EDDI
Electronic Design
Development Institute

에디로봇아카데미
임베디드 마스터 Lv2 과정

제 1기

2021. 12. 20

박성환

CONTENTS

EDDI
Electronic Design
Development Institute

• 원형 큐 - 전략

EDDI
Electronic Design Development Institute

To do

1) Full 과 Empty를 어떻게 구별할 것인지

2) Head 와 Tail을 통해 어떻게 데이터를 읽고 쓸것인지
   ① Head 와 Tail을 포인터로 처리할지 index로 처리할지 고민하기

3) 모든 상황 버퍼 시작과 끝 부분에서 예외 사항이 있는지 확인

4) 매번 다른 목적으로 원형 큐를 생성해서 독립적으로 사용 가능하도록 함

Strategy

1) Full : (Head + 1) == Tail 인 경우
   Empty : Head == Tail 인 경우
   *Full 같은 경우는 1개의 버퍼는 채우지 못하고 Full로 될 수 밖에 없음
   원형 큐이기 때문에 한 개의 버퍼 정도는 Full 구별용으로 쓰는데 지장 없음

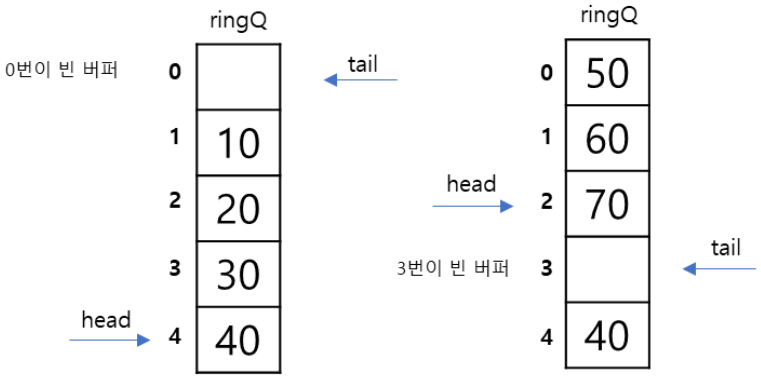
2) 방법의 차이겠지만 포인터보다 index로 +1씩 증가하면서 읽고 쓰는게 코드구현상 더 깔끔하다고 느낌
   ① Set 명령으로 쓰면 Head = Head + 1로 이동 후 데이터 쓰기
   Get 명령으로 읽으면 Tail = Tail + 1로 이동 후 읽기

3) 1), 2), 3)의 조건이면 다른 예외조건 발생하지 않음

4) head, tail, buf는 한몸으로 독립적으로 필요하기 때문에 구조체로 묶어서 여러 변수 만들어 각각 사용 가능하도록 함

• 원형 큐 – Full()

그림 예시



이 다음 set은 Full 조건
(head + 1) == tail 만족하므로
더 이상 쓸 수 없음

코드 구현

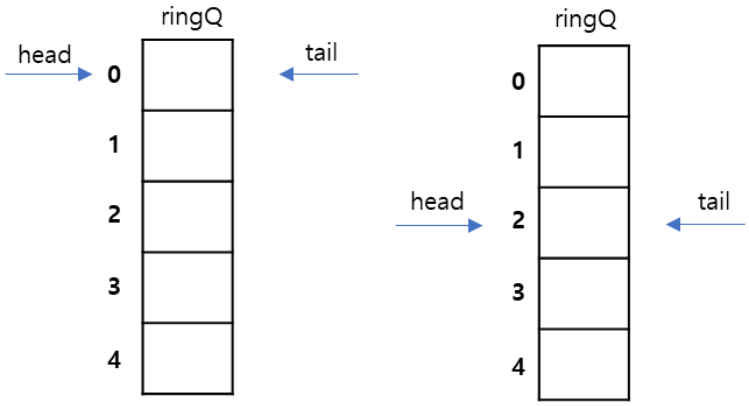
```
bool ringQ_is_Full(ringQ* ringq)
{                                   1)
    if( (ringq->head + 1)%BUF_SIZ == ringq->tail ) {
        return true; 2)
    }
    return false;
}
```

To do

1) Full 체크, Head +1 == Tail 이면 Full로 return true;
2) Head + 1이 5개를 넘지 않도록
   (Head + 1) % (BUF_SIZ)
3) 공백 하나를 두어서 Full 상태를 구분하도록 함

• 원형 큐 – Empty()

그림 예시



어떠한 상황에도 head == tail
이면 Empty

코드 구현

```
bool ringQ_is_Empty(ringQ* ringq)
{
    if( ringq->head == ringq->tail ) {
        return true;
    }
    return false;
}
```

