实　验　报　告

课程名称　操作系统　**实验名称**　作业调度

姓　名　李世旺　**学　号**　1307402068　**专业班级**　信息与计算科学

实验日期　2016　**年**　4　**月**　10　**日**　　**成绩**

实验目的

模拟 FCFS 与 SJF 作业调度

实验原理与方案

FCFS 表示先来先服务
SJF 表示短作业优先
先对文件中的元组进行排序，然后再操作。

执行结果与分析

FCFS：平均周转时间:2705685.864
SJF：平均周转时间:296515.847
短作业优先相比先来先服务大大降低了平均周转时间。

详细代码

代码文件 Job.java

```
package fcfs;

public class Job implements Comparable<Job>
{
    static int currenttime;
    private Integer arrivingtime;
    private Integer runtime;
    private Integer alltime;
```

```
public Job(int arrivingtime, int runtime)
{
	super();
	this.arrivingtime = arrivingtime;
	this.runtime = runtime;
}
/**
 *
 * @return the arrivingtime
 */
public Integer getArrivingtime()
{
	return arrivingtime;
}
/**
 * @param arrivingtime the arrivingtime to set
 */
public void setArrivingtime(int arrivingtime)
{
	this.arrivingtime = arrivingtime;
}
/**
 * @return the runtime
 */
public Integer getRuntime()
{
	return runtime;
}
/**
 * @param runtime the runtime to set
 */
public void setRuntime(int runtime)
{
	this.runtime = runtime;
}
/**
 * @return the alltime
 */
public Integer getAlltime()
{
	return alltime;
}
/**
```

```
     * @param alltime the alltime to set
     */
    public void setAlltime(int alltime)
    {
        this.alltime = alltime;
    }
    @Override
    public int compareTo(Job arg0)
    {
        return this.getArrivingtime().compareTo(arg0.getArrivingtime());
    }
}
```

代码文件 FCFSmain.java

```
package fcfs;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class FCFSmain
{
    public static void SJF()
    {
        int numofline;
        double sum = 0;
        String strbuff;
        ArrayList<Job> jobs = new ArrayList<Job>();
        try
        {
            BufferedReader data = new BufferedReader(new InputStreamReader(new
BufferedInputStream(Job.class.getResource("data_old.txt").openStream())));
            strbuff = data.readLine();
            // 字符分割
            String[] strcol = strbuff.split(" ");
            numofline = Integer.valueOf(strcol[0]);
            for (int i = 0; i < numofline; i++)
            {
                strbuff = data.readLine();
```

```
                // 字符分割
                strcol = strbuff.split(" ");
                jobs.add(new Job(Integer.valueOf(strcol[0]), Integer.valueOf(strcol[1])));
            }
            Collections.sort(jobs);
            Job.currenttime = 0;
            System.out.println("----Short Job First");
            System.out.println("----时间轴-----作业描述---------服务时长");
            while (jobs.size() > 0)
            {
                for (int i = 0; i < jobs.size() && jobs.get(i).getArrivingtime() <= Job.currenttime;
i++)
                {
                    if (jobs.get(i).getRuntime() < jobs.get(0).getRuntime())
                    {
                        Job temp = jobs.get(i);
                        jobs.set(i, jobs.get(0));
                        jobs.set(0, temp);
                    }
                }
                Job jb = jobs.get(0);
                if (jb.getArrivingtime() >= Job.currenttime)
                {
                    Job.currenttime = jb.getArrivingtime();
                    jb.setAlltime(jb.getRuntime());
                    Job.currenttime += jb.getRuntime();
                } else
                {
                    Job.currenttime += jb.getRuntime();
                    jb.setAlltime(Job.currenttime - jb.getArrivingtime());
                }
                // print

    System.out.printf("%8d%8d,%-8d%8d\n",Job.currenttime,jb.getArrivingtime(),jb.getRuntime(),jb.
getAlltime());

                sum += jb.getAlltime();
                jobs.remove(0);
            }
            System.out.println("----时间轴-----作业描述---------服务时长");
            //System.out.printf("alltime:" + String.format("%9.3f", sum / numofline));
            System.out.println("作业数:" + numofline);
            System.out.printf("平均周转时间:%9.3f", sum / numofline);
        } catch (IOException exception1)
```

```
        {
            System.out.println("read file error");
        }
    }
    public static void FCFS()
    {
        int numofline;
        String strbuff;
        ArrayList<Job> jobs = new ArrayList<Job>();
        try
        {
            BufferedReader data = new BufferedReader(new InputStreamReader(new
BufferedInputStream(Job.class.getResource("data_old.txt").openStream())));
            strbuff = data.readLine();
            // 字符分割
            String[] strcol = strbuff.split(" ");
            numofline = Integer.valueOf(strcol[0]);
            for (int i = 0; i < numofline; i++)
            {
                strbuff = data.readLine();
                // 字符分割
                strcol = strbuff.split(" ");
                jobs.add(new Job(Integer.valueOf(strcol[0]), Integer.valueOf(strcol[1])));
            }
            Collections.sort(jobs);
            Job.currenttime = 0;
            System.out.println("----First Come First Service");
            System.out.println("----时间轴-----作业描述---------服务时长");
            for (Job jb : jobs)
            {
                if (jb.getArrivingtime() >= Job.currenttime)
                {
                    Job.currenttime = jb.getArrivingtime();
                    jb.setAlltime(jb.getRuntime());
                    Job.currenttime += jb.getRuntime();
                } else
                {
                    Job.currenttime += jb.getRuntime();
                    jb.setAlltime(Job.currenttime - jb.getArrivingtime());
                }

    System.out.printf("%8d%8d,%-8d%8d\n",Job.currenttime,jb.getArrivingtime(),jb.getRuntime(),jb.
getAlltime());
```

```
                }
                double sum = 0;
                for (Job jb : jobs)
                {
                        sum += jb.getAlltime();
                }
                System.out.println("----时间轴-----作业描述---------服务时长");
                System.out.println("作业数:" + numofline);
                System.out.printf("平均周转时间:%9.3f", sum / numofline);
        } catch (IOException exception1)
        {
                System.out.println("read file error");
        }
        /*
         * Job[] jobs = new Job[3]; jobs[0] = new Job(2,3); jobs[1] = new
         * Job(6,6); jobs[2] = new Job(10,10);
         */
    }
    public static void main(String[] args)
    {
        SJF();
    }
}
```