

    for (int i = 0; i < R; i++)
    {
        cout << "输入第" << i + 1 << "个资源当前数量" << endl;
        cin >> *(Available + i);
    }
    for (int i = 0; i < P; i++)
    {
        for (int j = 0; j < R; j++)
        {
            cout << "输入第" << i + 1 << "进程对第" << j + 1 << "
种资源的最大需求" << endl;
            cin >> (*(Max + i) + j);
        }
    }
}

```

```

    }
    for (int i = 0; i < P; i++)
    {
        for (int j = 0; j < R; j++)
        {
            cout << "输入第" << i + 1 << "个进程已经获得的第" << j
+ 1 << "种资源数量" << endl;
            cin >> (*(Allocation + i) + j);
        }
    }

    for (int i = 0; i < P; i++)
    {
        for (int j = 0; j < R; j++)
        {
            (*(Need + i) + j) = (*(Max + i) + j) - (*(Allocation + i) + j);
        }
    }
    for (int i = 0; i < P; i++)
        *(Finish + i) = false;
    for (int i = 0; i < R; i++)
        *(Work + i) = *(Available + i);
    Safe(P, R);
}

```

```

int main()
{
    Init();
}

```

七、实验器材（设备、元器件）：

PC

八、实验步骤及数据结果分析：

输入资源分配表

```

1
请输入第1个进程已经获得的第2种资源数量:
0
请输入第1个进程已经获得的第3种资源数量:
0
请输入第2个进程已经获得的第1种资源数量:
6
请输入第2个进程已经获得的第2种资源数量:
1
请输入第2个进程已经获得的第3种资源数量:
2
请输入第3个进程已经获得的第1种资源数量:
2
请输入第3个进程已经获得的第2种资源数量:
1
请输入第3个进程已经获得的第3种资源数量:
1
请输入第4个进程已经获得的第1种资源数量:
0
请输入第4个进程已经获得的第2种资源数量:
0
请输入第4个进程已经获得的第3种资源数量:
2
P2 P3 P4 P1
请按任意键继续

```

报告评分:

指导教师签字: