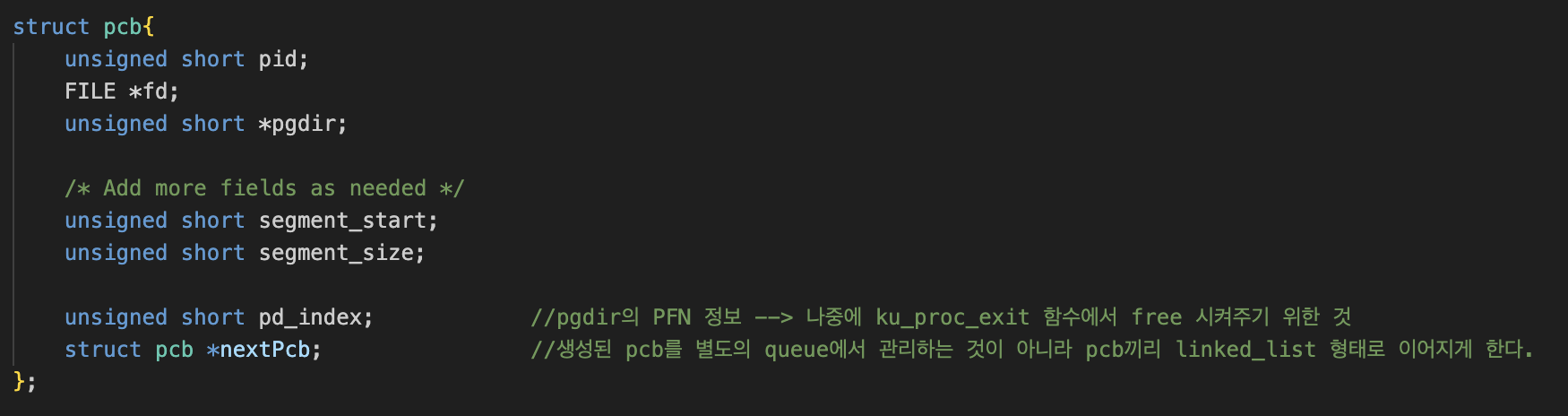
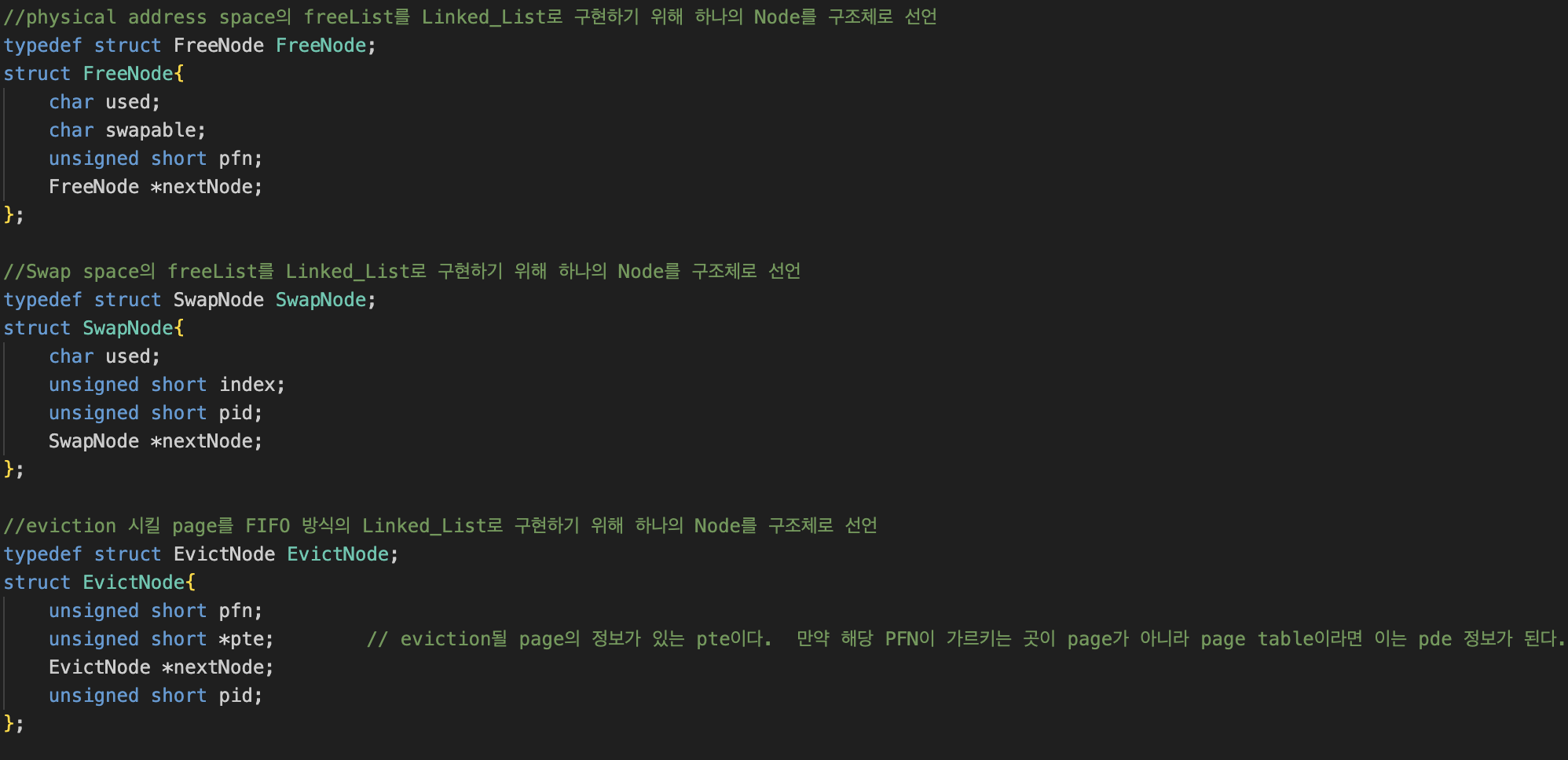
2024 운영체제 Assignment #1 (KUMOO)

컴퓨터공학부 202011253 권민혁

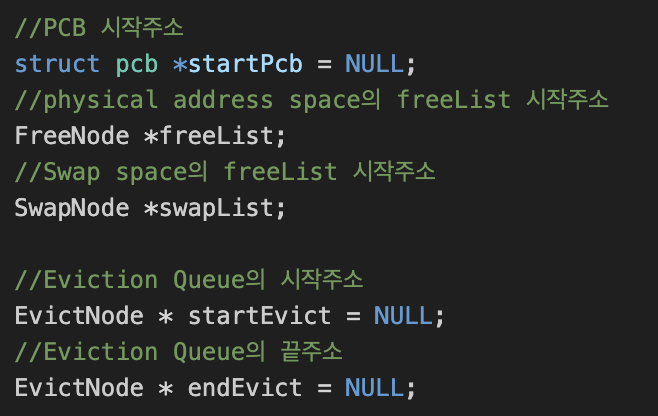
**Design and implementation**

Pcb 정보입니다. 주어진 코드에서 추가적으로 execuable 파일에서 읽어온 segment의 start와 size 정보를 가지고 있도록 구현했습니다. 이를 통해 만약 유효하지 않은 공간으로 접근하려고 하면 이를 막기 위한 정보입니다.

