2023-1-OS01-P01 결과 보고서

설계/구현 내용 설명

1. 입력 파일

   입력 파일의 첫 번째 줄은 process의 개수를 나타낸다. 이후, 각각의 process는 하나의 줄에 process id, arrival time, burst time에 대한 정보를 나타낸다. 따라서 하나의 입력 파일은 (process의 개수 + 1)개의 줄로 이루어져 있다.

2. Process list 구조

   입력 파일에 저장되어 있는 process data를 활용하여 2차원 배열의 process list를 정의한다. Index 0, 1, 2는 입력 파일로부터 받은 정보를 그대로 활용하여 입력하고, index 3의 burst time은 index 2의 정보를 그대로 복제하여 입력한다. Index 4, 5, 6의 값은 각 process의 실행 cycle이 완료된 후에 입력 가능한 정보이다. 따라서, 초기에 process list를 구축할 때에는 0을 입력하고, 이후 cycle이 실행됨에 따라 새롭게 update한다.

Index	0	1	2	3	4	5	6
변수	Process id	Arrival time	Remain burst time	Burst time	End time	Turnaround time	Waiting time

3. 변수 설정

   A. process_list: 전체 process에 대한 정보를 담은 2차원 list

   B. executedplist: 실행한 process cycle에 대한 정보를 담은 2차원 list

   C. stime: start time으로, process가 각 cycle의 CPU burst를 시작한 시점

   D. etime: end time으로, process가 각 cycle의 CPU burst를 마친 시점

   E. ttime: turnaround time으로, 전체 CPU burst time을 완료한 시점의 end time – arrival time 값

   F. wtime: waiting time으로, turnaround time – burst time

4. FCFS scheduling 구현

생성자 함수, scheduling 함수, executedplist 함수로 구성된 FCFS class를 구현하였다.

A. 생성자 함수

FCFS scheduling의 대상이 되는 process list를 입력 받는다. cycle을 실행한 process를 저장하기 위한 빈 list(executedplist)를 정의한다.

B. Scheduling 함수

- FCFS scheduling 기법의 start time을 찾는다. 방법은 다음과 같다.

  1. Start time을 찾기 위한 빈 list(findstime)를 정의한다.
  2. 입력받은 process list 중 end time이 가장 큰 process가 가장 최근에 실행된 process이므로 해당 process의 end time이 곧 다음에 실행될 process의 start time이 된다. 따라서 입력받은 process list에서 각 process들의 end time을 findstime에 넣은 후, 최댓값을 찾아 start time(stime)에 할당한다.

- FCFS scheduling을 실행하고, 실행 결과를 executedplist와 process list에 저장한다. 남은 burst time이 0인 경우, pass한다. 남은 burst time이 0이 아닌 경우, FCFS scheduling을 진행한다.

  1. Start time이 해당 process의 arrival time보다 작은 경우, start time에 arrival time을 할당한다.
  2. 실행을 시작하는 process들에 대해 process id, start time, 해당 실행 cycle의 burst time 정보를 temp 변수에 list 형태로 저장한다. 한 cycle의 실행 결과를 executedplist에 저장하기 위함이다.
  3. 실행이 완료된 process들에 대해 start time에 남은 burst time을 더하고 해당 값을 end time에 할당한다. Process list에 대하여 해당 process의 남은 burst time을 0으로 만들고, 계산한 end time 값을 입력한다. End time에서 arrival time을 뺀 값을 turnaround time에 할당하고, process list에 입력한다. Turnaround time에서 burst time을 뺀 값을 waiting time에 할당하고, process list에 입력한다.
  4. Temp에 end time에 대한 정보를 추가로 저장하고, temp를 executedplist에 추가한다.

C. Executedplist 반환 함수

Class 내에서 정의된 executedplist를 반환하는 함수를 생성한다.

5. RoundRobin scheduling 구현

생성자 함수, scheduling 함수, executedplist로 구성된 Round Robin class를 구현하였다.

A. 생성자 함수

Round Robin scheduling의 대상이 되는 process list를 입력 받는다. Timequantum을 입력 받는다. 실행중인 process를 저장하기 위한 빈 list(executedplist)를 정의한다.

