OS_Project 4

2014-1995 송원재

1. eos_semaphore_t 구조체
- int32s_t count : 공유 자원 instance 개수
- int8u_t queue_type : wait_queue type (0 or 1)
- os_node_t* wait_queue : wait queue head

2. eos_init_semaphore 함수
- sem→ count = initial_count
- sem→ queue_type = queue_type
- sem→wait_queue = NULL 로 초기화

3. eos_acquire_semaphore 함수
- 기본 수행 : count > 0 이면 return 1
- timeout == -1 : return 0
- timeout == 0 : task 의 state = WAITING(3), task→ sem = sem
  queue_type 에 따라 wait_queue 에 노드 추가
  eos_schedule()
  restore 되었을 때 count 확인하고 실패하면 위에 다시 동작
- timeout > 0 : sem_alarm 선언 후 timeout 후에 울리도록 set_alarm
  timeout == 0 일 때와 마찬가지 동작
  단, alarm handler 로 인하여 task→sem == null 이 되어있는 경우 semaphore 얻기 실패하면 return 한다.

4. eos_release_semaphore 함수
- count 개수 1 늘리고 wait_queue 에 대기중인 task 있으면 먼저 wait_queue 에서 제거한 후에 wakeup

5. eos_mqueue_t 구조체
- int16u_t queue_size : 큐 사이즈
- int8u_t msg_size : 메세지 사이즈
- void* queue_start : 큐 시작지점
- void* front : 큐에서 실제 메세지 시작지점
- void* rear : 큐에서 실제 메세지 끝지점
- int8u_t queue_type : 큐 타입
- eos_semaphore_t putsem : 메세지 넣는 세마포
- eos_semaphore_t getsem : 메세지 얻는 세마포

6. eos_init_mqueue 함수
- Parameter 의 값들을 해당 mqueue 에 배당.
- putsem 의 초기 count = queue_size
- getsem 의 초기 count = 0 (비어있으므로)

7. eos_send_message 함수
- putsem 의 count 얻어오는데에 성공했을 경우 : rear 뒷부분부터 msg 삽입 (queue 의 끝값에 닿았을 때 front 로 돌아가기)
- 작업 후에 realease_semaphore(getsetm) 수행

8. eos_receive_message 함수
- getsem 세마포에서 count 얻어오기 성공한 경우에 front 부터 msg_size 만큼 받아와 출력
- 수행 후 release_semaphore(putsem) 실행

새로 추가한 함수
core/task.c → os_getout_semaphore_wait_queue(void* arg)
- 기능 : acquire_semaphore 에서 timeout > 0 일 때 set alarm 의 alarm_handler 함수
- 추가 이유 : alarm 이 울린 시점에 해당 task 를 semaphore 의 wait_queue 에서 먼저 제거한 후 ready_queue 에 넣어줘야 하는데 수행시킬 함수가 마땅히 존재하지 않음.
- 동작 : arg 로 받아온 task 의 sem 값으로 (eos_tcb_t 구조체에 변수 추가) 어떤 semaphore 의 wait_queue 에 들어가 있는지 파악한 후에 해당 wait_queue 에서 해당 task node 제거. 그 후에 wakeup 하여 ready_queue 에 추가.

9. 수행코드 결과

```
[          work.c:                 receiver_task1] receive message from mq1
[          work.c:                 receiver_task2] receive message from mq2
[          work.c:                    sender_task] Send message to mq1
[          work.c:                    sender_task] Send message to mq2
[          work.c:                 receiver_task1] received message: xy
[          work.c:                 receiver_task2] received message: xy
[         timer.c:           eos_trigger_counter] tick = 1
[         timer.c:           eos_trigger_counter] tick = 2
[          work.c:                    sender_task] Send message to mq1
[          work.c:                    sender_task] Send message to mq2
[         timer.c:           eos_trigger_counter] tick = 3
[         timer.c:           eos_trigger_counter] tick = 4
[          work.c:                 receiver_task1] receive message from mq1
[          work.c:                 receiver_task1] received message: xy
[          work.c:                    sender_task] Send message to mq1
[          work.c:                    sender_task] Send message to mq2
LibreOffice Writer
[         timer.c:           eos_trigger_counter] tick = 5
[         timer.c:           eos_trigger_counter] tick = 6
[          work.c:                 receiver_task2] receive message from mq2
[          work.c:                 receiver_task2] received message: xy
[          work.c:                    sender_task] Send message to mq1
[          work.c:                    sender_task] Send message to mq2
[         timer.c:           eos_trigger_counter] tick = 7
[         timer.c:           eos_trigger_counter] tick = 8
[          work.c:                 receiver_task1] receive message from mq1
[          work.c:                 receiver_task1] received message: xy
[          work.c:                    sender_task] Send message to mq1
[          work.c:                    sender_task] Send message to mq2
[         timer.c:           eos_trigger_counter] tick = 9
[         timer.c:           eos_trigger_counter] tick = 10
[          work.c:                    sender_task] Send message to mq1
[          work.c:                    sender_task] Send message to mq2
[         timer.c:           eos_trigger_counter] tick = 11
[         timer.c:           eos_trigger_counter] tick = 12
[          work.c:                 receiver_task1] receive message from mq1
[          work.c:                 receiver_task1] received message: xy
[          work.c:                 receiver_task2] receive message from mq2
[          work.c:                 receiver_task2] received message: xy
[          work.c:                    sender_task] Send message to mq1
[          work.c:                    sender_task] Send message to mq2
[         timer.c:           eos_trigger_counter] tick = 13
[         timer.c:           eos_trigger_counter] tick = 14
[          work.c:                    sender_task] Send message to mq1
[          work.c:                    sender_task] Send message to mq2
```