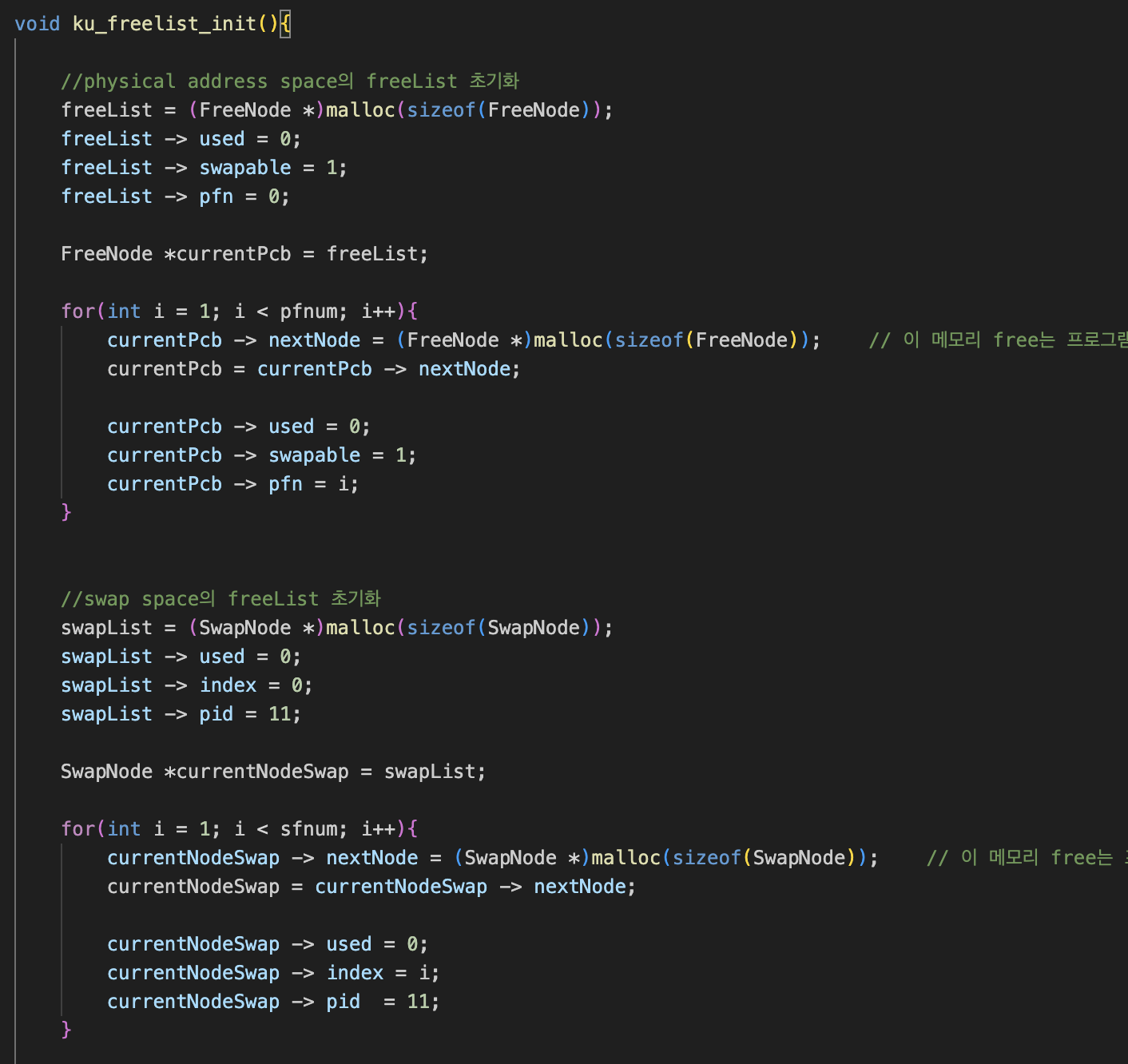
Pd\_index는 해당 process의 page directory가 존재하는 PFN을 의미합니다. pgdir의 정보만으로 PFN을 알아내기 번거로울 것 같아 해당 정보도 pdb에서 같이 관리하도록 하였습니다. 그리고 생성된 pcb 들이 서로 linked list 형태로 이어져 있도록 구현했습니다.



Linked List 를 구현하기 위한 각 node 정보를 구조체로 구현하였습니다. FreeNode는 물리 메모리에 대한 page를 관리하기 위한 linked list의 node 구조체입니다. 필드 값으로 used 를 두어서 free 공간인지 할당되었는지를 판단합니다. SwapNode는 swap space에 대한 page를 관리하기 위한 linked list의 node 구조체입니다. 마찬가지로 used 로 free인지 판단합니다. EvictNode는 eviction을 위한 queue를 관리하기 위해서 만든 node 구조체 입니다.

