

절체절명의 해누리호!

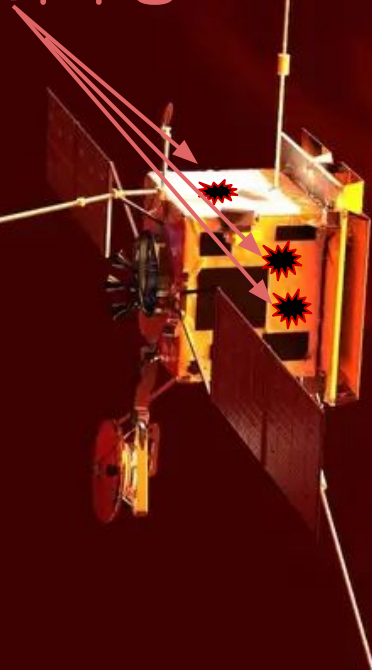
임의로 손상이 발생하는
RAM에 대응하는 가상 메모리 디자인



◁ (왼쪽부터) 해누리호 마스코트 “해나(Henna)” / 해누리호 뱃지
△ 해누리호 발사체 전경 (제공: 해누리호 탐사 센터)



우주 쓰레기로
인한 여러 구멍



△ 우주 쓰레기와 충돌하여 옆면 제어 컴퓨터 영역에 구멍이 뚫린 해누리호 상상도 (제공: 해누리호 탐사 센터)

CS330에 깊게 감명받은 당신은 URP 과제로 PintOS 기반 탐사선 OS를 개발했다. 해당 OS는 세계 유수의 OS를 누르고 **한국형 태양 탐사선 해누리호**의 OS로 선정되었다.

발사는 성공적이었다. 하지만 해누리호는 태양의 L1 라그랑주 point 진입과정 중 다수의 우주쓰레기와 충돌해, RAM을 보호하던 방사선 차폐막에 몇 개의 작은 구멍이 뚫리는 손상을 입었다. 다행히 이전의 경험을 토대로 태양광에 강한 FRAM을 탑재해 태양광의 영향은 피할 수 있었지만, **아차! 흑점 폭발**에 의한 태양풍을 미처 고려하지 못해 이온 입자에 영향을 받는 RAM의 일정 부분을 일시적으로 사용할 수 없게 되었다.

이제 해누리호의 운명은 당신에게 달렸다. 과제와 시험 준비로 정신 없지만, 당신은 **4월 30일**까지 해누리호의 OS에 긴급 패치를 작성해야 한다.

만약 성공하지 못한다면, 국민의 혈세가 들어간 해누리호 뿐만 아니라 당신의 학점 또한 ***폭발***할 수 있다.

주기적으로 손상되는 RAM을 위한 VM 시스템 디자인

- 태양풍 노출로 인해 발생하는 frame의 손상을 피할 수 없다.
- 당신은 손상으로 인한 속도 저하를 줄이려고 노력하며,
탐사선의 안전한 작동을 보장하는 새로운 VM 디자인을 제시해야 한다.
 - 태양풍에 노출되고 있는 메모리 영역에 대한 접근을 원천적으로 차단해야 한다.
 - 해당 디자인의 도입에 따른 속도 저하를 줄이려고 노력해야 한다.
- 애플리케이션에 어떤 수정도 요구해서는 안된다.

문제 상황: 태양풍 노출로 인한 RAM 데이터 손상

- 태양풍을 받고 있는 RAM 영역을 CPU가 읽으면 탐사선이 **폭발!**
 - 태양풍을 받는 RAM 영역은 이온 입자에 의한 교란으로 인해 사용이 불가능.
 - 어떠한 경우에도 OS/유저 프로그램은 해당 RAM 영역에 접근하지 말아야 함.
- 태양풍에 노출되었던 RAM은 태양풍의 영향이 사라지면 다시 재사용 가능
 - 온도가 낮아져서 해당 영역을 다시 사용할 수 있게 됨.
 - 단, 데이터는 이온 입자에 의해 손상됨.
 - 해당 영역에 저장된 값은 무작위로 변화.



△ 폭발하는 탐사선 자료화면 (제공: 해누리호 탐사 센터)
여러분의 학점이 될지도 모르는 일이다.

디자인 목표

- 해누리호가 폭파되는 일은 **절대** 발생하지 않아야 한다.
- 애플리케이션은 수정할 수 없다.
- 데이터 손상으로 인한 애플리케이션의 오동작이 없어야 한다.
- 속도 저하를 줄이려고 노력해야 한다.
 - 예: swap-in/out 을 최대한 피하기

시간에 따른 태양풍 이온 입자 영향의 변화

태양 활동 및 흑점 폭발에 따라 RAM에 태양풍에 영향을 받는 영역이 정해진 주기(30틱)로 랜덤하게 변화한다.