B. Scheduling 함수

- Round Robin scheduling 기법의 Start time을 찾는다. 방법은 다음과 같다.

  1. Start time을 찾기 위한 빈 list(findstime)를 정의한다.
  2. 입력받은 process list 중 end time이 가장 큰 process가 가장 최근에 실행된 process이므로 해당 process의 end time이 곧 다음에 실행될 process의 start time이 된다. 따라서 process list의 end time을 findstime에 넣은 후, 최댓값을 찾아 start time(stime)에 할당한다.

- Round Robin scheduling을 실행하고, 실행 결과를 process list에 저장한다. 남은 burst time이 0인 경우, pass한다. 남은 burst time이 0이 아닌 경우, Round Robin scheduling을 진행한다.

  1. Start time이 해당 process의 arrival time보다 작은 경우, start time에 arrival time을 할당한다.
  2. 남은 burst time이 timequantum보다 큰 경우, 실행을 시작하는 process들에 대해 process id, start time, 해당 실행 cycle의 burst time(=timequantum) 정보를 temp 변수에 list 형태로 저장한다. 한 cycle의 실행 결과를 executedplist에 저장하기 위함이다. 실행이 완료된 process들에 대해 start time에 남은 timequantum을 더하고 해당 값을 end time에 할당한다. Process list에 대하여 해당 process의 남은 burst time에서 timequantum을 뺀 값을 burst time으로 새롭게 할당하고, 계산한 end time 값을 입력한다. 전체 burst time이 다 실행된 것이 아니므로, turnaround time과 waiting time은 계산하지 않는다. Temp에 해당 cycle의 end time에 대한 정보를 추가로 저장하고, temp를 executedplist에 추가한다.

3. 남은 burst time이 timequantum보다 작은 경우, 실행을 시작하는 process들에 대해 process id, start time, 해당 실행 cycle의 burst time(=남은 burst time) 정보를 temp 변수에 list 형태로 저장한다. 한 cycle의 실행 결과를 executedplist에 저장하기 위함이다. 실행이 완료된 process들에 대해 start time에 남은 burst time을 더하고 해당 값을 end time에 할당한다. Process list에 대하여 해당 process의 남은 burst time을 0으로 만들고, 계산한 end time 값을 입력한다. End time에서 arrival time을 뺀 값을 turnaround time에 할당하고, process list에 입력한다. Turnaround time에서 burst time을 뺀 값을 waiting time에 할당하고, process list에 입력한다. Temp에 해당 cycle의 end time에 대한 정보를 추가로 저장하고, temp를 executedplist에 추가한다.

C. Executedplist 반환 함수

Class 내에서 정의된 executedplist를 반환하는 함수를 생성한다.

6. MFQ scheduling 구현

MFQ scheduling 과정을 사용자 정의 함수로 구현하였다.

Q0에 time quantum이 2인 Round Robin scheduling 기법을 할당하고, scheduling을 실시한다. 실행 과정을 담은 executedplist를 반환하고, Gantt Chart 출력을 위해 ganttchart list에 저장한다. Q1에 time quantum이 4인 Ronud Robin scheduling 기법을 할당하고, scheduling을 실시한다. 실행 과정을 담은 executedplist를 반환하고, Gantt Chart 출력을 위해 ganttchart list에 저장한다. Q2에 FCFS 기법을 할당하고, scheduling을 실시한다. 실행 과정을 담은 executedplist를 반환하고, Gantt Chart 출력을 위해 ganttchart list에 저장한다. 모든 process는 최초에 Q0로 진입하며, Qi에서 scheduling을 받아 실행하고 queue의 time quantum을 소모할 경우, Qi+1로 진입한다는 조건에 따라, Q0을 실행한 후, 이어서 Q1을 실행하고, 이어서 Q2를 실행한다. Q0의 우선순위가 가장 높고, Q2의 우선순위가 가장 낮으며, scheduling은 항상 높은 우선순위의 queue에서부터 이루어진다는 조건에 따라 Q0부터 Q1, Q2를 차례대로 실행한다. 이후, 최종 실행 결과에 대한 정보인 process list와 실행 과정에 대한 정보인 ganttchart를 return값으로 설정한다.