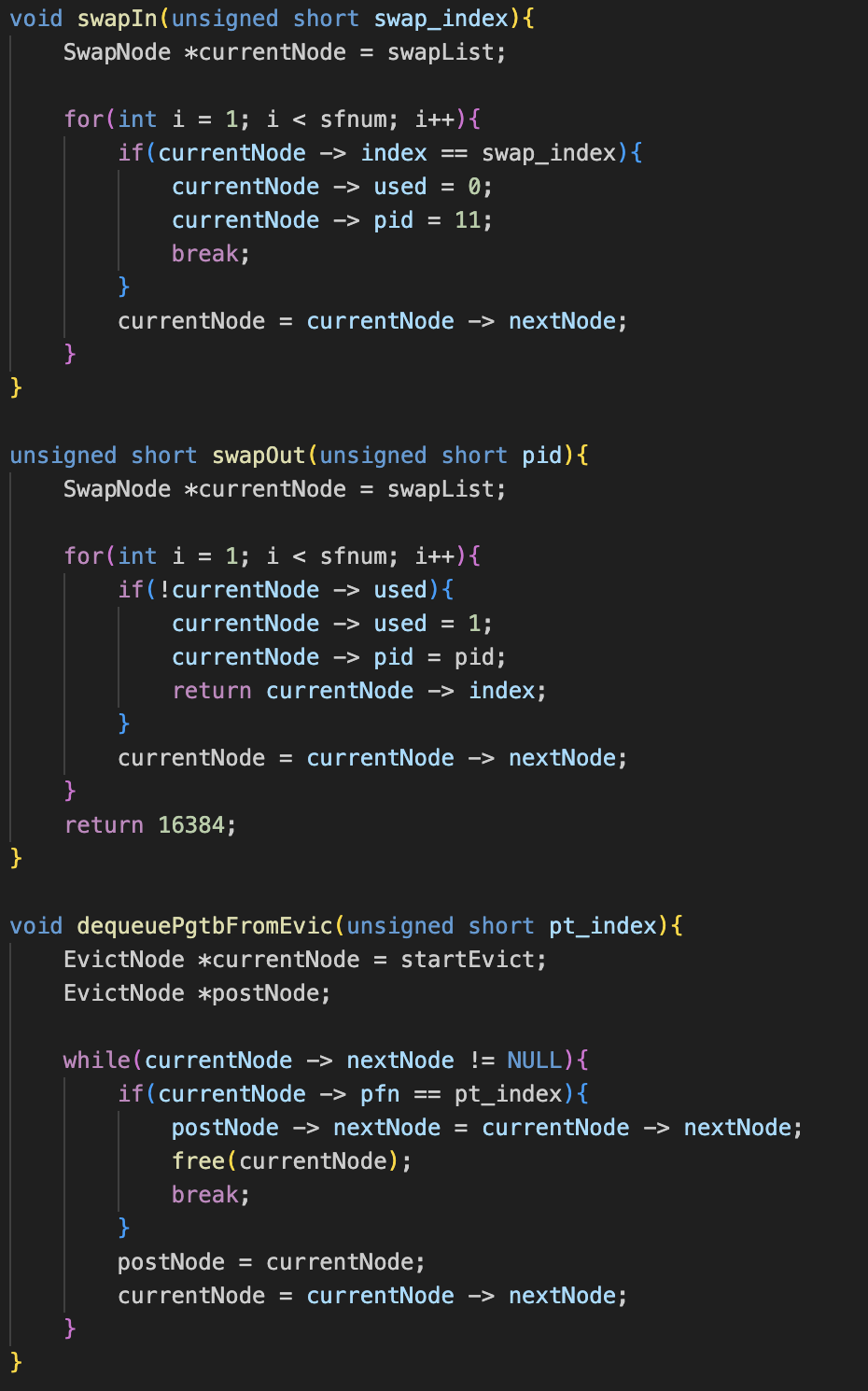


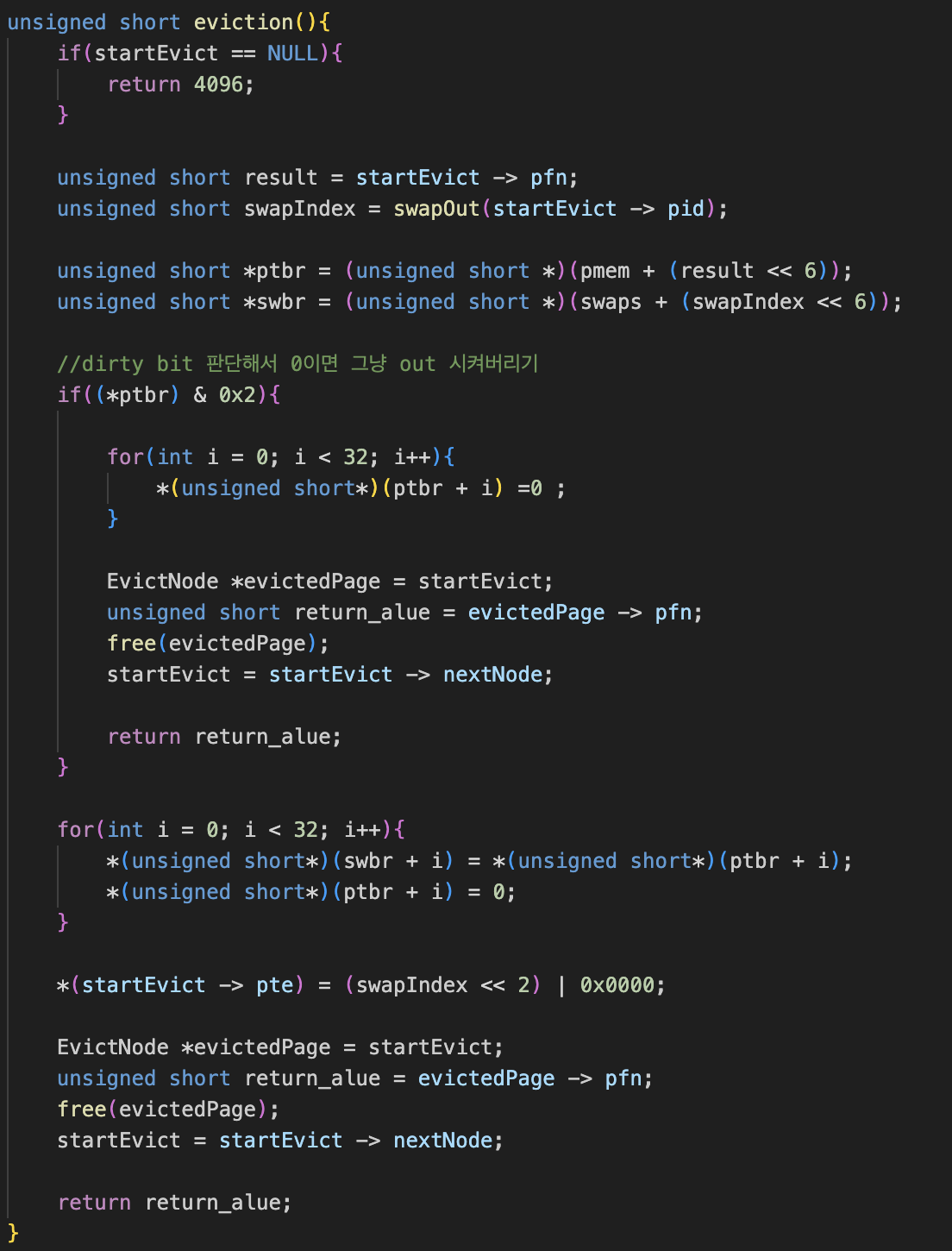
이처럼 전역 변수로 각 linked list의 시작 지점 포인터 변수를 선언했습니다.

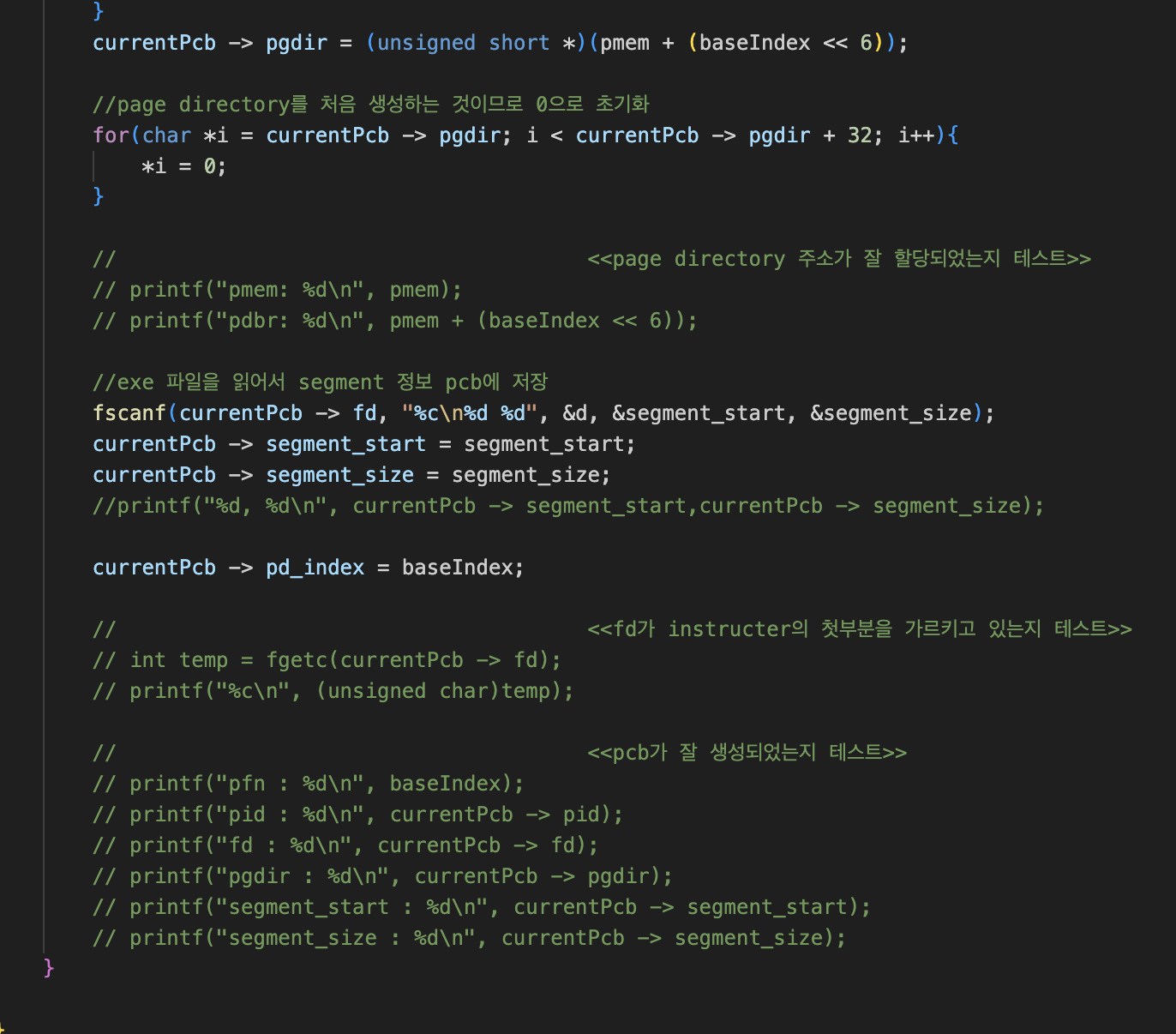
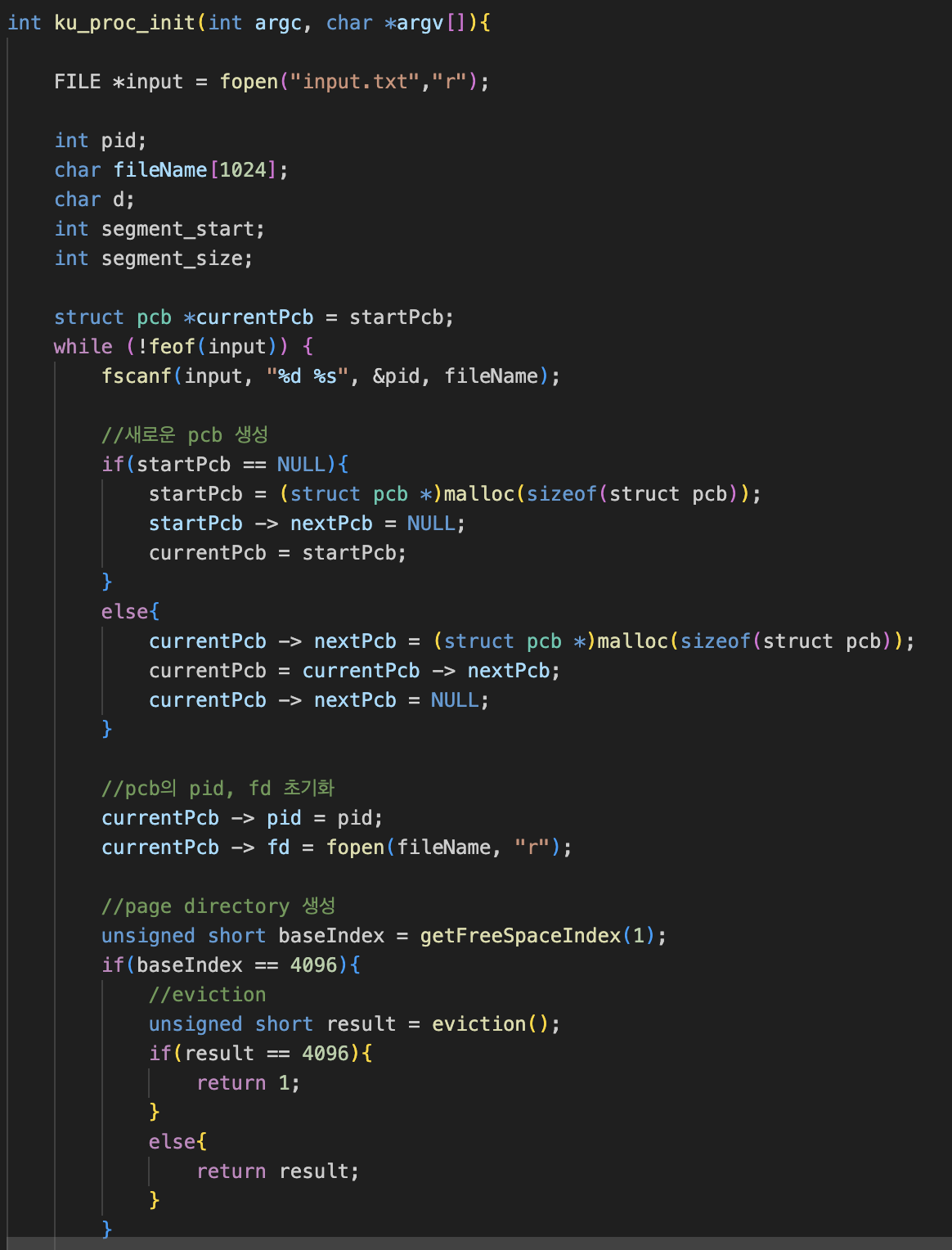