한 시점에 최소 0개에서 14개 frame이 영향을 받으며, 이 frame은 연속할 수도, 연속하지 않을 수도 있다.

카이스트 인공위성 센터 인턴 한수경이 다음 알고리즘을 개발함:

- 가시광선의 배출과 태양풍의 도달에 차이가 있어, 이 차이를 이용하여 지금으로부터 최대 30틱 후 태양풍에 영향을 받을 frame을 계산해내는 알고리즘

이 알고리즘을 이용한 프로그램은 위성에서 긴급 배포되어 있다.

다행히도 Kernel pool 영역의 RAM은 태양풍 안전구역에 위치하여 이온 입자의 영향을 받지 않는다.

(다음 페이지 모식도 참조)

흑점
폭발

* KST: Korea Standard Time
12:00:00 KST

이온 입자가 메모리 셀에 닿으면
해당 셀 내부 데이터가 손상됨

태양풍 안전구역.
Kernel page들이 저장되어 있다.



흑점
폭발

12:30:00 KST

이온 입자가 메모리 셀에 닿으면
해당 셀 내부 데이터가 손상됨

태양풍 안전구역.
Kernel page들이 저장되어 있다.



시간에 따라 교란/손상되는 메모리 영역 도식도 예시

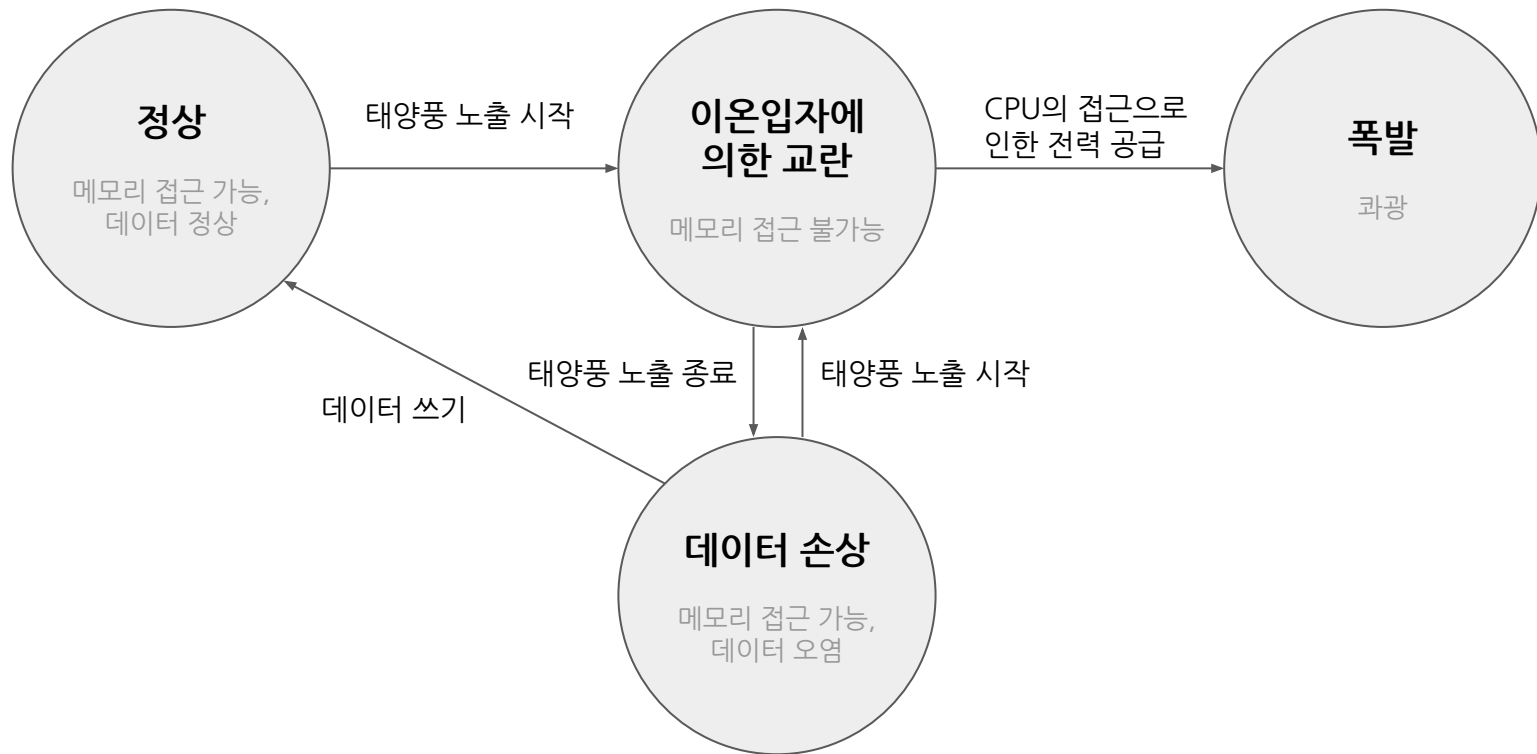
(그 어떤 메모리 사용도 하지 않은 경우)

시간(분)

Physical
Address

	0/2000	1/2000	2/2000	...	1997/2000	1998/2000	1999/2000
0x0000000				...	데이터 손상	데이터 손상	데이터 손상
0x0001000		교란	데이터 손상	...	데이터 손상	데이터 손상	데이터 손상
0x0002000				...	데이터 손상	데이터 손상	데이터 손상
0x0003000				...		교란	데이터 손상
0x0004000	교란	교란	데이터 손상	...	데이터 손상	교란	데이터 손상
0x0005000				...	교란	교란	데이터 손상
0x0006000	교란	데이터 손상	데이터 손상	...	데이터 손상	데이터 손상	데이터 손상
0x0007000				...	데이터 손상	데이터 손상	데이터 손상
...				...			
...				...			
...				...			
...				...			
...				...			
0x07cd000				...	교란	데이터 손상	데이터 손상
0x07ce000	교란	데이터 손상	데이터 손상	...	데이터 손상	데이터 손상	데이터 손상
0x07cf000				...			교란

각 Frame의 State Diagram



세부사항

- Physical memory size of user pool: 8,000KB (= 2,000 frames)
 - 30 ticks (1/2000분) 마다 교란되는 frame의 set이 바뀐다.
 - 한 시점에 2000개 frame 중 0 ~ 14개의 frame이 태양풍의 이온 입자에 의해 교란된다.
 - 교란되는 frame의 개수와 위치는 모두 임의로 결정된다. (규칙 없음)
 - 즉, 같은 frame이 연속으로 교란될 수 있다.
 - 교란된 frame에 cpu가 접근하게 되면 해누리호는 폭발한다.
 - 교란된 frame은 태양풍의 영향에서 벗어나면 즉시 사용 가능하나, frame의 내용은 쓰레기값이 된다.
- 특정 시점(t)에 교란되는(사용 불가능한) frame들을 미리 알 수 있다.
 - `int[] disturbed_frame_indices_at(tick t)`
 - 태양풍 이온입자에 의해 교란되는 frame의 index 리스트를 계산하여 return 하는 함수
 - 이 함수는 최대 30 ticks (1/2000분) 이내의 예측만을 정확하게 수행한다.
 - 먼 미래에 교란될 frame의 index는 알 수 없음 (널 포인터를 리턴함)
 - 여러분은 이 함수를 구현없이 사용할 수 있다.
- 디스크 접근은 오래 걸림
 - 메모리의 frame 1개를 디스크에 쓰는 데에는 1 tick (=1 밀리초) 소요
 - 디스크에서 frame 1개를 읽어들이며 메모리에 올리는 데에도 1 tick 소요
- 메모리 내 읽기/쓰기는 매우 빨라 거의 무시할 수 있음

답변 작성시 유의할 점

- (1) 창의적 시스템의 구현시에 유의해야 할 점을 고려해서 작성하세요
- (2) 디자인을 구체적으로, 상세하게 작성해 주세요.
 - 본인이 진행할 PintOS Project 3 VM 구현체에 기반해서 디자인을 작성해 주세요.
 - PintOS 구현 과정에서 새로 만든 Data Structure / Operation을 디자인 숙제 제출물에서 언급할때는 먼저 그것에 관한 설명을 해주세요.
 - e.g., frame table, supplemental page table, 추가한 함수들, ...
 - PintOS에서 가져오는 Data structure / operation이 있다면 이름과 목적/구조를 간략히 설명해주세요.
 - e.g., `void *palloc_get_page (enum palloc_flags);`
 - 한 문장 요약 (무슨 일을 하는 function이고, 어디에 쓰고자 하는지)
 - 디자인 제출안은 코드 수준까지 내려간 상세함을 갖추어야 합니다.