Embedded System

5차 실습 과제

컴퓨터공학과

12161558

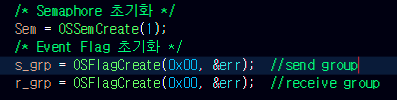
김혜윤

* 목적 : Semaphore와 Event flag 활용법
* 실습내용

1. 5개의 task 생성
   1. 1개의 Decision task
   2. 4개의 Random task
2. 4개의 Random Task에서 0부터 63사이의 랜덤숫자를 생성
3. 4개의 Random Task는 해당 숫자를 배열 변수를 저장
   1. OSSemPend과 OSSemPost 이용
   2. OSFlagPost와 OSFlagPend 함수 활용
4. Decision task는 4개의 task중에서 가장 작은 랜덤 숫자를 보낸 task를 선택해서 해당 배열 변수에 ‘W’ 문자를 저장하고, 나머지에게는 ‘L’ 문자를 저장한다.
   1. OSSemPend, OSSemPost, OPFlagPost, OSFlagPend 함수 활용.
5. ‘W’ 문자를 받은 TASK의 색을 칠해준다.
   1. 1 순위 : RED
   2. 2 순위 : CYAN
   3. 3 순위 : BLUE
   4. 4 순위 : GREEN

* **CODE**





Semaphore와 event flag 선언과 초기화

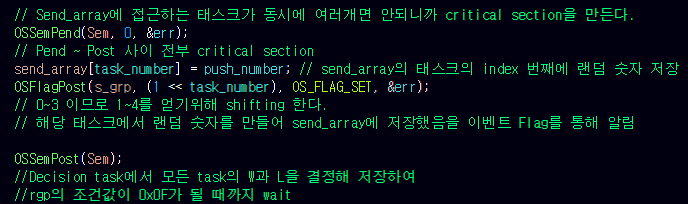




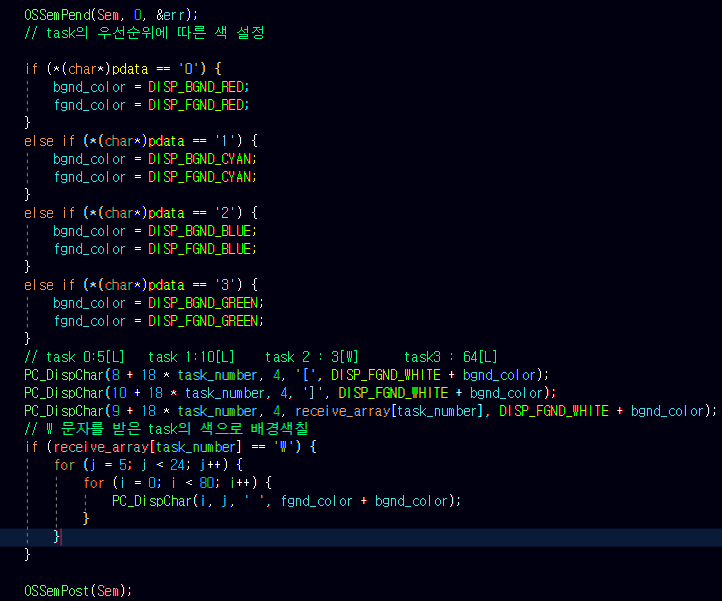
모든 random task들이 랜덤숫자를 받을 때까지 wait한다.

OSFlagPend() 함수를 이용해 기다리고, OSFlagPost() 함수를 이용해 flag를 1로 set 해준다.

여기서 1은 값이 들어왔다는 표시이고, 1111은 모든 random task들이 랜덤숫자를 입력 받았음을 알려준다.



공유 변수인 send\_array에 접근하기 때문에 semaphore를 이용해서 critical section을 만들어준다. OSSemPend() ~ OSSemPost() 사이는 전부 critical section으로 한번에 하나의 task만 접근 가능하다. Send\_array의 task의 index 번째에 랜덤 숫자 저장한다. 그 후 저장했음을 이벤트 flag를 통해 알린다.



이 부분은 if문으로 인해 한 task만 접근할 수 있지만, 공유변수인 receive\_array에 접근하기 때문에 원칙으로 인해 semaphore을 이용하여 critical section을 만들어 공유변수 보호함.

* 결과화면 캡쳐