Free list를 생성하는 함수입니다. 계속 말했듯이 각 free list는 linked list 형식으로 구현했습니다. freeNode는 물리 메모리 공간의 free space 공간을 관리하는 것이므로 node의 개수를 pfnum으로 했습니다. 마찬가지로 swapNode의 개수도 sfnum입니다.

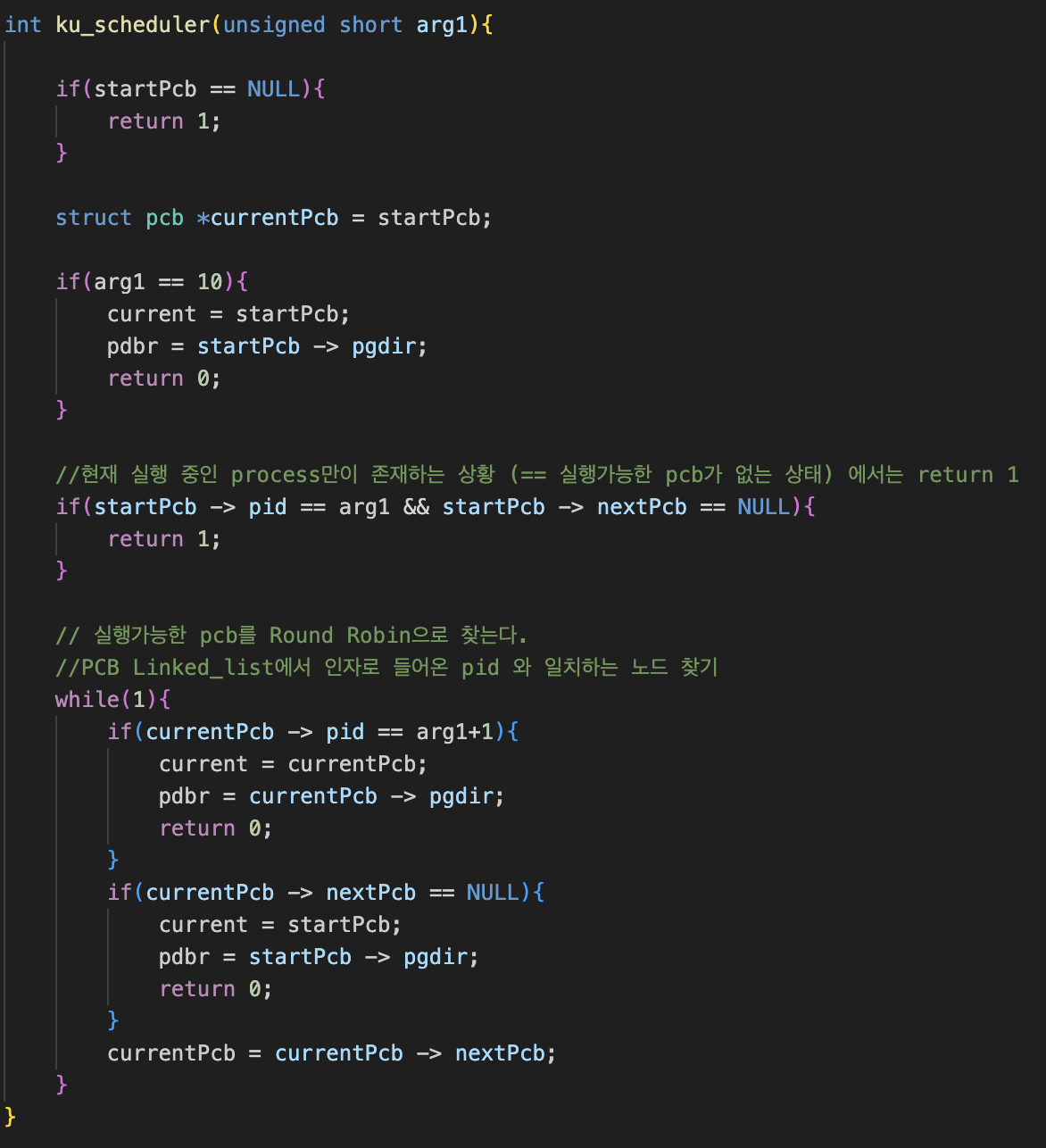
물리 메모리 공간 중에서 free space를 할당받고, 리턴하는 함수입니다. Free space를 탐색하는 방법으로는 sequential하게 탐색하도록 구현했습니다. Linked list를 sequential하게 탐색합니다.

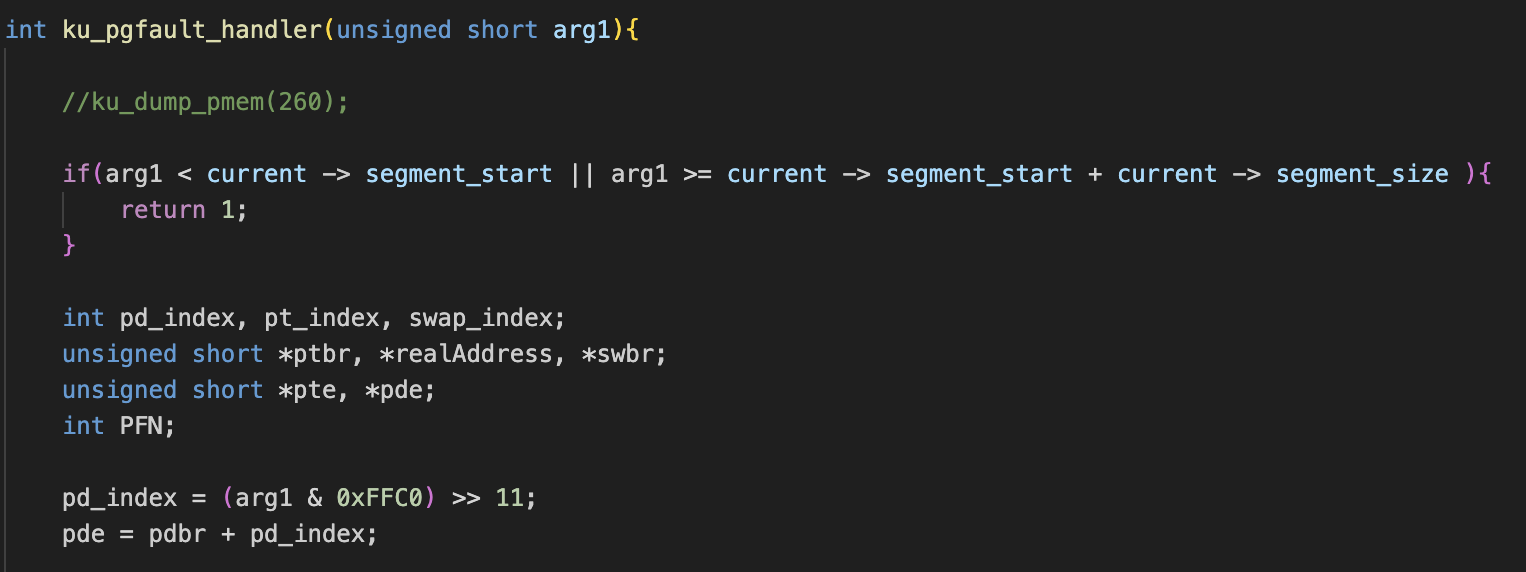
마찬가지로 swap space 공간에서 free space를 할당받기 위한 swap out 함수와 할당받은 공간을 리턴하기 위한 swapIn 함수입니다. 이또한 물리 메모리 공간과 마찬가지로 swquential 하게 linked list 공간을 탐색합니다. DequeuePgtbFromEvic 함수는 pte가 하나라도 유효한 page table인 경우에는 evict 되면 안되므로 큐에서 해당 node를 빼는 함수입니다. 이또한 sequential 하게 큐를 탐색하다가 인자로 들어온 pfn 과 같은 노드를 만나면 제거합니다.