7. Main 함수 구현

A. 입력 파일 불러오기

입력 파일을 불러와 process list에 process에 대한 정보(process id, arrival time,

remain burst time)를 차례대로 입력한다. Burst time에 remain burst time(burst time의 초기값)을 할당한다. 아직 실행되지 않은 process들이므로, end time, turnaround time, waiting time은 0으로 설정한다. 각각의 process에 대한 위 정보를 리스트로 표현하고, 각 process들을 process list에 추가하여 2차원 리스트를 구성한다.

B. MFQ scheduling

MFQ scheduling 함수를 실행하여 table에 process list를, chart에 executedplist를 저장한다.

C. 결과 출력

- Process table 및 average turnaround time, average waiting time 출력

Table에 저장된 process들을 process id를 기준으로 정렬한다. Process id 순서대로 process id, arrival time, burst time, turnaround time, waiting time을 table 형태로 출력한다. 이후, average turnaround time과 average waiting time을 계산하여 출력한다.

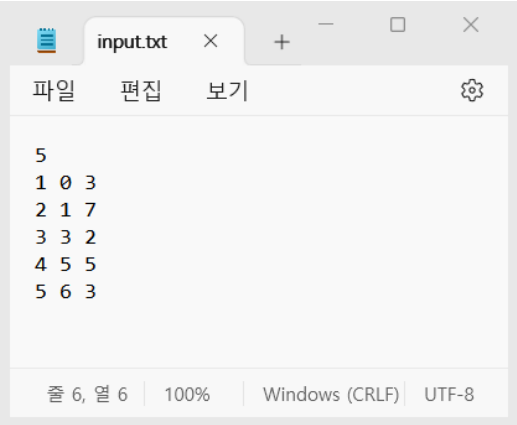
- Gantt Chart 출력

Chart에 저장된 실행 cycle에 대한 정보를 입력된 순서대로 출력한다. (list 자료형이므로 별도의 정렬 과정을 거치지 않는다.) cycle의 start time과 end time을 출력하고, cycle의 burst time 만큼 해당 cycle의 process id를 출력한다.

다양한 입력에 대한 실행 결과

1. Input1.txt

[input 입력 파일]



[output process table / average turnaround time / average waiting time]

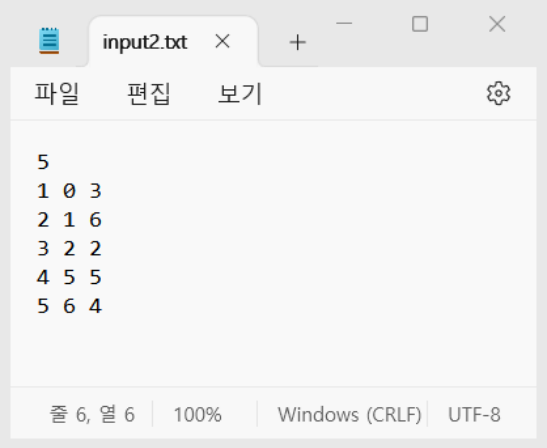
```
Process Table
-----------------------------------------------------------------------
| PID | Arrival Time | Burst Time | Turnaround Time | Waiting Time |
|  1  |      0       |     3      |       11        |      8       |
|  2  |      1       |     7      |       19        |      12      |
|  3  |      3       |     2      |        3        |      1       |
|  4  |      5       |     5      |       13        |      8       |
|  5  |      6       |     3      |       13        |      10      |
-----------------------------------------------------------------------
Average Turnaround Time:  11.8
Average Waiting Time:  7.8
```

[Gantt chart]

```
Gantt Chart
0--------------2
|       1      |
|       1      |
2--------------4
|       2      |
|       2      |
4--------------6
|       3      |
|       3      |
6--------------8
|       4      |
|       4      |
8--------------10
|       5      |
|       5      |
10-------------11
|       1      |
11-------------15
|       2      |
|       2      |
|       2      |
|       2      |
15-------------18
|       4      |
|       4      |
|       4      |
18-------------19
|       5      |
19-------------20
|       2      |
-----------------
```

