운영체제의 기초 Project 3

전기정보공학부

2017-17088 박찬정

**수정한 API**

**eos\_tcb\_t**: 실행 주기에 대한 값인 period와, task 실행 시점에 대한 값인 started\_at을 추가함. 타입은 모두 int32u\_t임.

**eos\_schedule()**: 멀티 레벨 큐는 이미 구현이 되어있어 state 관리하도록 코드 추가.

**eos\_create\_task()**: tcb의 state, started\_at 값을 관리하도록 코드 추가.

**새로 구현한 API**

**eos\_set\_period()**: 인자로 받은 task의 period 값을 인자로 받은 period 값으로 저장함.

**eos\_sleep()**: eos\_alarm\_t 구조체를 하나 새로 생성하고, timeout 값을 계산하여 eos\_set\_alarm 함수를 통해 알람 큐에 저장함.

**\_os\_wakeup\_sleeping\_task()**: 인자로 받은 task를 깨우고, started\_at과 state 값을 관리함.

**eos\_set\_alarm()**: 인자로 받은 counter에서 인자로 받은 alarm을 삭제하고 시작함. Timeout이 0이거나 entry가 NULL인 경우 그대로 함수를 끝내고, 그렇지 않은 경우 alarm의 내부 값을 관리함. 이후 \_os\_add\_node\_priority 함수를 사용하여 counter에 alarm을 다시 넣어줌.

**eos\_trigger\_counter()**: 인자로 받은 counter의 tick을 1 증가시킴. 이후 알람 큐의 헤드를 반복하여 확인하면서 현재 tick 값이 timeout 값과 동일한 경우 큐에서 pop시키고, handler를 실행시킴. 작업이 끝나면 eos\_schedule 함수를 호출하여 나머지 task가 실행되도록 함.

**실행 결과**

텍스트이(가) 표시된 사진

자동 생성된 설명