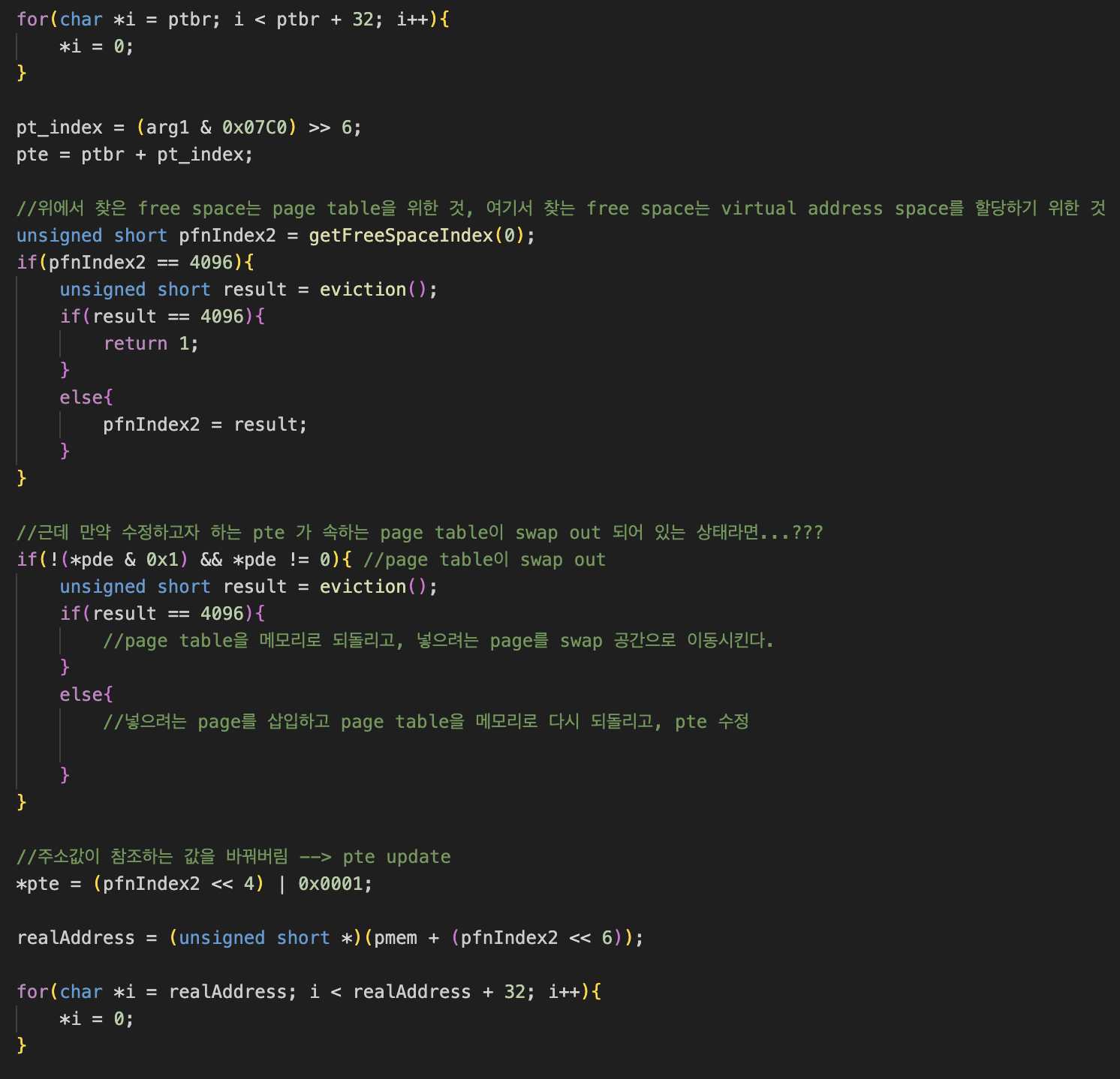
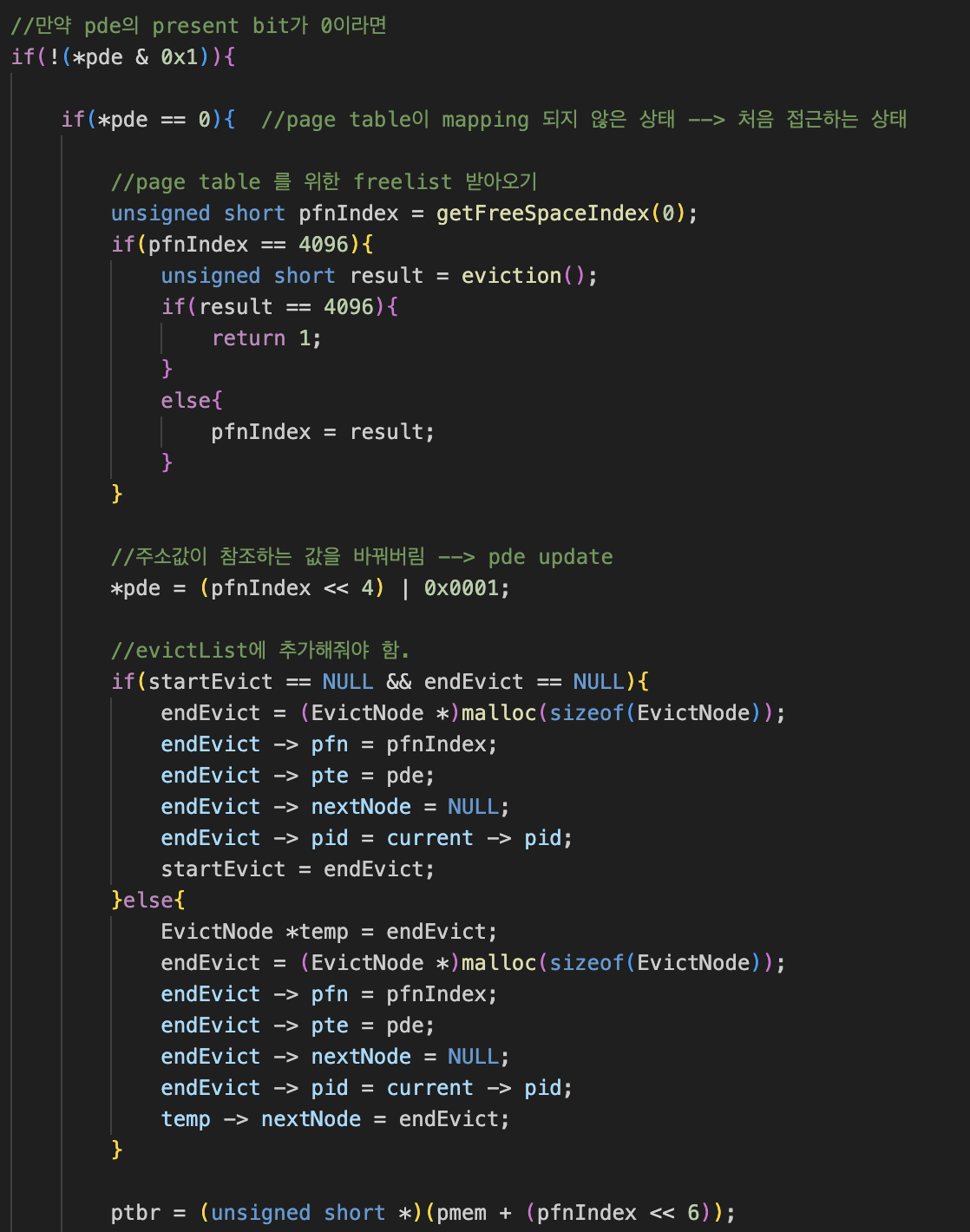
만약 물리 메모리 공간을 할당 받기 위해서 getFreeSpaceIndex 함수를 호출했는데 모든 물리 메모리 공간의 페이지가 할당되어 있는 상태라면 eviction 큐에서 하나를 선택해서 evict 시켜서 swap out 시킨 다음에 해당 page를 swap out 된 공간에 할당해야 합니다. 이를 위한 함수로 evict 큐에서 FIFO 형식으로 evict 할 노드를 찾아서 swap 공간에 swap out 시키고, 해당 공간의 pfn을 반환하는 함수입니다. Evict 시키고 나면 evict 큐를 update 시켜주기도 합니다.