Left Shift + Left Alt

```
[         timer.c:          eos_trigger_counter] tick = 13
[         timer.c:          eos_trigger_counter] tick = 14
[          work.c:                 sender_task] Send message to mq1
[          work.c:                 sender_task] Send message to mq2
[         timer.c:          eos_trigger_counter] tick = 15
[         timer.c:          eos_trigger_counter] tick = 16
[          work.c:               receiver_task1] receive message from mq1
[          work.c:               receiver_task1] received message: xy
[          work.c:                 sender_task] Send message to mq1
[          work.c:                 sender_task] Send message to mq2
[         timer.c:          eos_trigger_counter] tick = 17
[         timer.c:          eos_trigger_counter] tick = 18
[          work.c:               receiver_task2] receive message from mq2
[          work.c:               receiver_task2] received message: xy
[         timer.c:          eos_trigger_counter] tick = 19
[         timer.c:          eos_trigger_counter] tick = 20
[          work.c:               receiver_task1] receive message from mq1
[          work.c:               receiver_task1] received message: xy
[          work.c:                 sender_task] Send message to mq1
[          work.c:                 sender_task] Send message to mq2
[         timer.c:          eos_trigger_counter] tick = 21
[         timer.c:          eos_trigger_counter] tick = 22
[         timer.c:          eos_trigger_counter] tick = 23
[         timer.c:          eos_trigger_counter] tick = 24
[          work.c:               receiver_task1] receive message from mq1
[          work.c:               receiver_task1] received message: xy
[          work.c:               receiver_task2] receive message from mq2
[          work.c:               receiver_task2] received message: xy
[         timer.c:          eos_trigger_counter] tick = 25
[         timer.c:          eos_trigger_counter] tick = 26
[          work.c:                 sender_task] Send message to mq1
[          work.c:                 sender_task] Send message to mq2
[         timer.c:          eos_trigger_counter] tick = 27
[         timer.c:          eos_trigger_counter] tick = 28
[          work.c:               receiver_task1] receive message from mq1
[          work.c:               receiver_task1] received message: xy
```

LibreOffice Writer

Left Shift + Left Alt

```
[         timer.c:          eos_trigger_counter] tick = 27
[         timer.c:          eos_trigger_counter] tick = 28
[          work.c:               receiver_task1] receive message from mq1
[          work.c:               receiver_task1] received message: xy
[         timer.c:          eos_trigger_counter] tick = 29
[         timer.c:          eos_trigger_counter] tick = 30
[          work.c:               receiver_task2] receive message from mq2
[          work.c:               receiver_task2] received message: xy
[         timer.c:          eos_trigger_counter] tick = 31
[         timer.c:          eos_trigger_counter] tick = 32
[          work.c:               receiver_task1] receive message from mq1
[          work.c:               receiver_task1] received message: xy
[          work.c:                 sender_task] Send message to mq1
[          work.c:                 sender_task] Send message to mq2
[         timer.c:          eos_trigger_counter] tick = 33
[         timer.c:          eos_trigger_counter] tick = 34
[         timer.c:          eos_trigger_counter] tick = 35
[         timer.c:          eos_trigger_counter] tick = 36
[          work.c:               receiver_task1] receive message from mq1
[          work.c:               receiver_task1] received message: xy
[          work.c:               receiver_task2] receive message from mq2
[          work.c:               receiver_task2] received message: xy
[         timer.c:          eos_trigger_counter] tick = 37
[         timer.c:          eos_trigger_counter] tick = 38
[          work.c:                 sender_task] Send message to mq1
[          work.c:                 sender_task] Send message to mq2
[         timer.c:          eos_trigger_counter] tick = 39
```

Left Shift + Left Alt