

실험.실습 보고서

(3) 반

조원: (학번:12131489,이름:김영훈)

1. 제목

5주차 실습과제 - mailbox 및 message queue 사용하기

2. 목적

mailbox 및 message queue를 활용하는 방법을 익히고, 이를 통해 task 간 데이터를 주고 받는 프로그램을 만든다.

3. 실습에 필요한 기초지식

mailbox, message queue에 대한 개념을 이해해야 한다. 또한 이를 활용하기 위하여 post, pend 등 함수들의 기능에 대해 알아야 하고 내부 구현에 대해 이해해야 어떻게 동작하는지 파악할 수 있다.

4. 실습 절차, 내용 및 결과

Mailbox

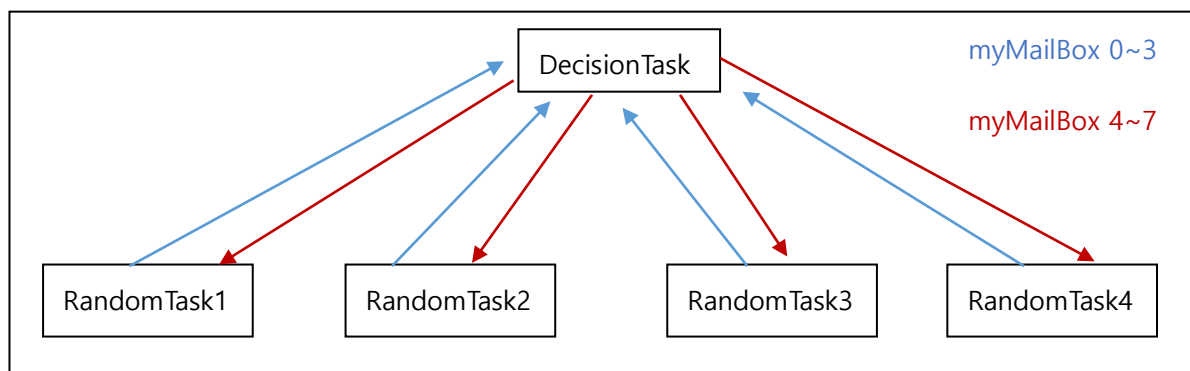
TaskStartCreateTasks 함수에서 8개의 Mailbox를 생성한다. 그리고 task를 DecisionTask 1개, RandomTask 4개를 생성한다. DecisionTask는 RandomTask로부터 랜덤한 숫자를 받아, 가장 작은 숫자를 준 task에게 'W' 전달, 나머지 task에는 'L' 전달하는 task이다.

RandomTask는 랜덤한 값을 생성하여 출력하고, DecisionTask에게 보낸 뒤 DecisionTask에게 다시 문자를 받아 문자가 'W'인 task의 색깔로 화면을 칠한다.

RandomTask -> 각 task가 어떻게 동작하는지 자세히 보면 RandomTask는 랜덤한 값을 생성하여 출력하고, OSMboxPost 함수를 이용하여 Mailbox(0~3번)에 보낸다. 이후, RandomTask는 Mailbox(4~7번)의 값을 받기위해 OSMboxPend를 호출하고 block 상태에 빠진다.

