

운영체제

-mybuddy 구현-

소프트웨어학과

201520860

노근탁

구조체정의 : chunk구조체에는 해당 chunk의 시작page와 order를 받는 변수와 tailq를 사용하기 위한 포인터 TAILQ_ENTRY를 포함시켰고 chunk_list에는 free list구현을위해 필요한 TAILQ의 header를 포함시켰으며 해당 freelist에 포함된 chunk의 수를 나타내기위한 변수를 선언했습니다. 마지막으로 buddy 구조체는 주어진 코드 그대로 구현하였습니다.

Init_buddy : chunk_list의 free list를 MAX_ORDER만큼 할당해준뒤 함수의 parameter로 전달받은 내가 allocator에서 사용할 page의 크기를 MAX_ORDER와 비교해서 같으면 해당 order에 맞는 freelist에 넣어주고 MAX_ORDER가 작으면 MAX_ORDER까지 나누어서 저장하여서 초기상태를 구현하였습니다.

Finibuddy : chunk list를 처음부터 끝까지 조회하여 해당 freelist에 들어가있는 chunk수를 나타내는 변수를 확인하여 0보다 크면 TAILQ_REMOVE를 사용 후 free를해주어서 memory leak이 나오지 않게 구현하였습니다.

Allocpage : parameter로 전달받은 order에 해당하는 freelist에 allocate가 가능한 chunk가있는지 조회하고 있으면 그대로 allocate해주고 없으면 해당 order의 위를 있을 때까지 조회하여서 있으면 그 chunk를 쪼개서 할당할 수 있도록 구현하였습니다. 그리고 할당되는 chunk의 첫번째 page를 parameter에있는 *Page에 저장하였습니다.

Freepage : free 할 chunk가 left인지 right인지 확인하고 각각 해당 order의 freelist에 buddy가 없으면 그냥 free하고 buddy가 있으면 합치고 order를 증가시키고 left 인지 right인지 판단하고 buddy를 찾고 있으면 합치고 없으면 free하는 과정을 buddy가 없거나 max_order까지 조회를 하거나 하도록 반복문을 사용하여 구현하였습니다.

Print_free_pages : parameter로 전달받은 order의 freelist에 chunk가있으면 TAILQ_remove로 꺼내서 주어진 형태에 맞게 출력 후 TAILQ_INSERT로 다시 넣어서 구현하였습니다.

Get_unusable_index : 전달받은 order가 0이라면 모든 크기의 chunk가 usable 하므로 0을 반환하였고 0이아니라면 전달받은 order전까지 조회하며 free list에 들어가있는 chunk의 숫자와 해당 chunk에 들어가는 page숫자를 곱하여 unusable한 page 숫자를 구하였고 그 값을 free한 page 총 숫자로 나누어서 반환하게 하였습니다.

고찰 : 해당 과제를 수행하면서 buddy allocator가 신경써야할 부분이 생각보다 많다는 것을 알았고 이에 대해 이해하는 기회가 되었던 것 같습니다. 그리고 기존의 난이도 때문에 어려움이 많았는데 이에 피드백 해주셔서 감사드립니다.