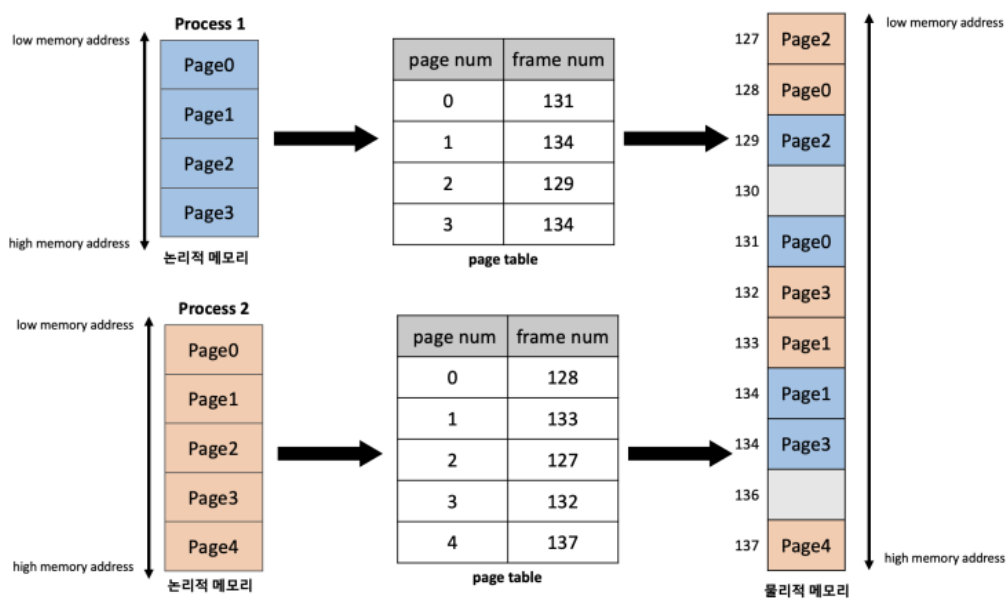


### ● paging이란 뭔가?

▶ paging 기법은 process가 할당받은 메모리 공간을 동일한 크기의 page 단위로 나누어, 물리적 메모리에서 연속되지 않은 서로 다른 위치에 page들을 저장하는 메모리 관리 기법입니다. paging 기법에서는 물리적 메모리를 page와 같은 크기의 frame으로 미리 나누어둡니다. page 단위로 메모리 적재가 이뤄지기 때문에 미리 분할해두면 빠르게 메모리 할당이 이뤄질 수 있기 때문입니다. 또한 paging 기법에서는 주소 바인딩(address binding)을 위해 모든 프로세스가 각각의 주소 변환을 위한 page table을 갖습니다. 하나의 프로세스 내에서도 페이지 단위로 다른 물리적 메모리에 저장되기 때문에 주소 바인딩을 위해서는 별도의 page table이 필요한 것입니다. 주소 바인딩(address binding)이란 CPU가 기계어 명령을 수행하기 위해 process의 논리적 주소가 실제 물리적 메모리의 어느 위치에 매핑되는지 확인하는 과정입니다.



### ● 논리적 주소, 물리적 주소, 주소 바인딩

▶ 논리적 주소(logical address)란 process가 memory에 적재되기 위한 독자적인 주소 공간으로 각 process마다 독립적으로 할당되며, 0번지부터 시작됩니다. 물리적 주소(physical address)란 process가 실제로 RAM 메모리에 적재되는 위치를 말합니다. 주소 바인딩(address binding)이란 CPU가 기계어 명령을 수행하기 위해 process의 논리적 주소가 실제 물리적 메모리의 어느 위치에 매핑되는지 확인하는 과정입니다.

### ● paging 기법 사용시 발생할 수 있는 메모리 단편화(Memory fragmentation) 문제에 대해 설명해라

▶ 물리적 메모리 공간이 작은 조각으로 나뉘어져서 메모리가 충분히 존재함에도 할당이 불가능한 상태를 보고 메모리 단편화가 발생했다고 말합니다. paging 기법에서는 process의 논리적 주소 공간과 물리적 메모리가 같은 크기의 page 단위로 나누어지기 때문에 외부 단편화 문제는 발생하지 않습니다. 외부 단편화 문제란 메모리상의 비어있는 공간의 크기가 작아서, 빈 메모리 공간임에도 활용되지 못하는 문제입니다. 하지만 process 주소 공간의 크기가 page 크기의 배수라는 보장이 없기 때문에, 프로세스의 주소 공간 중 가장 마지막에 위치한 page에서는 내부 단편화 문제가 발생할 가능성이 있습니다. 내부 단편화 문제란 프로세스가 필요한 양보다 더 큰 메모리가 할당되어서 메모리 공간이 낭비되는 상황입니다.