**《程序设计课程设计》实验报告**

**实验名称 《JobShop管理游戏系统程序设计》概要设计<v1.1>**

**班 级 11**

**组 号 36**

**姓 名 宋伊雯、张志博、赵佳祺**

# 1．用户界面设计（待定）

## 1.1 用户界面

8

机器数

产品数

5

确认

图1. Job shop游戏动画版输入界面1

工序时间

上一产品

上一工序

1

指定机器

10

下一工序

下一产品

已输入的产品加工订单信息展示

订单确认

产品编号

5

图2. Job shop游戏动画版输入界面2

确认

取消

确认保存当前输入？

已输入的产品加工订单信息展示

1

5

指定机器

10

工序时间

上一产品

上一工序

1

10

下一工序

下一产品

已输入的产品加工订单信息展示

订单确认

产品编号

5

图3. Job shop游戏动画版确认界面



46

51

34

29

19

7

0

图. 加工顺序甘特图

图形窗口中设计总产品数目和机器数目的输入框，工序所花时间的输入框、工序指定机器号的输入框，工序确认的按钮，下一产品的按钮，以及订单收齐确认按钮。界面元素大致如图1、2所示。

初始输入时，首先输入产品总数和机器总数，然后是产品1的加工要求，用户输入工序时间和机器号后，按工序确认可以产生一个工序操作，顺序产生多道工序后可以按“下一产品”按钮进入下一个产品的工序输入。依次操作直到所有产品的加工要求输入完成，按“订单确认”按钮启动程序计算。

[图片,布局说明; 若有多种界面,请逐一说明]

## 1.2 操作元素和操作效果

[操作元素图片]

[操作的上下文描述]

[操作后显示效果图片]

# 3 高层数据结构设计

(包括：重要的数据常量定义、数据变量定义，即各模块要共享的数据类型和参数设计，相当于头文件内容，加文字描述)

**3.1全局常量/变量定义**

**3.1.1 全局变量：**

**int** jobNum, machineNum; //产品数目, 机器数目

**3.1.2 结构：**

**typedef struct** job { //产品，用于输入与处理  
 **int** machine; //该工序指定机器号  
 **int** time; //该工序所花时间  
 **struct** job \*nextMachine; //下一机器  
} \*JOBPTR;  
**typedef struct** machine { //机器，用于处理和输出  
 **int** job; //当前产品  
 **int** time; //当前产品工序所花时间  
 **int** totalTime; //当前机器时间线（意义待定）  
 **struct** machine \*nextJob; //下一产品  
} \*MACHINEPTR;

示例：

#define MAXELES 99 //最大数组元素数

struct solutionOper{...};//解决方案的加工描述

...

**3.2 \*\*模块常量与变量定义**

**3.2.1 main模块变量：**

**int** makeSpan; //最大完工时间

**int** inputMode; //输入模式，1为键盘输入，2为文件输入，3为图形界面输入

**3.2.1 其他模块变量：（待定）**

# 4 系统模块划分

## 4.1 系统模块结构图

模块划分思路说明。

main.c

graph.c

io.c

schedule.c

**1. 模块名称 main.c**

模块功能简要描述：主体流程控制。

**2. 模块名称 schedule.c**

模块功能简要描述：完成对文件或键盘输入的订单数据获取，保存到公共变量中，输出结果。

**3. 模块名称 schedule.c**

模块功能简要描述：安排在每台机器上工件的加工顺序，使得总的完工时间(Makespan)最小，保存到公共变量中。

**4. 模块名称 graph.c**

模块功能简要描述：通过图形界面完成订单数据的获取，保存到公共变量中，通过图形界面输出结果。

## 4.2各模块函数说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 函数原型 | 功能 | 参数 | 返回值 |
| 1 | JOBPTR inputByKeyboard(); | inputMode==1时被main调用。  完成对文件输入的订单数据获取，保存到公共变量中。 | void | 产品指针数组 |
| 2 | JOBPTR inputByFile(); | inputMode==2时被main调用。  完成对文件输入的订单数据获取，保存到公共变量中。 | void | 产品指针数组 |
| 3 | JOBPTR inputByGraph(); | inputMode==3时被main调用。  完成对图形界面输入的订单数据获取，保存到公共变量中 | void | 产品指针数组 |
| 4 | **void** schedule(JOBPTR); | 安排在每台机器上工件的加工顺序，使得总的完工时间(Makespan)最小，保存到公共变量中。 | 产品指针数组 | void |
| 5 | **void** outputOnScreen(MACHINEPTR); | inputMode==1|| inputMode==2时被main调用。  在命令行界面输出结果。 | 机器指针数组 | void |
| 6 | **void** outputByFile(MACHINEPTR)); | 被main调用。  在文件中输出结果。 | 机器指针数组 | void |
| 7 | **void** outputBygraph(MACHINEPTR)); | inputMode==3时被main调用。  在图形窗口中用绘制的甘特图来表示操作在机器上的安排。 | 机器指针数组 | void |

## 4.3 函数调用图示及说明

main

outputOnScreen

inputByKeyboard

outputByFile

outputBygraph

schedule

inputByFile

inputByGraph

解释说明：如图中所示，main通过对3钟input的调用，完成对文件、键盘、图形界面输入的订单数据获取；main再通过对schedule的调用，完成对每台机器上工件的加工顺序和总的完工时间(Makespan)最小输出的安排、获取；main最后通过对outputOnScreen或outputByGraph的调用，完成对结果的命令行或图形化界面输出；outputOnScreen或outputByGraph通过对ouputByFile的调用，完成对结果的文件输出。

# 5 高层算法设计（待定）

**（用伪代码、NS图或者自然语言描述清楚核心算法的程序设计思路）**

说明重要的控制策略算法思路；

**教师评语：**