OS Experiment 02

实验报告

***16301124 梁博泓***

**1.数据结构**

(Process.java)

// 代表进程的 类

// (属性)

public int processNo;

// 进程编号

public int arrivalTime;

// 进程到达时间

public int cpuBurstTime;

// 进程占用CPU剩余时间

public int priority ;

// 进程优先级

public boolean isArrived;

// 进程是否到达

public boolean isWaiting;

// 进程是否等待

public boolean isExecuting;

// 进程是否运行

public double waitTime;

// 进程等待时间

public double executeTime;

// 进程运行时间

public double completionTime;

// 进程完成时间

-------------------------

// (方法)

getCompletionTime（）

// 获取进程的完成时间

(Scheduling.java)

// 代表调度的 类

// (属性)

public int processNum;

// 总计进程的数量

Process arrayProcess[]=new Process[10];

// 进程的数组，预先开辟了10个地址空间

-------------------------

// (方法)

public

void showInformationOfProcesses() ;

// 显示进程数组的信息（验证用，在核心算法中无实际意义）

public boolean isFininshed() ;

// 检查当前调度中是否所有的进程都被执行完毕

public

void isArrivedCheck(int currentTime);

// 检查当前所有进程是否到达执行队列

public

void executePerSeconds(int currentTime)

;

// 每秒运行进程的实际操作；

// 如果在当前进程的属性中是正在运行，增加其运行时间并且扣除 cpuBurstTime

// 如果当前进程正在等待，增加其等待时间

（public void）

FCFS;

RR\_quantum\_5;

SJF\_NonPreemptive;

SJF\_NonPreemptive; PriorityScheduling\_NonPreemptive;

// 与其名称对应的调度算法

（Run.java）

// 主函数入口

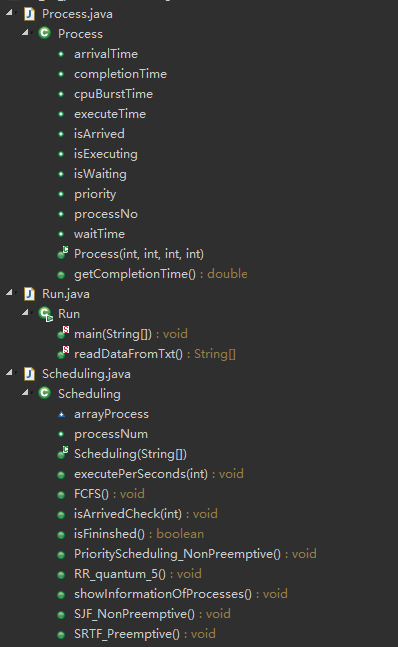
-------------------------

// (方法)

readDataFromTxt();

// 负责从文件中读取进程信息

（数据结构属性及方法一览）



**2.流程整理**

在实现所有的调度算法中，通用的是时间序列下的检查进程是否到达的函数 isArrivedCheck() 以及在每一秒的执行函数 executePerSeconds()

而在各个调度算法中，所执行的代码是为了确定当前各个进程的执行或者等待的状态的修改；为了方便说明，我将流程进行了绘图，由于板式不便于统一，将图放在了这份 [**补充文档**](调度原理.docx)之中。

*（Word下可以按住”Ctrl”键，点击上方超链接，打开对应文档）*

**3.实验代码**

（详见压缩文件中目录 ***\16301124\_梁博泓\_OS实验作业\实验2：模拟进程调度算法\实验源码\OS\_Experiment\_02\src\ex\_02\_processScheduling*** 中的下述文件）

(Tips: OS\_Experiment\_02 为eclipse单独的工程文件夹)

**Process.java**

**Scheduling.java**

**Run.java**

**4.输入数据**

**文件位置：**

*\16301124\_梁博泓\_OS实验作业\实验2：模拟进程调度算法\实验源码\OS\_Experiment\_02\****input.txt***

**内容：**

5

1 0 20 4

2 10 25 2

3 15 15 3

4 30 45 1

5 40 5 5

**5.输出结果**

(控制台输出结果)

