mud lecture

[Rock lecture] 5. Pointer example - Reducing unit limit

why do you ask
2014. 1. 23. 23:48

It is very difficult to increase the number of units. It's hard to beg.
First of all, the conclusion I have reached is 'theoretically impossible'.
Even if you go through all the difficulties up to the structural offset, you will be stuck at the .text: 00430A04 part.
(This part is used for selecting units by dragging, etc.)
The .data:006BD3D0 part is an array with exactly 1700 elements.

But reducing the unit limit is easy.

Let's start with the practice questions from the previous lecture.
In my previous lectures, I gave too much explanation.

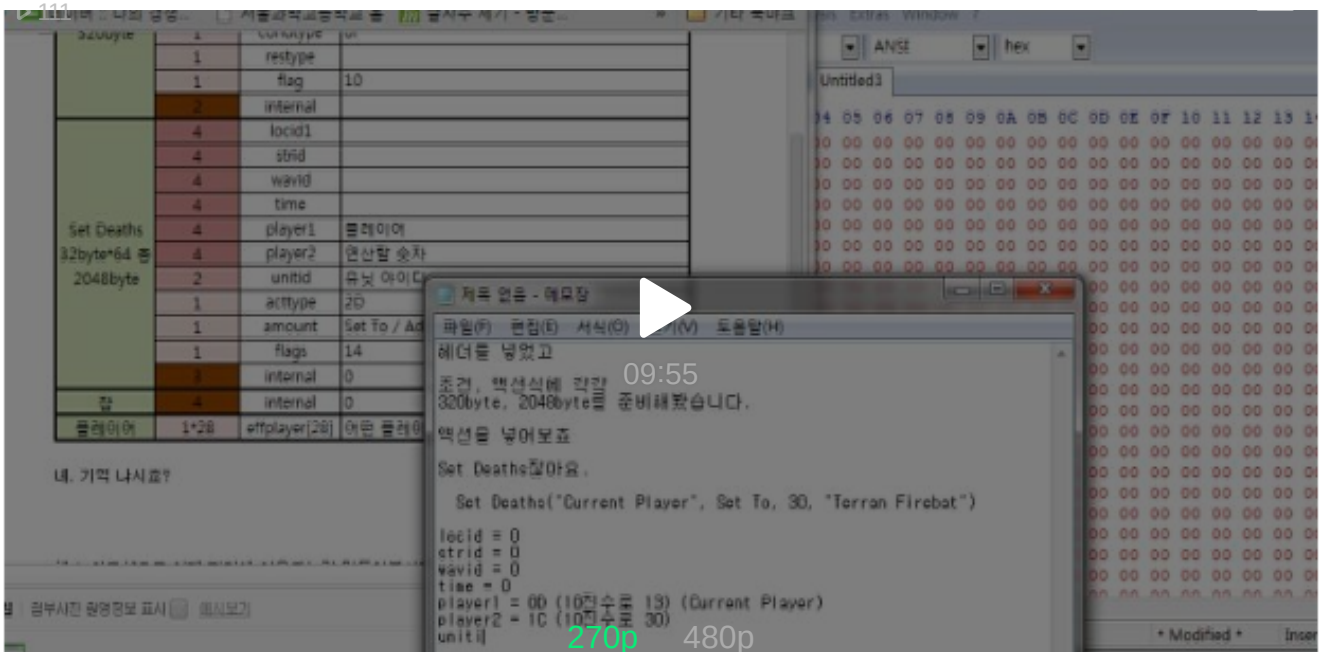
```
Triggers("Player 1") {
Conditions:
  Deaths("Player 7", At Least, 1,  "Terran Marine");

Actions:
  Set Deaths("Current Player", Set To, 30, "Terran Firebat")
}
```

분류	크기	이름	설명 (자주 쓰이는 용도를 기준으로 함)
Deaths 20byte*16 층 320byte	4	locid	
	4	player	플레이어 번호
	4	amount	비교할 양
	2	unitid	유닛 아이디. Terran marine = 0.
	1	comparison	At least / At most / Exactly
	1	condtype	0F
	1	restype	
	1	flag	10
	2	internal	
Set Deaths 32byte*64 층 2048byte	4	locid1	
	4	strid	
	4	wavid	
	4	time	
	4	player1	플레이어
	4	player2	연산할 숫자
	2	unitid	유닛 아이디
	1	acttype	2D
	1	amount	Set To / Add / Subtract
	1	flags	14
	3	internal	0
잡	4	internal	0
플레이어	1*28	effplayer[28]	어떤 플레이어에게 이 트리거가 적용되는가?