2. Input2.txt

[input 입력 파일]



[output process table / average turnaround time / average waiting time]

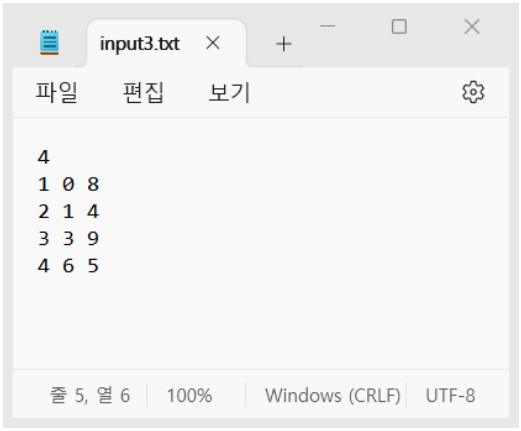
```
Process Table
-------------------------------------------------------------------
| PID | Arrival Time | Burst Time | Turnaround Time | Waiting Time |
|  1  |      0       |     3      |       11        |      8       |
|  2  |      1       |     6      |       14        |      8       |
|  3  |      2       |     2      |        4        |      2       |
|  4  |      5       |     5      |       13        |      8       |
|  5  |      6       |     4      |       14        |      10      |
-------------------------------------------------------------------
Average Turnaround Time:  11.2
Average Waiting Time:  7.2
```

[Gantt chart]

```
Gantt Chart
0--------------2
|       1      |
|       1      |
2--------------4
|       2      |
|       2      |
4--------------6
|       3      |
|       3      |
6--------------8
|       4      |
|       4      |
8--------------10
|       5      |
|       5      |
10-------------11
|       1      |
11-------------15
|       2      |
|       2      |
|       2      |
|       2      |
15-------------18
|       4      |
|       4      |
|       4      |
18-------------20
|       5      |
|       5      |
-----------------
```

3. Input3.txt

[input 입력 파일]



[output process table / average turnaround time / average waiting time]

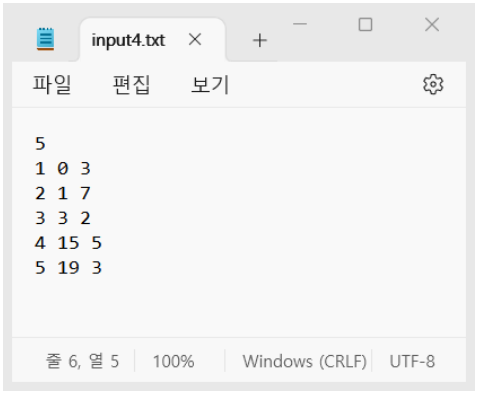
```
Process Table
-------------------------------------------------------------------------
| PID | Arrival Time | Burst Time | Turnaround Time | Waiting Time |
|  1  |      0       |     8      |       23        |      15      |
|  2  |      1       |     4      |       13        |      9       |
|  3  |      3       |     9      |       23        |      14      |
|  4  |      6       |     5      |       15        |      10      |
-------------------------------------------------------------------------
Average Turnaround Time:  18.5
Average Waiting Time:  12.0
```

[Gantt chart]

```
Gantt Chart
0--------------2
|      1       |
|      1       |
2--------------4
|      2       |
|      2       |
4--------------6
|      3       |
|      3       |
6--------------8
|      4       |
|      4       |
8--------------12
|      1       |
|      1       |
|      1       |
|      1       |
12-------------14
|      2       |
|      2       |
14-------------18
|      3       |
|      3       |
|      3       |
|      3       |
18-------------21
|      4       |
|      4       |
|      4       |
21-------------23
|      1       |
|      1       |
23-------------26
|      3       |
|      3       |
|      3       |
----------------
```

4. Input4.txt

[input 입력 파일]