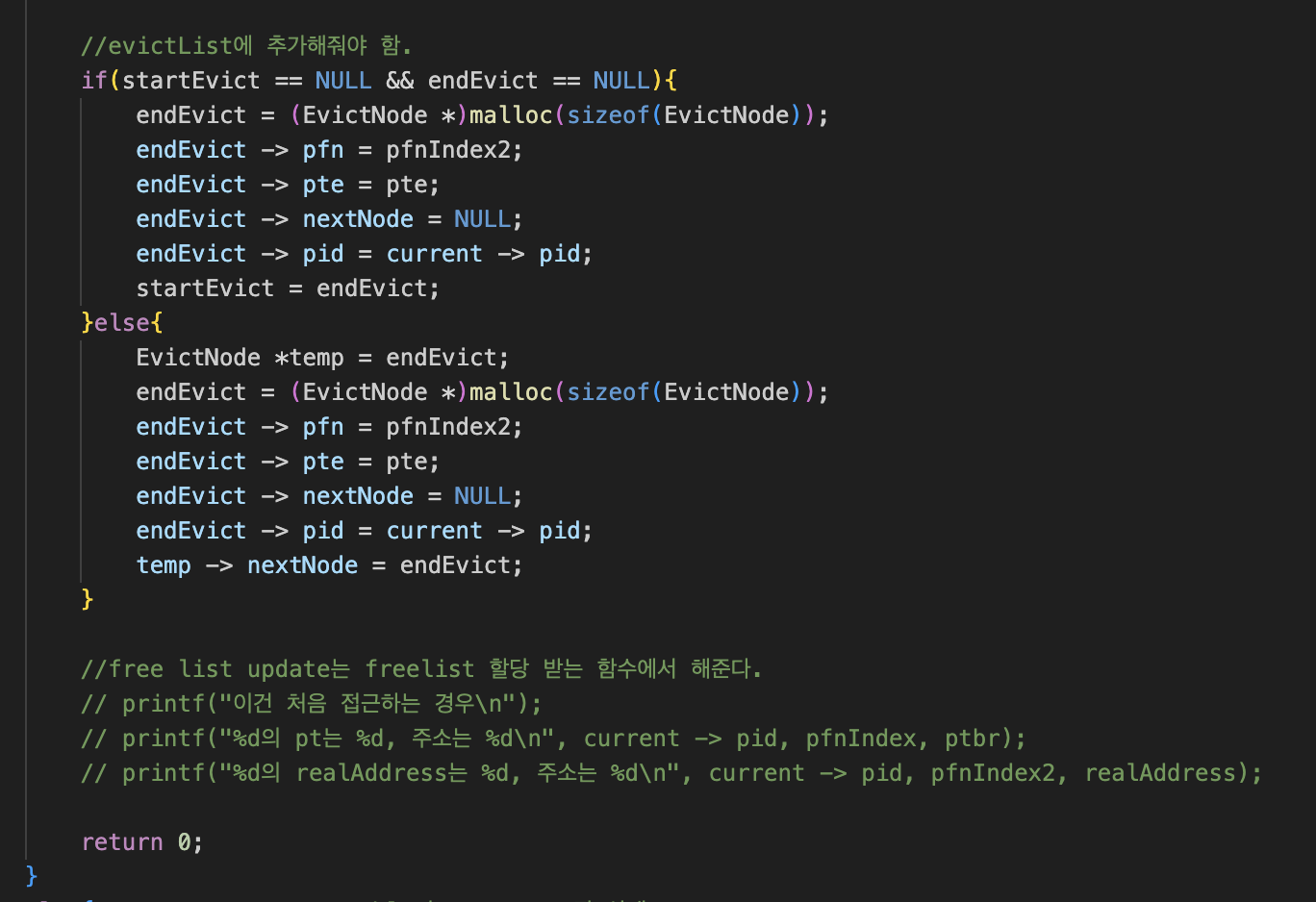


컴퓨터가 처음 시작했을 때 불려지는 함수로, 실행할 모든 process 에 대해서 pcb를 생성해주는 작업을 수행합니다.생성된 pcb는 linked list 형태로 관리되며, pcb 가 생성되면서 해당 process 의 page directory도 생성됩니다.해당 공간은 eviction의 대상이 되지 않으므로 eviction 큐에 넣지 않습니다.