Yes. Remember?

Now let's make something used for that with the hex workshop.
Replace with video.

[Rock lecture] 5. Pointer example - Reducing unit limit



whyask37's blog

I didn't explain about effplayer in the last lecture.
It was a strange problem to solve the practice problems;

Originally, I was thinking of increasing the unit limit by twisting the TRIG paragraph a lot.
It must be hard. Sorry.

Now let's get to the point.

The variable that counts the current number of units in the star is at 006283F0.
So when this offset is exactly 1700, a Cannot error is displayed.
So this offset starts from 0
If you start at 1200, it seems like you can build a maximum of 500 units.

SetDeaths(161827, Add, 1200, 0);

It will end with one line.

But in this case, 1700 CUnits will be used by 500 units.

In order to make it easier to apply
Let's make CUnit 0 to 499 500 units return. (500th unit is number 499)
Let's start applying pointers in earnest.

1. When there are 500 units, the cans are made to float. SetDeaths(7, Add, 1200, 13485);
2. Make sure that only the 0~499th CUnit structures are written. <- Let's learn this.

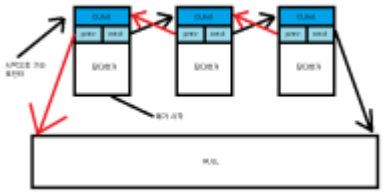
---

goal

1. CUnit is managed as a doubly linked list. A more detailed look at how Star utilizes doubly lin ked lists.
2. Find out what it means to manipulate pointers.

whyask37's blog

Remember this picture?

