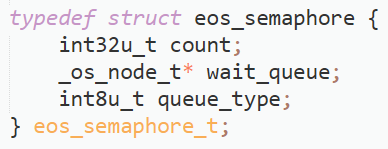
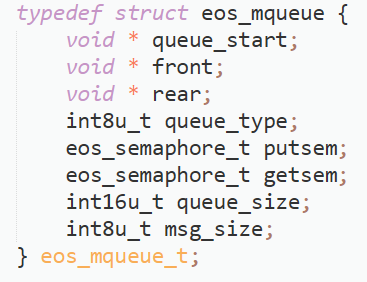
[OS 프로젝트 4] 2014-17952 정재철

[정의한 구조체]

**eos\_semaphore\_t <eos/eos.h>**

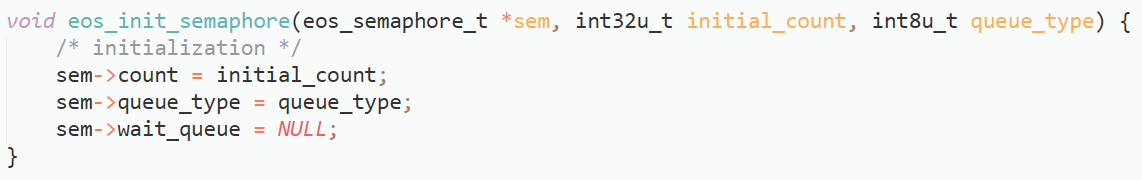
count, wait queue 포인터, queue type(FIFO or priority based)로 구성되어 있다.

**eos\_mqueue\_t <eos/eos.h>**

queue start, front, rear 포인터, queue type(FIFO or priority based), put/get semaphore, queue size, message size로 구성되어 있다.

[구현한 함수]

1. **eos\_init\_semaphore <eos/sync.c>**



세마포 구조체의 멤버 변수를 함수에 입력된 인자로 초기화한다.

1. **eos\_acquire\_semaphore <eos/sync.c>**

세마포를 건드리기 전에 항상 eos\_disable\_interrupt를 통해 인터럽트를 끈다. 인자로 주어진 세마포를 체크하는데,

* 만약 0보다 크면 세마포를 하나 줄이고 인터럽트를 다시 킨 후 1을 리턴한다.

0보다 작은 경우 인자로 주어진 timeout에 따라 세 경우로 나뉜다.

* timeout=-1인 경우 그 즉시 0을 리턴한다.

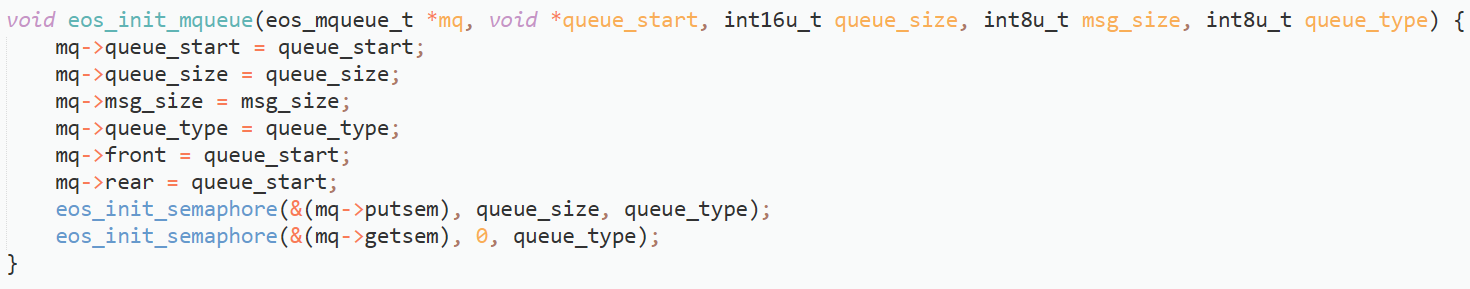
-1이 아닌 경우 세마포의 wait queue에 현재 태스크를 추가한다.

* timeout=0인 경우 eos\_release\_semaphore가 발생하기 전까지 대기한다. 이는 그냥 eos\_schedule을 수행하면 된다.
* timeout>0인 경우 eos\_set\_alarm을 이용하여 시스템 타이머의 wait queue에도 태스크를 추가한다. 이렇게 하면 알람에 의해 깨어날 수도 있고 eos\_release\_semaphore에 의해 깨어날 수도 있다. 한 번 깨어난 태스크를 또 깨우지 않기 위해서는 <eos/task.c>에 있는 \_os\_wakeup\_sleeping\_task에 tcb의 상태가 WAITING인지 확인하고 만약 아니면 그냥 리턴하도록 하는 코드를 추가하면 된다.

1. eos\_release\_semaphore <eos/sync.c>

세마포 구조체에 접근하기 전 eos\_disable\_interrupt로 인터럽트를 비활성화한다. 인자로 주어진 세마포에 1을 더하고, 만약 세마포가 1 이상이고 wait queue에 노드가 존재한다면 큐에서 제거한 후 \_os\_wakeup\_sleeping\_task로 깨운다.