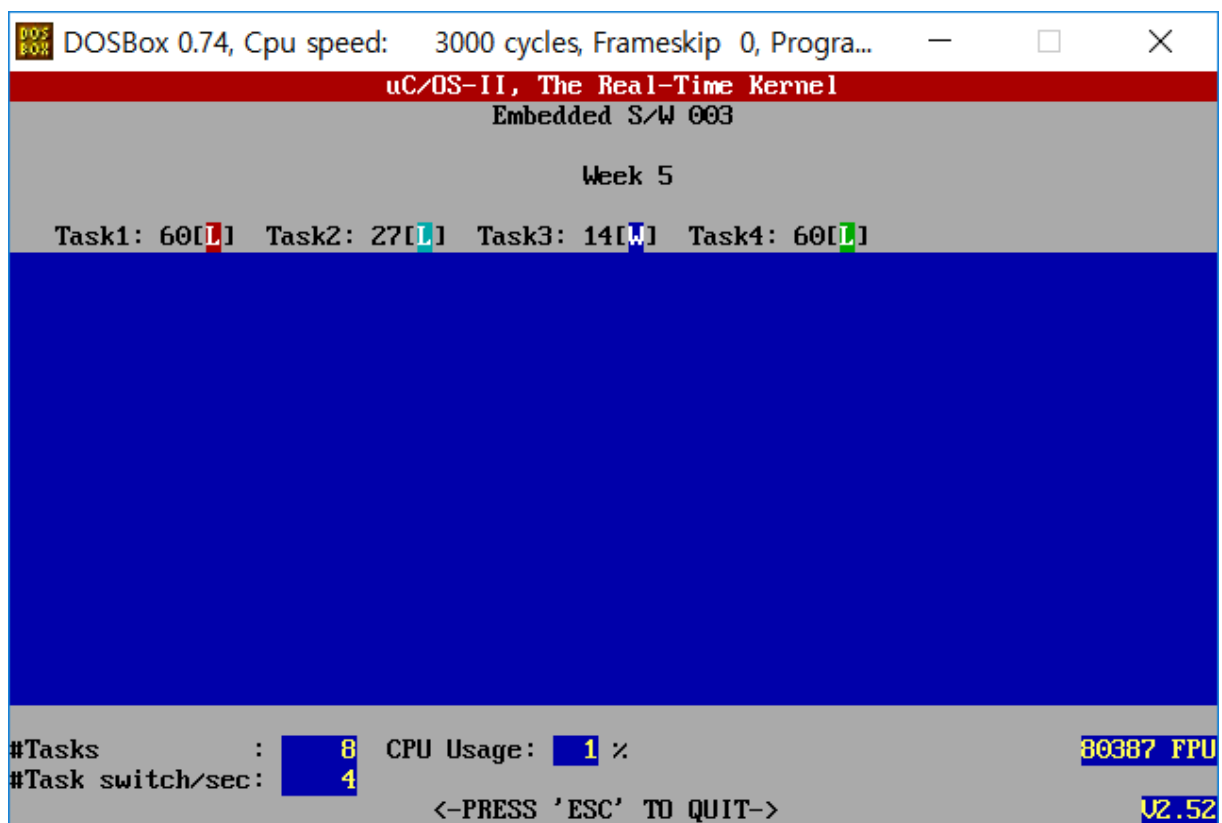
DecisionTask -> OSMboxPend 함수로 각 Mailbox(0~3번)에 message가 오기를 기다리고 있었던 DecisionTask는 message를 받게 된다. 각 task로부터 받은 message를 task_value 배열에 저장하고, 모든 message가 도착하면 그 중 최소값을 찾는다. 그리고 최소값과 task_value의 값을 비교하여 가장 작은 값을 보낸 task에게 'W'를 보내고 나머지는 'L'을 보낸다(OSMboxPost를 이용하여 Mailbox(4~7번)에 보내는 것이다) 그리고 최소값을 찾기 위해 사용된 myRdlTbl 배열을 초기화하고, 다시 처음으로 돌아가 OSMboxPend 함수로 block 상태에 빠진다.

RandomTask -> DecisionTask의 Pend를 통해 RandomTask는 block에서 빠져나와 message를 받는다. message 값을 alphabet 변수에 저장된다. 각 alphabet 값을 task에 맞는 화면 위치에 출력한다. alphabet의 값이 'W'일 경우 화면의 색상을 지정된 해당 task의 색상으로 바꿔준다. 이후 5초 동안 delay를 주고 처음으로 돌아간다.



Message queue

Message queue 또한 Mailbox와 같은 원리로 동작하였다. 단, Mailbox와 달리 여러 message를 받을 수 있는 공간을 배열로 할당하여 하나의 Message queue에 여러 message를 담을 수 있는 차이점이 있었다.



Mailbox 실행화면

```
uC/OS-II, The Real-Time Kernel
Embedded S/W 003

Week 5

Task1: 20[L] Task2: 02[L] Task3: 19[L] Task4: 40[L]

#Tasks      : 8 CPU Usage: 3 % 80387 FPU
#Task switch/sec: 20
<-PRESS 'ESC' TO QUIT-> U2.52
```

Message queue 실행화면

5. 결론

이번 실습에서는 Mailbox, Message queue를 통해서 task 간 데이터를 주고 받을 수 있는 프로그램을 작성하였다. 코드를 작성하면서 Post와 Pend를 통해 각 task 간 데이터를 잘 주고 받을 수 있도록 동기화하는 것이 중요하다는 것을 느꼈다. 동기화가 제대로 안 되어 있을 경우 예상치 못한 오류들이 발생했었다. Post와 Pend 함수를 DecisionTask, RendomTask에서 알맞게 사용하여 message가 순차적으로 전달될 수 있도록 하는 것이 중점이었다.

이번 실습을 통해 Mailbox와 Message queue를 사용하는 방법에 대해 익히고 여러 Task를 어떻게 동기화 시킬 수 있을지 고민해보는 시간이 되었다.