

< 구현 결과 >

1. Periodic task 5개를 computation time period 순으로 입력한다.
2. Aperiodic task 3개를 computation time arrival time 순으로 입력한다.
3. Polling server task를 capacity period 순으로 입력한다.

< 실행화면 >

- ## 1. 입력화면

```
C:\WINDOWS\system32\cmd.exe

**Realtime Scheduling **
First, Input 5 periodic tasks ( Computation time / Period ).
cf. If P0 has Computation time is 1, Period 5 then you enter like 1 5.
1 5
1 10
1 15
3 30
5 60

Second, Input 3 Aperiodic tasks (Computation time / Arrival Time).
cf. If APO has Computation time is 1, Arrival Time is 7 then you enter like 1 7.
2 7
1 10
1 29

Last, Input 1 Polling server information
cf. If the Polling server has Capacity is 1, Period 5 then you enter like 1 5.
1 5
```

2. 입력이 완료된 후, 출력되는 화면

(요약) Background APS, Polling APS 각각 평균 지연 시간을 계산하여 출력한 뒤, Gantt chart를 함께 출력합니다.

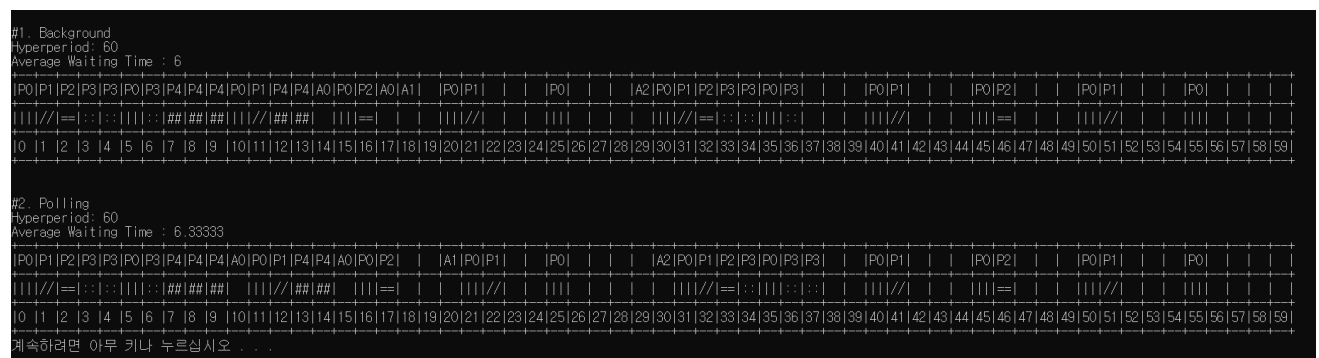
Gantt chart를 출력할 때에는 1행에는 Periodic task의 경우 P0, P1.. 등으로 나타내었고, Aperiodic task의 경우에는 A0, A1 .. 등으로 나타내었습니다.

알파벳 뒤의 숫자는 우선순위가 높을수록 작은 수를 가지도록 설정했습니다.

2행에는 각 태스크별로 그래픽화하여 보일 수 있도록 설정하였고, 3행은 시간을 나타내도록 설정하였습니다.

즉, 3열까지 해석하자면 P0가 0~1초, P10| 1~2초, P2가 2~3초 동안 실행된 것을 확인할 수 있습니다.

**** 가로로 길게 출력되도록 하였으므로 전체 창으로 해야 완전한 이미지를 확인할 수 있습니다. ****



3. 잘못된 입력이 들어왔을 시 (태스크가 마감시간을 지키지 못할 때)

```

**Realtime Scheduling **
First, Input 5 periodic tasks ( Computation time / Period ).
cf. If P0 has Computation time is 1, Period 5 then you enter like 1 5.
1 10
8 10
2 15
1 30
1 60

Second, Input 3 Aperiodic tasks (Computation time / Arrival Time).
cf. If AP0 has Computation time is 1, Arrival Time is 7 then you enter like 1 7.
2 10
1 4
1 6

Last, Input 1 Polling server information
cf. If the Polling server has Capacity is 1, Period 5 then you enter like 1 5.
1 5

#1. Background
Hyperperiod: 60
The task does not end within the deadline.

#2. Polling
Hyperperiod: 60
The task does not end within the deadline.계속하려면 아무 키나 누르십시오 . . .

```

마감시간을 지키지 못하면, The task does not end within the deadline 이라는 문구를 출력하고

더 이상 실행하지 않도록 설정했습니다.