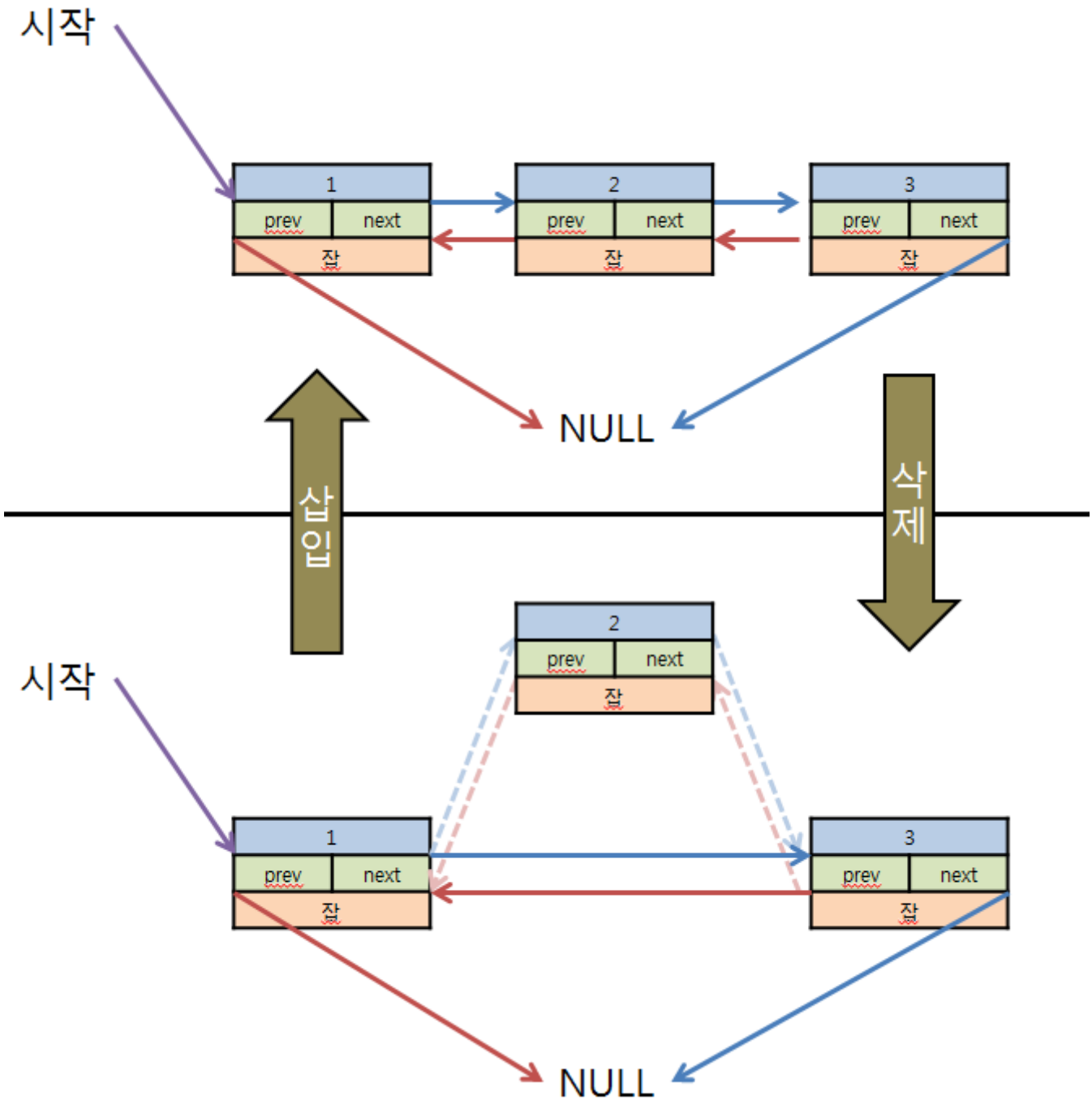


In a doubly linked list, units can be manipulated through prev and next pointers.
Now let's play with this doubly linked list. First, let's learn how to insert and delete elements, whi ch are basic among the basics .

1. How to Manage CUnit in Star

Insertion and deletion of elements in a double list are performed by changing only the pointer a s shown in the figure below.

whyask37's blog



스타에서는 흔히 말하는 head랑 tail에도 데이터를 채워넣습니다.

Q) 맨 처음 원소를 삭제하면 어떻게 되나요?
A) 맨 처음 원소를 가르키는 포인터가 2번째 원소를 가르키게 되고, 2번째 원소의 prev가 NULL이 되고, 끝.
원소가 하나밖에 없었다면 맨 처음 원소를 가르키는 포인터가 NULL을 가르키고 끝.

Q) 맨 끝을 가르키는 포인터는 왜 없어요?
A) 사실 있어요. 그림에서는 까먹고 생략했어요. 맨 끝에 원소 삭제할때도 위 답변과 비스무리하게 해요.

---

whyask37's blog

- E : 지금 쓰이지 않는 Unitnode table의 원소들의 리스트
- U : 지금 쓰이고 있는 Unitnode table의 원소들의 리스트

처음에 E는 1700개의 Unitnode 테이블에 있는 모든 유닛들을 담고 있습니다.
그 다음 유닛 생성시에 (처음 맵을 불러올 때나 마린 뽑을 때 등)

```
TotalUnitNum == 1700:
 캔낫을 띄우고 끝

E에 원소가 남아 있으면
 - E에서 CUnit 하나를 빼온다. (E에서 이 CUnit은 삭제됨)
 - 생성할 유닛에 맞춰서 빼낸 CUnit을 초기화시킨다.
 - U에 E에서 빼낸 CUnit을 삽입한다.
아니면:
 - 에러를 띄우고 끝
```

유닛 파괴시에는 그 유닛을 U에서 삭제해서 다시 E에 집어넣겠죠.

글만 있는게 조금 이해하기 어렵다고는 하던데, 매번 그림그리기도 그렇고 그냥 글로 보세요.

이제 스타에서 CUnit이 어디에 위치하는지를 조금 더 자세하게 알아봅시다.

스타에서는 0059CCA8부터 336byte짜리 CUnit이 1700개가 나열되어있습니다. (배열)
그래서 맨 처음 시작할 때에 이 1700개를 전부 E에 넣습니다.

맨 처음에 1699번 유닛이 처음의 E 유닛이고, 0번 유닛이 마지막 E 유닛입니다.

사족) 유닛은 0~1699번이 있고요, 각각 0번에서부터 1~1700번째 유닛이고 1699번에서 1700~1번째 유닛입니다. 번이랑 번째 구분

그 다음 맵에 미리 배치해둔 유닛이 나타날때마다
E에서 하날 빼서 U에 넣고 (그러니까 1699번 유닛)
E에서 하날 빼서 U에 넣고 (1698번 유닛)
... (이 순서대로...)