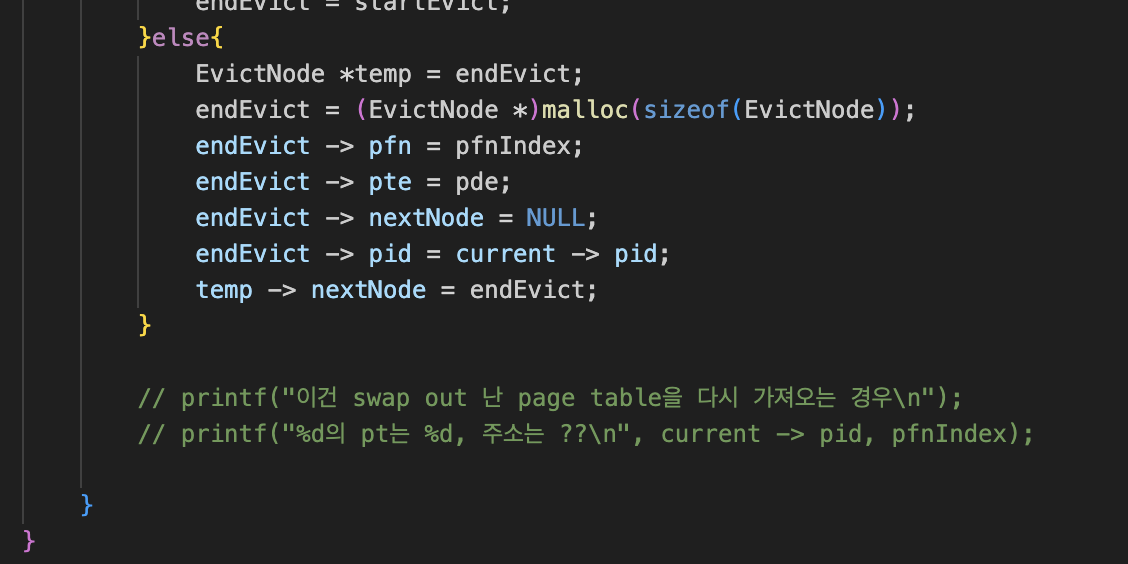
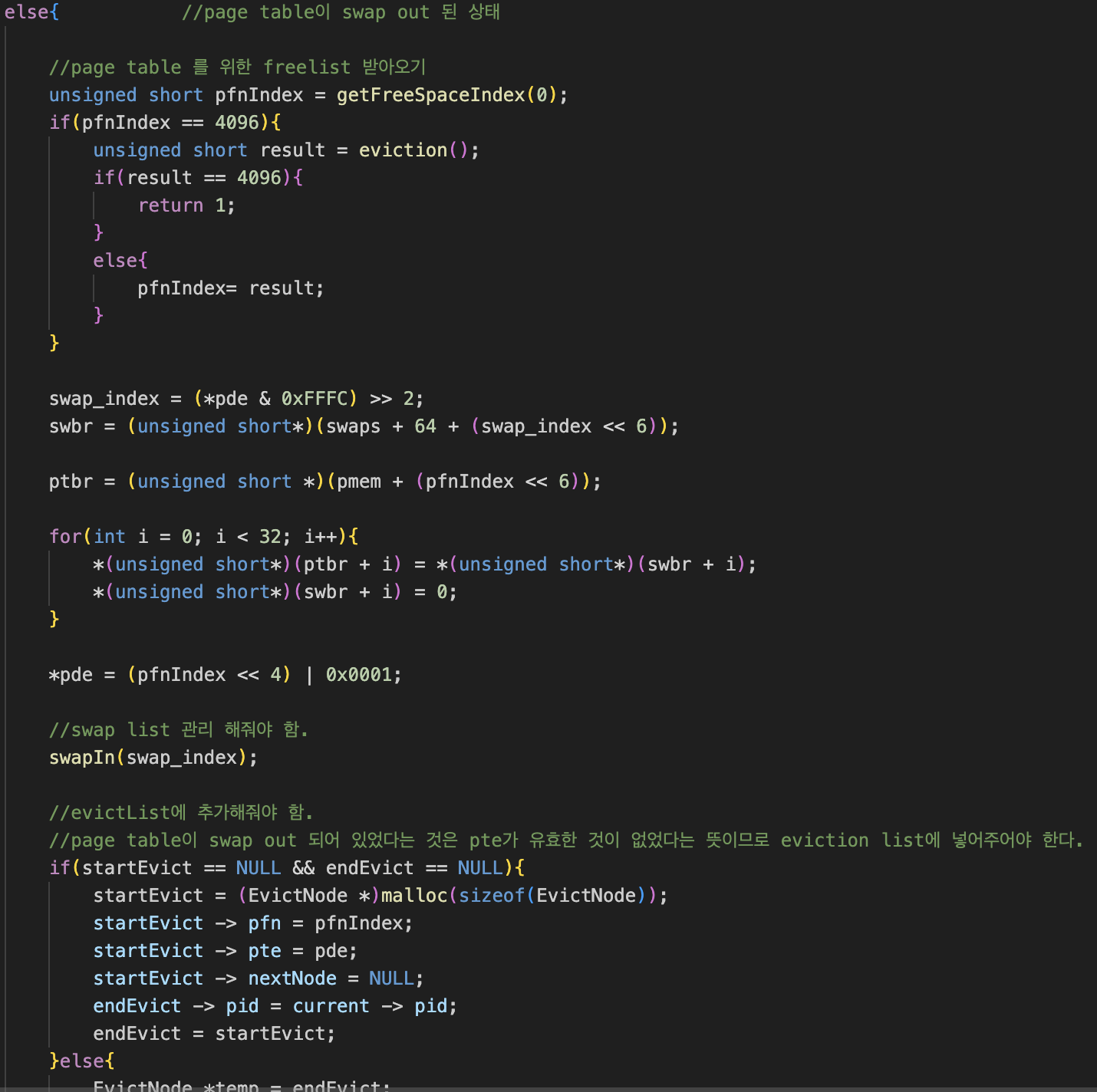
cpu가 다음 실행시킬 process를 결정하기 위한 정책이 담긴 scheduler 입니다. RR 정책으로 구현되어 있고, ku\_proc\_init 함수에서 생성한 pcb linked list 를 탐색하면서 다음 실행할 Process 로 context switching 작업을 수행합니다. Context switching은 전역변수로 선언된 current 포인터 변수와 pdbr의 주소 값을 바꾸고자 하는 process 에 해당하는 값으로 바꾸어 주는 것을 의미합니다. 인자로 현재 실행중인 process 의 pid가 들어오는데, pcb linked list를 sequential하게 탐색하면서 그 다음 node를 다음으로 실행할 process 로 여깁니다. 만약 현재 실행중인 process가 pcb linked list의 마지막 node에 해당한다면 처음 node로 바꾸어 줍니다.

Page fault가 발생했을 때 실행되는 handler 함수 입니다. 우선 유효한 접근인지 판단을 하고, page directory의 주소를 계산합니다.

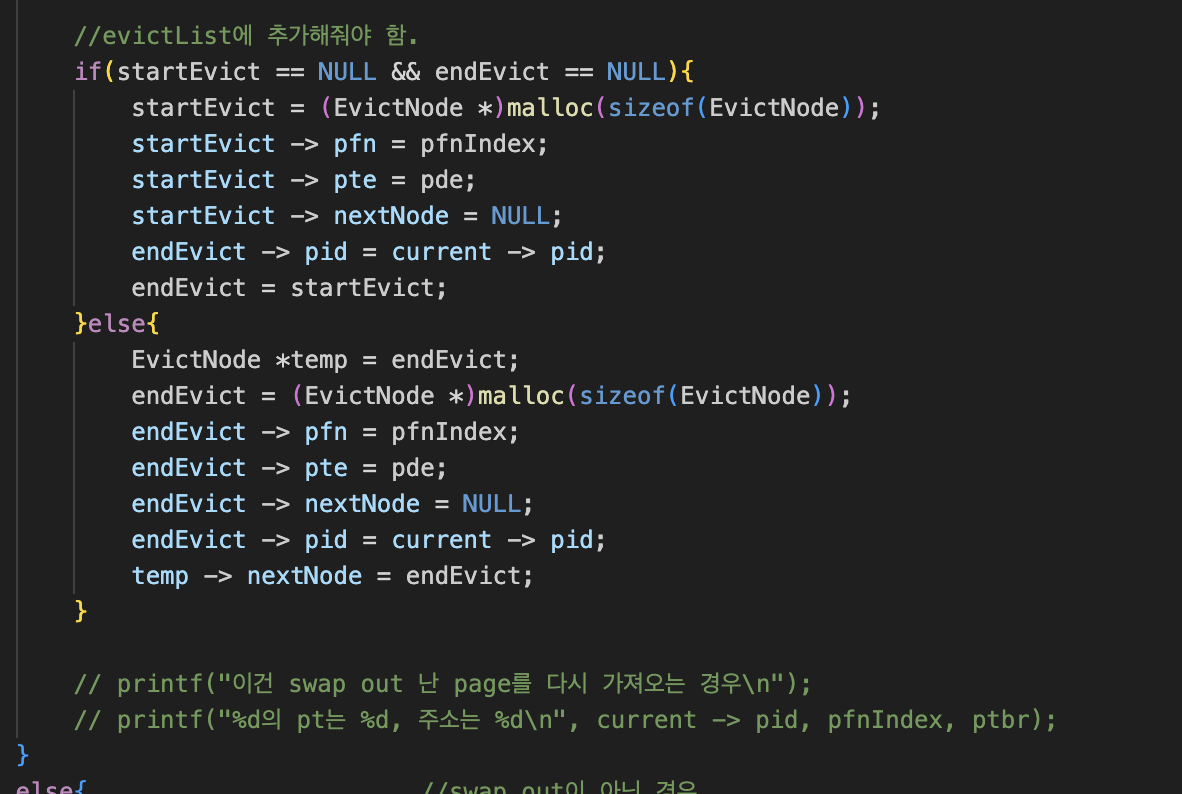
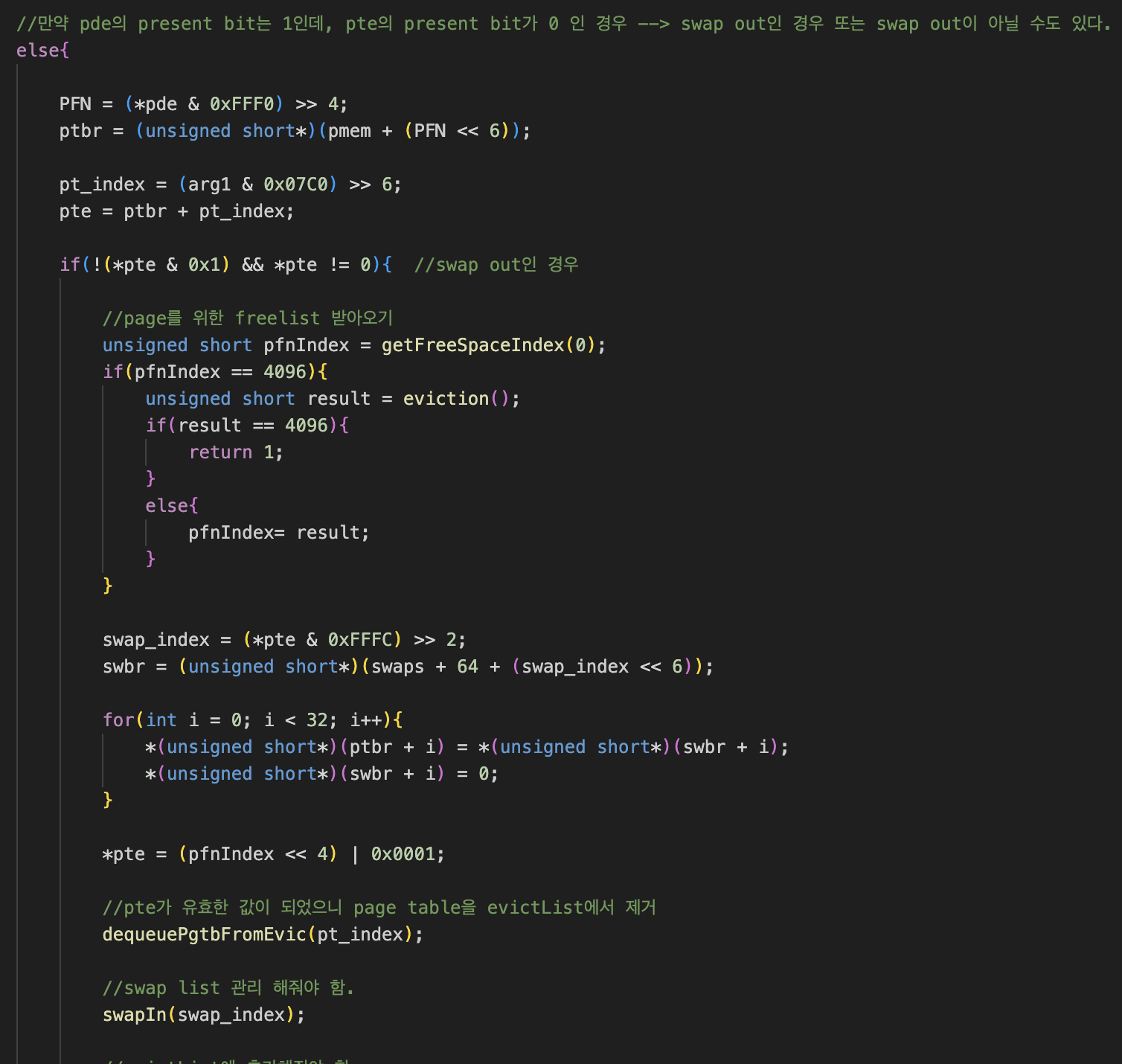