1. **eos\_init\_mqueue <eos/comm.c>**



메시지 큐의 멤버 변수들을 주어진 인자로 초기화한다. putsem과 getsem을 eos\_init\_semaphore로 초기화한다.

1. **eos\_send\_message <eos/comm.c>**

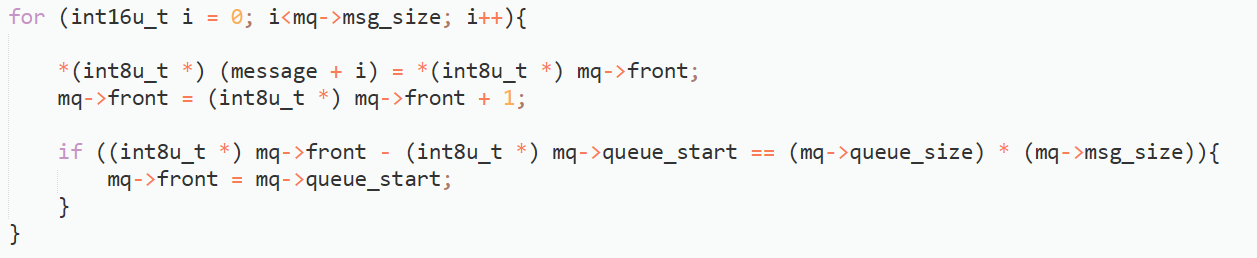
eos\_acquire\_semaphore로 putsem이 1 이상일 때까지 timeout 동안 기다린다. 만약 얻지 못하면 0을 리턴한다. 얻은 경우 메시지 큐에 메시지를 입력한다. 변수가 void 포인터로 이루어져 있으나 테스트 프로그램에서 int8u\_t 변수를 사용하므로 아래와 같이 구현하였다.



eos\_release\_semaphore로 getsem에 1을 추가한다.