합니다. 1700개 E를 모두 쓰면 그 다음부터는 U에서 E로 들어온 순서대로 다시 계속 메모리를 돌게 됩니다.

간단하게 이야기합시다.
A ==> B 하면 A의 prev가 B를 가르키고 B의 prev가 A를 가르키고 라는 뜻입니다.

(A --> B는 그냥 A의 next가 B를 가르킨다)

whyask37's blog

E U E시작
(1699) [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] (0)

이런 상태입니다.

유닛 하나 생성

E U
(1699) [유닛] [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] (0)
(1699) [유닛] ==> [유닛] [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] (0)
(1699) [유닛] ==> [유닛] ==> [유닛] [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] (0)
(1699) [유닛] ==> [유닛] ==> [유닛] ==> [유닛] [유닛] ==> [유닛] ==> [유닛] ==> [유닛] (0)
(1699) [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] [유닛] ==> [유닛] ==> [유닛] (0)

이런 식으로 계속 E에서 하나를 빼서 U에 넣는 작업이 계속됩니다.
E의 시작부분에서 하나를 떼서 U에 넣는거죠.

그러면 이 작업이 안되도록 하려면?

예를 들어 위에서 유닛 수 제한을 5개로 두고 싶다 합시다.
그러면 E를 5개째에서 잘라버리면

(1699) [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] | [유닛] ==> [유닛] ==> [유닛] (0)
(1699) [유닛] [유닛] ==> [유닛] ==> [유닛] ==> [유닛] | [유닛] ==> [유닛] ==> [유닛] (0)
(1699) [유닛] ==> [유닛] [유닛] ==> [유닛] ==> [유닛] | [유닛] ==> [유닛] ==> [유닛] (0)
(1699) [유닛] ==> [유닛] ==> [유닛] [유닛] ==> [유닛] | [유닛] ==> [유닛] ==> [유닛] (0)
(1699) [유닛] ==> [유닛] ==> [유닛] ==> [유닛] [유닛] | [유닛] ==> [유닛] ==> [유닛] (0)
(1699) [유닛] ==> [유닛] ==> [유닛] ==> [유닛] ==> [유닛] | [유닛] ==> [유닛] ==> [유닛] (0)

더이상 E에 남아있는게 없으므로 유닛 생성이 안되겠죠.
그리고 저 5개 안에서 유닛들이 계속 생성되겠죠.

죽은 U들은 다시 E에 들어가고
그 죽은 U들도 원래 저 5개 안에 있던 E 속에 있던 CUnit이었을거고
따라서 다시 E에 들어가는것도 저 5개 안
U에 들어가는것들도 저 5개 안

트리거를 짜봅시다. 이번에는 매우 간단합니다.
예시로 500개 유닛 제한을 걸어봅시다.

whyask37's blog

즉, 1200번 CUnit의 next는 원래 1199번 CUnit을 가르키고 있었을텐데, 이걸 NULL을 가르키도록 포인터를 조작하고
E의 마지막 원소를 가르키는 포인터가 원래 0번 CUnit을 가르키고 있었을텐데, 1200번 CUnit을 가르키도록 하면 됩니다.

사족) 원칙적으로야, 1199번째 CUnit의 prev도 NULL로 만들어야겠지만, 어차피 1200번 CUnit에서 1199번 CUnit을 못 가도록 하는게 목표,

---

사실 지금까지 사기를 쳤습니다.
실제 E의 구조는 다음과 같습니다.

0이 시작이고
E시작 : 0 - 1699 - 1698 - 1697 - .... - 4 - 3 - 2 - 1 **: E끝**
이따구로 되있어요. 좀 짜증나네요. 왜 왔다갔다하는거야

그러므로 위의 논의에서 1200과 1199는 각각 1201과 1200으로 바꿔서 읽어주시면 됩니다.
...라고 했는데 이렇게 되면 구조오프셋 0, 1201~1699번이 사용되는 하여튼 뭔가 빡치네요.

밑에부터는 사기 안칠께요.

트리거를 만들어봅시다.
- 마지막 E 유닛을 가르키는 포인터를 1201번 유닛을 가르키도록 조작합니다.
- 1201번째 유닛의 next를 NULL (0) 으로 바꿉니다.
- 그 다음에, 총 유닛 수를 **1200**만큼 증가시킵니다. <- **여긴 그대로**

마지막 E 유닛을 가르키는 포인터는 0x0062843C입니다.