만약 pde의 present bit이 0이라면 page table도 mapping이 되지 않은 상태인 것입니다. Page table을 mapping 하기 위해서 free space를 우선 할당 받습니다. 가능한 free space가 존재하지 않는다면 위에서 만든 eviction 함수를 호출해서 해당 page 주소를 받습니다.. 그리고 page table이 할당 되었으면 pde의 값을 update 시켜줍니다.. Page table이 mapping 되었고 해당 page table의 모든 pte가 아직 유효하지 않은 값이기 때문에 해당 page table도 eviction 의 대상이 됩니다. 그래서 eviction 큐에 해당 node를 마지막에 추가합니다. (eviction 은 FIFO 정책을 따르기 때문이다.) 그리고 page table의 pte가 가르키는 실제 segment에 대한 mapping 도 해주어야 하므로 해당 부분도 page table 과 동일한 과정으로 진행합니다.



만약 pde가 0이 아니면 page table이 swap out 되어 있는 상태라는 것입니다. 그래서 page table을 swap in 해주고 위와 똑같은 과정을 진행합니다.



만약 pde의 present bit이 1인데 pte의 present bit이 0이면서 pte가 0 이 아니라면 pte가 가르키는 page가 swap out 되었다는 소리이므로 해당 page를 swap in 시키고, pte를 수정합니다. 만약 만약 pde의 present bit이 1인데 pte가 0이면 해당 page는 mapping이 된 적 없다는 뜻이므로 free space에 해당 page를 mapping 시켜 주면 됩니다.



마지막으로 종료된 process의 pcb, 물리 공간, swap 공간을 리턴하는 함수입니다.

**Function Description**

|  |  |  |
| --- | --- | --- |
| Ku\_freelist\_init | Description | 프로그램이 (OS가) 시작되고 바로 불려지는 함수로, 물리공간, 가상공간에 대해 free space를 관리하는 linked list를 생성합니다. 각 노드 별로 used 필드 값이 있어서 해당 page 공간에 값이 있는지 없는지를 나타냅니다. |
| Parameters | Void |
| Return value | void |

|  |  |  |
| --- | --- | --- |
| getFreeSpaceIndex | Description | 물리 메모리 공간에서 free space의 PFN을 반환하는 함수입니다. Free space를 관리하기 위한 linked list를 sequential하게 탐색하면서 가장 처음 발견한 used가 0인 노드의 PFN을 반환합니다. |
| Parameters | Int isPD  해당 요청이 지금 page directory를 위한 요청인지를 판단하고 맞다면 노드의 swapable을 0으로 세팅합니다. |
| Return value | Free space 공간의 PFN을 반환합니다. |

|  |  |  |
| --- | --- | --- |
| returnFreeSpaceIndex | Description | 할당되었던 space를 다시 free list로 만들어주는 함수입니다. Linked list 를 sequential 하게 탐색하다가 인자로 들어온 pfn과 동일한 노드가 있으면 해당 노드의 used를 0으로 세팅합니다. |
| Parameters | 할당이 풀리는 space의 PFN |
| Return value | void |

|  |  |  |
| --- | --- | --- |
| swapIn | Description | 할당되었던 space를 다시 free list로 만들어주는 함수입니다. Linked list 를 sequential 하게 탐색하다가 인자로 들어온 index과 동일한 노드가 있으면 해당 노드의 used를 0으로 세팅합니다. |
| Parameters | 할당이 풀리는 space의 index |
| Return value | void |

|  |  |  |
| --- | --- | --- |
| swapOut | Description | swap 공간에서 free space의 index를 반환하는 함수입니다. Free space를 관리하기 위한 linked list를 sequential하게 탐색하면서 가장 처음 발견한 used가 0인 노드의 index를 반환합니다. |
| Parameters | 실행 중인 process 의 pid |
| Return value | Swap Free space의 index를 반환합니다. |

|  |  |  |
| --- | --- | --- |
| dequeuePgtbFromEvic | Description | Page table에 유효한 pte가 하나라도 있으면 eviction 큐에서 page table을 제거하기 위한 함수입니다. |
| Parameters | eviction 큐에서 제거하고자 하는 PFN이 들어옵니다. |
| Return value | void |

|  |  |  |
| --- | --- | --- |
| eviction | Description | 현재 eviction 큐에서 가장 앞에 해당하는 Page를 swap out 시킨 뒤, 해당 page 가 존재했던 PFN을 반환하는 함수입니다. |
| Parameters | void |
| Return value | Eviction 된 page가 있던 PFN |

|  |  |  |
| --- | --- | --- |
| Ku\_proc\_init | Description | Input.txt 파일을 읽어서 pcb와 page directort를 생성하는 함수입니다. |
| Parameters | Argc, argv  딱히 쓰이지 않습니다. |
| Return value | 정상적으로 성공하면 0, 실패하면 1 |

|  |  |  |
| --- | --- | --- |
| Ku\_scheduler | Description | cpu가 다음에 어떤 process 를 실행할지 pcb linked list를 탐색하면서 지정해주는 함수입니다. |
| Parameters | 현재 실행 중인 process의 pid |
| Return value | 정상적으로 성공하면 0, 실패하거나 예외가 발생하면 1 |

|  |  |  |
| --- | --- | --- |
| Ku\_pgfault\_handler | Description | Va 를 pa 로 변환하는 과정에서 page fault 가 발생했을 시 불려지는 함수로, 경우의 수를 4가지로 나눕니다. Pde의 present 가 0, pde의 present가 0이지만 pde가 0이 아닌 경우(page table이 swap out 된 경우), pte의 present 가 0인 경우, pte의 present가 0이지만 pte가 0이 아닌 경우입니다. 각각 해당하는 로직이 구현되어 있습니다. |
| Parameters | 현재 page fault 가 발생한 va |
| Return value | 정상적으로 성공하면 0, 실패하거나 예외가 발생하면 1 |

|  |  |  |
| --- | --- | --- |
| Ku\_proc\_exit | Description | Process 가 하나 종료할 때마다 불려지는 함수입니다. 해당 process 에게 할당되어 있던 pcb, 물리 메모리 공간, swap 공간을 다시 free space로 되돌려 놓는 작업을 합니다. |
| Parameters | 종료된 process의 pid |
| Return value | 정상적으로 성공하면 0, 실패하거나 예외가 발생하면 1 |