\*\*\*\* FCFS 算法 \*\*\*\*

时刻 进程序号

----------------------

1 1

2 1

3 1

4 1

5 1

6 1

7 1

8 1

9 1

10 1

11 1

12 1

13 1

14 1

15 1

16 1

17 1

18 1

19 1

20 1

21 2

22 2

23 2

24 2

25 2

26 2

27 2

28 2

29 2

30 2

31 2

32 2

33 2

34 2

35 2

36 2

37 2

38 2

39 2

40 2

41 2

42 2

43 2

44 2

**45 2**

**46 3**

47 3

48 3

49 3

50 3

51 3

52 3

53 3

54 3

55 3

56 3

57 3

58 3

59 3

**60 3**

**61 4**

62 4

63 4

64 4

65 4

66 4

67 4

68 4

69 4

70 4

71 4

72 4

73 4

74 4

75 4

76 4

77 4

78 4

79 4

80 4

81 4

82 4

83 4

84 4

85 4

86 4

87 4

88 4

89 4

90 4

91 4

92 4

93 4

94 4

95 4

96 4

97 4

98 4

99 4

100 4

101 4

102 4

103 4

104 4

**105 4**

**106 5**

107 5

108 5

109 5

110 5

----------------------

平均等待时间 ：27.8 s

平均周转等待时间 ：49.8 s

----------------- E N D -----------------

\*\*\*\* RR\_quantum\_5 算法 \*\*\*\*

时刻 进程序号

----------------------

1 1

2 1

3 1

4 1

**5 1**

**6 1**

7 1

8 1

9 1

10 1

11 2

12 2

13 2

14 2

15 2

16 3

17 3

18 3

19 3

20 3

21 1

22 1

23 1

24 1

25 1

26 2

27 2

28 2

29 2

30 2

31 3

32 3

33 3

34 3

35 3

36 4

37 4

38 4

39 4

40 4

41 5

42 5

43 5

44 5

45 5

46 1

47 1

48 1

49 1

50 1

51 2

52 2

53 2

54 2

55 2

56 3

57 3

58 3

59 3

60 3

61 4

62 4

63 4

64 4

65 4

66 2

67 2

68 2

69 2

70 2

71 4

72 4

73 4

74 4

75 4

76 2

77 2

78 2

79 2

80 2

81 4

82 4

83 4

84 4

85 4

86 4

87 4

88 4

89 4

90 4

91 4

92 4

93 4

94 4

95 4

96 4

97 4

98 4

99 4

100 4

101 4

102 4

103 4

104 4

105 4

106 4

107 4

108 4

109 4

110 4

----------------------

平均等待时间 ：28.8 s

平均周转等待时间 ：50.8 s

----------------- E N D -----------------

\*\*\*\* SJF\_NonPreemptive 算法 \*\*\*\*

时刻 进程序号

----------------------

1 1

2 1

3 1

4 1

5 1

6 1

7 1

8 1

9 1

10 1

11 1

12 1

13 1

14 1

15 1

16 1

17 1

18 1

19 1

**20 1**

**21 3**

22 3

23 3

24 3

25 3

26 3

27 3

28 3

29 3

30 3

31 3

32 3

33 3

34 3

35 3

36 2

37 2

38 2

39 2

**40 2**

**41 2**

42 2

43 2

44 2

45 2

46 2

47 2

48 2

49 2

50 2

51 2

52 2

53 2

54 2

55 2

56 2

57 2

58 2

59 2

60 2

61 5

62 5

63 5

64 5

65 5

66 4

67 4

68 4

69 4

70 4

71 4

72 4

73 4

74 4

75 4

76 4

77 4

78 4

79 4

80 4

81 4

82 4

83 4

84 4

85 4

86 4

87 4

88 4

89 4

90 4

91 4

92 4

93 4

94 4

95 4

96 4

97 4

98 4

99 4

100 4

101 4

102 4

103 4

104 4

105 4

106 4

107 4

108 4

109 4

110 4

----------------------

平均等待时间 ：17.8 s

平均周转等待时间 ：39.8 s

----------------- E N D -----------------

\*\*\*\* SRTF\_Preemptive 算法 \*\*\*\*

时刻 进程序号

----------------------

1 1

2 1

3 1

4 1

5 1

6 1

7 1

8 1

9 1

10 1

11 1

12 1

13 1

14 1

15 1

16 1

17 1

18 1

19 1

20 1

21 3

22 3

23 3

24 3

25 3

26 3

27 3

28 3

29 3

30 3

31 3

32 3

33 3

34 3

35 3

36 2

37 2

38 2

**39 2**

**40 5**

41 5

42 5

43 5

44 5

45 2

46 2

47 2

48 2

49 2

50 2

51 2

52 2

53 2

54 2

55 2

56 2

57 2

58 2

59 2

60 2

61 2

62 2

63 2

64 2

65 2

66 4

67 4

68 4

69 4

70 4

71 4

72 4

73 4

74 4

75 4

76 4

77 4

78 4

79 4

80 4

81 4

82 4

83 4

84 4

85 4

86 4

87 4

88 4

89 4

90 4

91 4

92 4

93 4

94 4

95 4

96 4

97 4

98 4

99 4

100 4

101 4

102 4

103 4

104 4

105 4

106 4

107 4

108 4

109 4

110 4

----------------------

平均等待时间 ：14.6 s

平均周转等待时间 ：36.6 s

----------------- E N D -----------------

\*\*\*\* PriorityScheduling\_NonPreemptive 算法 \*\*\*\*

时刻 进程序号

----------------------

1 1

2 1

3 1

4 1

5 1

6 1

7 1

8 1

9 1

10 1

11 1

12 1

13 1

14 1

15 1

16 1

17 1

18 1

19 1

**20 1**

**21 3**

22 3

23 3

24 3

25 3

26 3

27 3

28 3

29 3

30 3

31 3

32 3

33 3

34 3

35 3

36 2

37 2

38 2

**39 2**

**40 2**

41 2

42 2

43 2

44 2

45 2

46 2

47 2

48 2

49 2

50 2

51 2

52 2

53 2

54 2

55 2

56 2

57 2

58 2

59 2

60 2

61 5

62 5

63 5

64 5

65 5

66 4

67 4

68 4

69 4

70 4

71 4

72 4

73 4

74 4

75 4

76 4

77 4

78 4

79 4

80 4

81 4

82 4

83 4

84 4

85 4

86 4

87 4

88 4

89 4

90 4

91 4

92 4

93 4

94 4

95 4

96 4

97 4

98 4

99 4

100 4

101 4

102 4

103 4

104 4

105 4

106 4

107 4

108 4

109 4

110 4

----------------------

平均等待时间 ：9.0 s

平均周转等待时间 ：31.0 s

----------------- E N D -----------------

**6.实验总结**

1>目前应该实现了实验要求的所有功能点，没有发现存在致命缺陷；

2>代码中有很多类似注释：

*// test*

*//System.out.println();*

无实际意义，大都是在算法的关键位置用来纠错的打印语句。

3>在类Proces中的属性isArrived ,在代码中没有被使用到；

在构思前期，本想用作表示线程的到达情况，但在实际操作过程中发现通过其他属性的组合可以替代其作用；

在工程的最后感到积重难返，没有删除这一无用变量，抱歉。