**[9] C/C++ 언어에서 하드웨어를 제어하기 위하여 개발된 기능 두가지가 무엇인지 나열하고, 해당 기능이 왜 하드웨어를 제어할 수 있다는 것인지 설명합니다. Note3, 좀더 보완 필요**

Physical memory management

포인터(pointer)는 물리적인 컴퓨터 메모리의 주소값을 저장할 수 있는 변수이다. 포인터도 컴퓨터 메모리의 바이트를 잡아먹는 변수이다. 위치값을 정확하게 저장할 수 있다. 4기가바이트 램에서 0바이트부터 4가가번째 -1까지 위치값을 정확하게 지정할 수 있다. 그렇기에 컴퓨터 하드웨어의 메모리에 직접 접근하고 다룰 수 있어 하드웨어를 제어할 수 있는 것이다. 그렇기에 피지컬한 컴퓨터 메모리를 실제로 가리킬 수 있기 때문에 하드웨어에 손상을 줄 수 있어 위험하다.

Bit operation

비트연산을 할 수 있기 때문이다. 1바이트를 8비트 처리하고, &, <(꺽쇠), 등의 기호를 사용한다. 기본적인 데이터의 형태는 0과 1비트 이기 때문에 하드웨어를 제어하고 대화를 나눌려면 비트연산을 해야하기에 비트연산이 하드웨어를 제어할 수 있는 것이다. 그런데 비트 연산 같은 경우에는 C/C++뿐만 아니라 다른 언어들에서도 가능하다.

**수민이가 조사한 것**

(1) Hardware device가 컴퓨터 메모리를 통해서 CPU와 정보를 주고 받는 memory-mapped I/O가 가능합니다. Memory-mapped I/O는 주변 장치를 메모리 주소 공간에 매핑하여 CPU가 메모리와 마찬가지로 주변 장치에 액세스할 수 있도록 하는 것입니다. 각 주변 장치는 메모리 주소를 할당받고, CPU는 주소를 통해 주변 장치와 통신합니다. 이로써 주변 장치를 제어하고 데이터를 주고받을 수 있습니다. (2) C/C++에서 포인터는 메모리 주소를 저장하고 해당 주소에 직접 접근할 수 있는 변수입니다. 따라서 C/C++ 포인터를 사용하면 Memory-mapped I/O에서 특정 주변 장치 레지스터를 가리키고 해당 레지스터 값을 읽거나 쓸 수 있습니다.