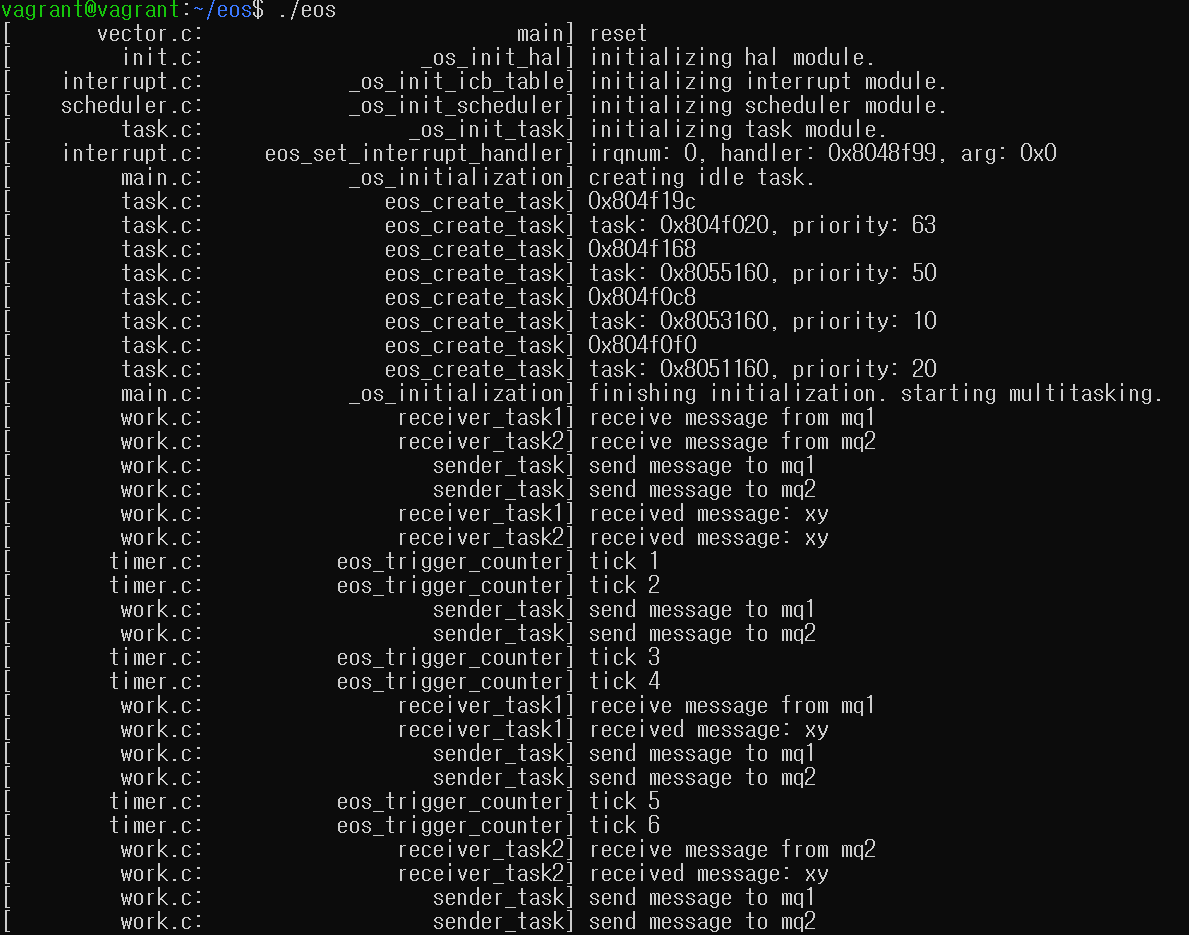
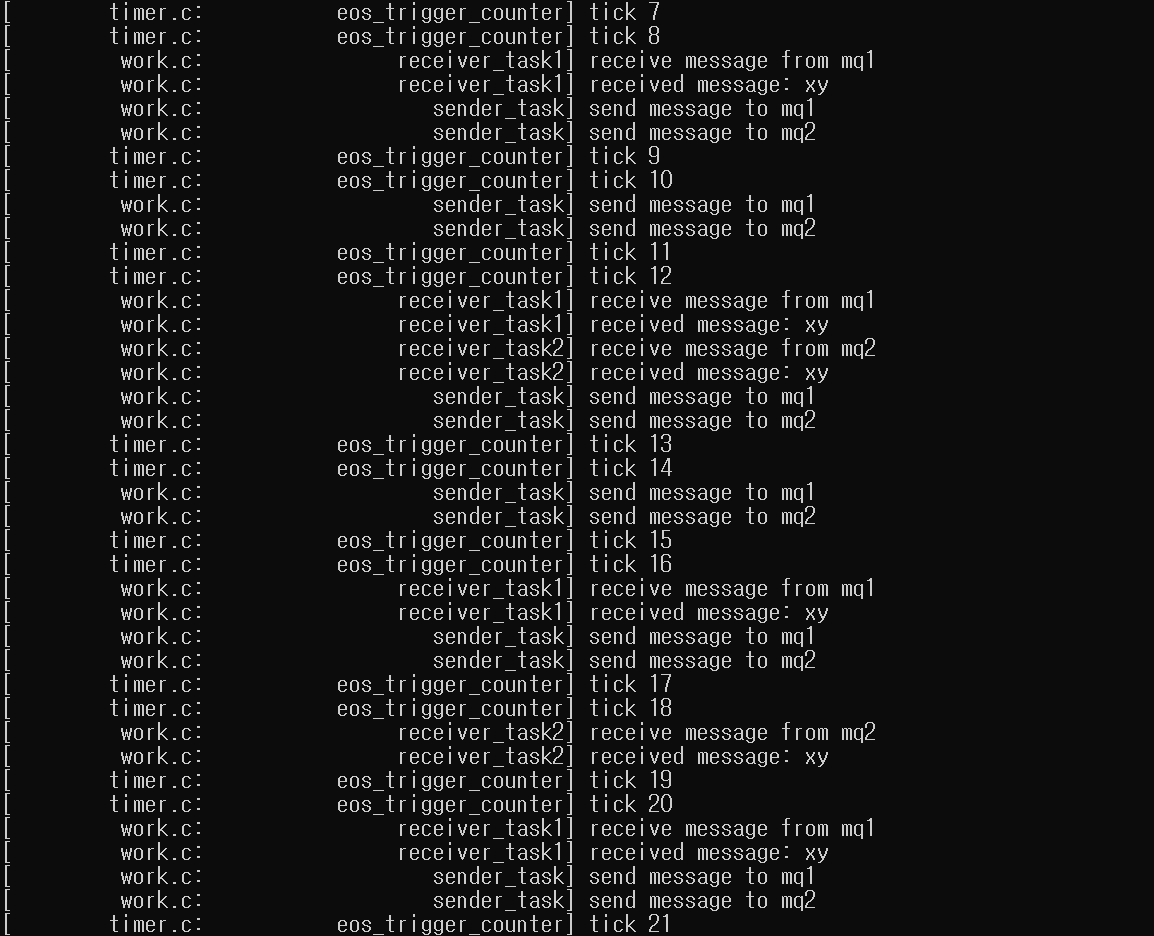
1. **eos\_receive\_message <eos/comm.c>**

eos\_send\_message와 완전히 반대로 동작하면 된다. eos\_acquire\_semaphore로 getsem이 1 이상일 때까지 timeout 동안 기다린다. 만약 얻지 못하면 0을 리턴한다. 얻은 경우 메시지에 메시지 큐의 내용을 입력한다. 변수가 void 포인터로 이루어져 있으나 테스트 프로그램에서 int8u\_t 변수를 사용하므로 아래와 같이 구현하였다.



eos\_release\_semaphore로 putsem에 1을 추가한다.

테스트 프로그램

....

tick 724까지 실행시켜보았다.

