# 2주차 실습 보고서

## 실습 주제

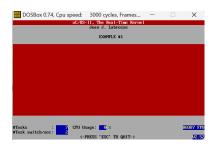
- 4개의 태스크 생성
- 각 태스크에 우선순위 0,1,2,3 할당
- 각 태스크는 화면 전체를 Red, Blue, Brown, Green으로 색칠함

#### 성공 실패 여부

- 1. 첫 번째 시도 (실패)
  - 각 태스크 수행 후 sleep 하지 않음
- 2. 두 번째 시도 (성공)
  - 각 태스크 sleep

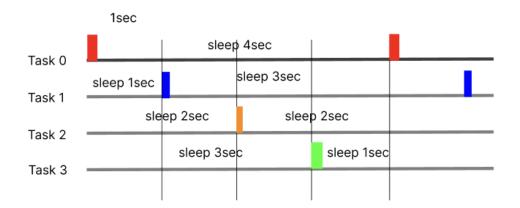
## 원인 분석

1. 첫 번째 시도 실패 원인 분석



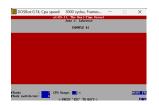
- RTOS는 priority에 따른 preemption을 지원하기 때문에, 생성한 4개의 task 0,1,2,3 중 우선순위가 가장 높은 0번 task인 RED로 칠하는 동작이 항상 수행되게 된다.
- 2. 두 번째 시도 성공 원인 분석

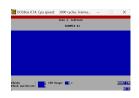
2주차 실습 보고서

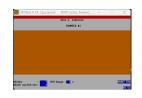


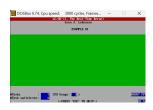
- x=0 ~ 80, y=5 ~20 반복문을 돌면서 색을 칠하는 시간은 1초 중에 정말 빠르게 수행되고, 1초 동안 화면에 색을 출력하기 위해서는 나머지 Task들은 1초 이상 sleep되어 있어야 한다.
- 또한, 다음 Task가 수행되기 위해서 색을 칠한 (우선순위가 높은)Task는 자신의 차례가 올 때까지 sleep되어야 한다.
- 색을 칠한 1초 이후, 다음 Task가 sleep이 끝나 스케쥴링되고 다음 색을 빠르게 칠한 뒤 이전의 동작을 반복한다.
- OSTimeDlyHMSM(0, 0, 1, 0);

#### 결과









#### 5. 결과 분석

- 의도한 대로 4개의 Task가 1초간 번갈아 가며 Red, Blue, Brown, green화면을 보여 줌.
- RTOS는 <mark>우선순위 기반 선점형 스케쥴링</mark>을 지원하기 때문에 우선순위가 낮은 Task를 수행하기 위해서는 우선순위가 높은 Task를 block 해야함.
- 따라서 4개의 Task를 번갈아 수행하기 위해서는 각 Task의 우선순위를 고려하여 적절 히 block하는게 중요함.

2주차 실습 보고서