Q) 이런 괴상한 오프셋들을 어떻게 찾는가요?
A) Ollydbg를 쓰든, IDA Pro를 쓰든, 둘 다 쓰든 (전 둘 다 함께 씁니다. 디버거 - 디스어셈블러 조합) 해가지고 스타를 뜯으면 나와요.
비프로그래머들은 하기 힘들어요.

whyask37's blog

Unitnode Structure에 따르면
CUnit* next; 는 각 CUnit의 +008 오프셋에 있다고 합니다.

0059CCA8 이 Unitnode Table의 오프셋이고
1200번 유닛의 오프셋은 0x0059CCA8 + 336 * 1201 = 0x005FF4F8
1200번 유닛의 next의 오프셋은 위에거에 +4하면 되니까 0x005FF4FC
에 해당하는 EPD의 플레이어는 119910네요.

이렇게 하면 될듯합니다.

* 마지막 E 유닛을 가르키는 포인터를 1200번 유닛을 가르키도록 조작합니다.
  SetDeaths(161846, SetTo, 0x005FF4F8, 0);
  인데 스타포지에서 16진수는 인식 못하니까 SetDeaths(161846, SetTo, 6288632, 0);
* 500번째 유닛의 next를 NULL (0) 으로 바꿉니다.
  SetDeaths(119910, SetTo, 0, 0);
* 그 다음에, 총 유닛 수를 1200만큼 증가시킵니다.
  SetDeaths(161827, Add, 1200, 0);

이렇게 하면 0~1199번 유닛은 ㅂㅂ되는거죠.

(0~499번 유닛을 쓰고싶고 500~1699유닛을 없애고 싶으시다면 E에서 처음 시작하는 원소 가르키는 포인터가 0x00628438에 있으니까 이것도 잘 만지작하세요)

그래서 실제로 만들어보면 다음과 같이
500개만 생성되는걸 알 수 있습니다.

사족) SetDeaths(161827, Add, 1200, 0); 를 지워도 500개밖에 안생기기는 하는데, 이러면 501번째 유닛을 만들려고 할 때 캔낫이 아니라 이상한 에러가 떠서 알아보기 힘습니다.)

첨부파일의 500test.scx의 Player 1 트리거 참고.
꽤나 간단한 포인터 조작 예제지만 그래도 포인터 자체가 익숙해지기 어려운 개념이라 하나 만들어봤습니다.
연결 리스트를 연습하는 좋은 예제도 되고요.

연습문제 1 : 예제로 올린 500test.scx처럼 맵에 원래 놓여있던 유닛이 스타트 로케이션밖에 없으면 0~499번 CUnit만 쓰이도록 잘 조작할 수 있다. 어떻게 이렇게 할 수 있을까?
Hint: E의 시작을 가르키는 포인터는 00628438에 있고 끝을 가르키는 포인터은 0062843C에 있습니다.

whyask37's blog

어떻게 하든지 님 마음대로.

연습문제 2 : 유닛 제한을 700개로 바꿔보자.

사족) 맵 시작할 때 유닛이 2개 이상 있을 경우 0~499번 CUnit만 쓰이게 하려면 구조오프셋 외에 UnitFinder라는 특이한 자료구조도 다뤄야 하는데, 이거 설명하려면 너무 어려우므로 이 강좌에선 다루지 않습니다. (이분 검색을 X축 Y축으로 하게 해주는 자료구조인데, 이것만 없었어도 유닛 제한을 5000까지 늘릴 수 있었는데 참 아쉽네요) 그냥 유닛 제한을 500개로 하고싶다면 위 강좌대로 하면 됩니다.

#IT·컴퓨터

첨부파일

500test.scx

0

[뻘강의] 5. 포인터 예제 - 유닛 제한 줄이기

2014. 1. 23.

0

[뻘강의] 4. TRG 파일 포맷

2014. 1. 21.

3

[뻘강의] 3. 포인터

2014. 1. 21.

0

이 블로그 인기글

MPQ 가지고놀기 (1) - 간단한 MPQ 파일 분석

2013. 10. 19.

11

5. SFmpq (ShadowFlare's MPQ Library) 와 예제

2013. 9. 11.

1

[뻘강의] 13. 트리거 프로그래밍 - TRIG-MRGN 루프

2014. 2. 24.

0

4. scenario.chk

2013. 9. 10.

0

whyask37's blog

2014. 1. 19.

1

back to top

View in PC version

0