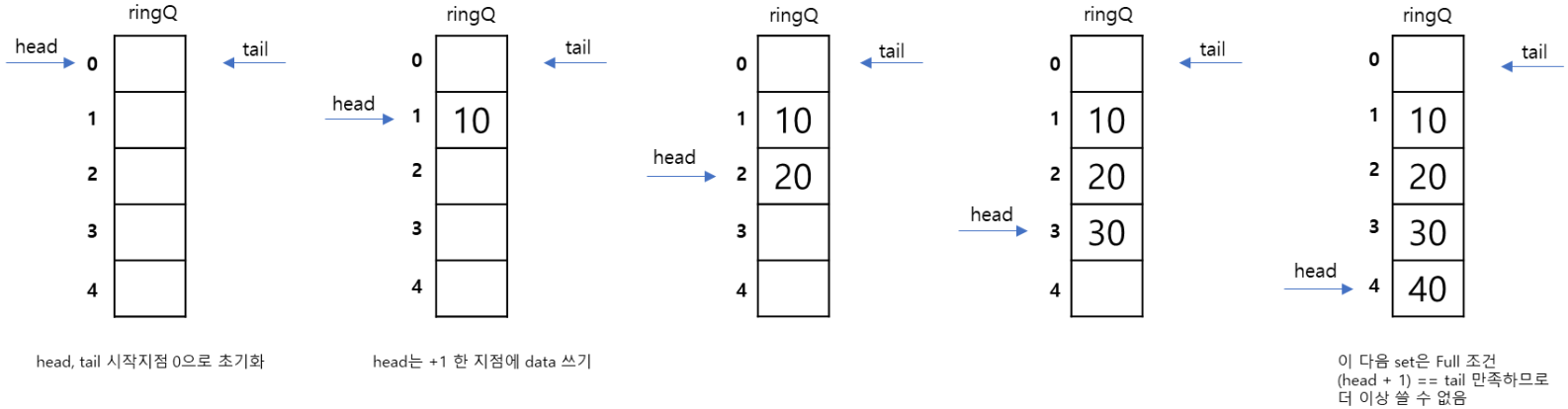
To do

1) Head 와 Tail이 같으면 어떠한 조건 상황에도 Empty임

• 원형 큐 – Set()

EDDI
Electronic Design
Development Institute

그림 예시



head, tail 시작지점 0으로 초기화

head는 +1 한 지점에 data 쓰기

이 다음 set은 Full 조건
(head + 1) == tail 만족하므로
더 이상 쓸 수 없음

• 원형 큐 – Set()

EDDI
Electronic Design Development Institute

함수

```
void ringQ_set(ringQ* ringq, int data)  1회 set
{
                          1)
    if( ringQ_is_Full(ringq) ) {
        printf("ring buffer is Full\n");
        return;
    }
                                  2),3)
    ringq->head = (++ringq->head)%BUF_SIZ;
    ringq->buf[ringq->head] = data;
    printf("ringq_set[%d] = %d\n", ringq->head, ringq->buf[ringq->head]);
}
```

```
void ringQ_set_block(ringQ* ringq, int *data, int size)
{                                                  size 만큼 set
    static int cnt = 0;

    while(size--)
    {                  1)
        if(ringQ_is_Full(ringq)) {
            printf("ring buffer is Full\n");
            return;
        }
        ringQ_set(ringq, *data++);
    }
}
```

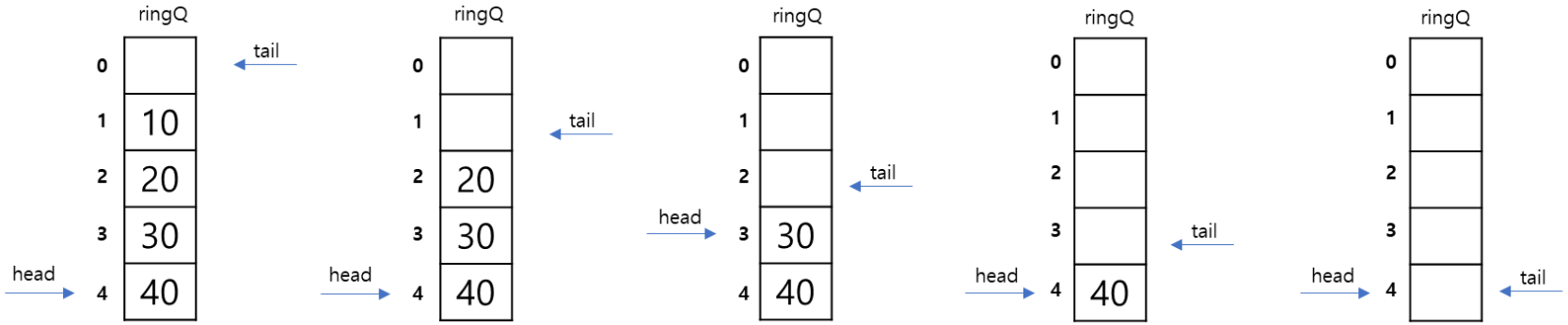
To do

1) Full 체크, Full이면 더 이상 쓰지 않고 return
2) Head = Head + 1로 이동 후, 데이터 쓰기
3) rinQ 버퍼 사이즈가 5개이면 head가 증가하면서 5가 넘지 않도록
   Head = (Head + 1) % (BUF_SIZ)

• 원형 큐 – Get()

EDDI
Electronic Design
Development Institute

그림 예시



index 1부터 읽기 시작

이 다음 get은 Empty 조건
head == tail 이므로
더 이상 읽을 수 없음

• 원형 큐 – Get()

EDDI
Electronic Design Development Institute

함수

```
int ringQ_get(ringQ* ringq) 1회 get
{
    int data;

    if( ringQ_is_Empty(ringq) ) {   1)
        printf("ring buffer is Empty\n");
        return -1;
    }

    ringq->tail = (++ringq->tail)%BUF_SIZ;   2),3)
    data = ringq->buf[ringq->tail];
    printf("ringq_get[%d] = %d\n", ringq->tail, data);

    return data;
}
```

```
int ringQ_get_block(ringQ* ringq, int size)
{                                   size 만큼 get
    int data;

    while(size--)
    {
        if( ringQ_is_Empty(ringq) ) {   1)
            printf("ring buffer is Empty\n");
            return -1;
        }

        data = ringQ_get(ringq);
    }
    return data;
}
```

To do

1) Empty 체크, Empty 이면 더 이상 읽지 않고 return
2) Tail = Tail + 1로 이동 후, 데이터 읽기
3) rinQ 버퍼 사이즈가 5개이면 Tail이 증가하면서 5가 넘지 않도록
   Tail = (Tail + 1) % (BUF_SIZ)