[output process table / average turnaround time / average waiting time]

```
Process Table
-------------------------------------------------------------------------
| PID | Arrival Time | Burst Time | Turnaround Time | Waiting Time |
|  1  |      0       |     3      |       22        |      19      |
|  2  |      1       |     7      |       30        |      23      |
|  3  |      3       |     2      |        3        |      1       |
|  4  |      15      |     5      |       14        |      9       |
|  5  |      19      |     3      |       11        |      8       |
-------------------------------------------------------------------------
Average Turnaround Time:  16.0
Average Waiting Time:  12.0
```

[Gantt Chart]

```
Gantt Chart
0--------------2
|       1       |
|       1       |
2--------------4
|       2       |
|       2       |
4--------------6
|       3       |
|       3       |
15-------------17
|       4       |
|       4       |
19-------------21
|       5       |
|       5       |
21-------------22
|       1       |
22-------------26
|       2       |
|       2       |
|       2       |
|       2       |
26-------------29
|       4       |
|       4       |
|       4       |
29-------------30
|       5       |
30-------------31
|       2       |
-----------------
```

실행 결과를 보는 방법에 대한 설명

1. Process table, average turnaround time, average waiting time

```
Process Table
-----------------------------------------------------------------
| PID | Arrival Time | Burst Time | Turnaround Time | Waiting Time |
|  1  |      0       |     3      |       11        |      8       |
|  2  |      1       |     7      |       19        |      12      |
|  3  |      3       |     2      |        3        |      1       |
|  4  |      5       |     5      |       13        |      8       |
|  5  |      6       |     3      |       13        |      10      |
-----------------------------------------------------------------
Average Turnaround Time:  11.8
Average Waiting Time:  7.8
```

PID: process를 구분하기 위한 process id이다.

Arrival Time: process의 도착시간이다.

Burst Time: process의 수행시간이다.

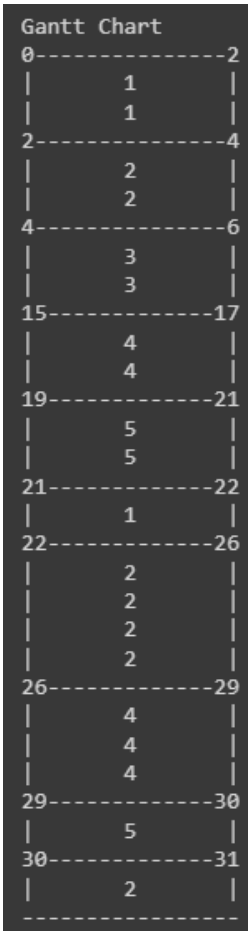
Turnaround Time: process가 도착해서 CPU를 빠져나가기까지 소요된 시간이다.

Waiting Time: process가 CPU에 도착한 후, 빠져나가기까지 소요된 시간 중, CPU를 할당받지 않고 대기한 시간이다.

Average Turnaround Time: 전체 process의 평균 turnaround time으로, 각 process의 turnaround time의 합을 전체 process의 개수로 나눠서 구한다.

Average Waiting Time: 전체 process의 평균 waiting time으로, 각 process의 waiting time의 합을 전체 process의 개수로 나눠서 구한다.

2. Gantt chart



각 process가 CPU를 할당받아 사용한 시간을 chart로 나타낸 것이다. 한 cycle의 process id 및 start time, end time과 한 cycle의 소요시간에 대한 정보를 포함한다. 위 사진의 경우, time 0에 process1이 실행되어 2만큼의 time동안 실행되어 time 2에서 종료되었다. (process id인 1이 두 번 출력되었음을 통해 알 수 있음.) 이어서 time2에 process2가 실행되었으며, 2만큼의 time동안 실행되어 time 4에서 종료되었다. (process id인 2가 두 번 출력되었음을 통해 알 수 있음.) process3는 time 4부터 time6까지 2만큼의 time동안 실행되었다. 이어, process4가 time 15부터 time 17까지 2만큼의 time동안 실행되었다. Process3의 end time은 6인데, process4의 start time이 15인 것을 통해 time 6부터 time 15까지 CPU에 할당된 process가 없었음을 알 수 있다.