

OS Assignment

team 27 :

20150109 김동균

20150195 김정우

Question 1)

케밥 가게:

- 대기실 : 의자 n개
- 조리실

규칙:

1. 손님이 없으면 조리실은 sleep한다.
2. 대기실이 꽉 차 있는데 손님이 오면 그 손님은 대기하지 않고 즉시 나간다.
3. 대기실에 남은 의자가 있으면 손님이 와서 대기할 수 있다.
4. 조리실이 sleep 중인데 손님이 오면 조리실이 깨어나야 한다.

조리실은 초기값 0인 세마포어 sema_cook을 가지고 있고 각 손님은 초기값 0인 세마포어 sema_wait를 가지고 있다 가정하자.

cook:

repeat:

// 규칙 1

if no customer in waiting room:

P(sema_cook)

// 손님에게 줄 케밥을 만든다.

making_kebab()

// 손님을 깨운다.

V(sema_wait of the longest-waited customer in waiting room)

// 손님이 케밥을 가져갈 때까지 대기

P(sema_cook)

customer:

// 규칙 2

if waiting room is full:

leave_without_kebab()

// 규칙 4

if no customer in waiting room:

V(sema_cook)

// 규칙 3

enter to waiting room

// 케밥이 나올 때까지 대기

P(sema_wait of itself)

// 케밥이 나오면 대기실에서 나가고 케밥을 얻은 뒤 조리실을 깨운다.

exit from waiting room

getting_kebab()

V(sema_cook)

Question 2)

```
struct lock {  
    int is_blocked;    // 대기 중인 프로세스가 있으면 1, 없으면 0. 0으로 초기화.  
};  
  
void acquire(struct lock *lock) {  
    while (!compare_and_swap(&lock->is_locked, 0, 1));  
}  
  
void release(struct lock *lock) {  
    compare_and_swap(&lock->is_locked, 1, 0);  
}
```

Question 3)

프로세스 시스템: