Project3 report

20160463 성해빈

구현한 함수

EduOM_CreateObject

Object 생성이다.
Object의 header를 초기화하고, 나머지 기능은 eduom_CreateObject에게 맡긴다.

eduom_CreateObject

기본적으로 이 함수가 복잡한 것은 object를 넣을 page를 찾는 과정에 있다.

처음에는 1차 페이지 선정을 한다.
nearObj가 NULL이 아닌 경우 nearObj과 같은 페이지로 선정하고,
NULL 인 경우에는 sm_CatOverlayForData 페이지의 정보를 이용해 availSpaceList를 탐색해가며 적절한 페이지를 선정한다.

그 다음에는 2차 페이지 선정을 한다.
이게 뭔 소리냐면 1차 페이지 선정때 고른 페이지가 여유 공간이 없으면 새로 페이지를 할당한다.

페이지 선정이 끝나면 object를 삽입한다. obj 내용을 복사하고, 헤더를 바꿔준다.

EduOM_DestroyObject

Object 삭제인데, 당연한 헤더 갱신과 함께 만약 삭제 후 page가 비어있고 file의 첫 page가 아니라면 page를 할당 해제하는 게 유의할 점이다.

EduOM_CompactPage

페이지 data를 차곡차곡 쌓는 느낌으로 생각하면 된다.
slot을 쭉 돌면서 object들을 순서대로 차곡차곡 복사해준다.
인자로 특정 slotNo가 있으면 걔만 특별취급하고 마지막으로 저장해준다.

EduOM_ReadObject

페이지 안에 있는 object data를 length만큼 읽는다.
length가 REMAINDER이면 object header에 있는 length까지 (object 끝까지) 다 읽는다.

EduOM_NextObject

일단 페이지를 찾아낸다. curOID가 NULL이면 file의 첫 페이지, 아니라면 curOID의 페이지다.

그 페이지에서 curOID의 slot 다음 번호부터(NULL이면 첫 slot부터) EMPTYSLOT이 아닌 object slot을 찾는다.
만약 그 페이지에 없다면 다음 페이지까지 쭉쭉 가고 파일 끝까지 가면 EOS를 반환한다.

EduOM_PrevObject

거의 EduOM_NextObject의 복붙인데, 방향이 반대인 걸 내 NextObject 코드 형식 상으로 표현하기 까다로워 좀 변형을 가했다.
nSlots의 정보는 GetTrain을 한 이후에 알 수 있다는 것이 코드를 그래도 같다붙이기에 까다로웠던 요인인 것 같다.
NextObject는 0부터 시작하면 되는데 PrevObject는 nSlots-1부터 시작해야 하니까.
조금의 hack으로 curOID를 매 루프 끝마다 NULL로 만들어주는 방식으로 페이지를 넘겨야 할때 이전 페이지의 마지막 slot을 보는 것을 구현했다.

느낀 점 discussion

sm_CatOverlayForData 가 object로 저장된다는 걸 좀 더 명시적으로 말해줬으면 좋았겠다!
정말 숨겨져있는 것처럼 매뉴얼에 조그맣게 나와있는데, 헤더 코드가 힌트를 그나마 얻어서 유추했던 것 같다.

edu에 쓰지 않는 원래 cosmos의 잔재라면 좀 지워줬으면 좋겠다.
이게 뭐지?? 하고 시간낭비하고 있다가 'educosmos에서는 사용하지 않음'을 보면 화가 난다.

project2가 cache 구현 같았다면, project3는 os의 heap space 구현 같았다.
딱 이렇게 size에 따른 free space list를 관리하고, object라고 하는 data structure로 실제 내가 아는 linux heap과 데이터구조가 매우 유사하다.
OS 코드도 아니면서 OS스러운 요소가 많다는게 데이터베이스 코딩의 매력인가 싶다.

어떻게 보면 C로 코딩하는 것에 대한 불편함인데,
getTrain을 해놓은 pid 변수의 내용을 바꾼 뒤 freeTrain에서 segfault가 나서 ???? 했던 때가 많았다.
pid는 그 페이지를 찾아주는 역할을 하기 때문에 당연히 바꾸면 어떤 페이지를 free하라는 건지 찾을 수 없다.
알고보면 쉽지만 진짜 이런 자잘한 실수를 하기 너무 쉽다.
이래서 C가 싫다.

중복을 싫어해서, 코드를 깔끔하게 만드려고 노력을 너무 많이 했다.
허나 C의 한계로 깔끔하게 하려다 포기한 부분도 많다.