**Assignment 1**

(GPU 컴퓨팅)

담당교수: 공영호 교수님

학과: 컴퓨터정보공학부

학년: 3

학번: 2018202074

이름: 김상우

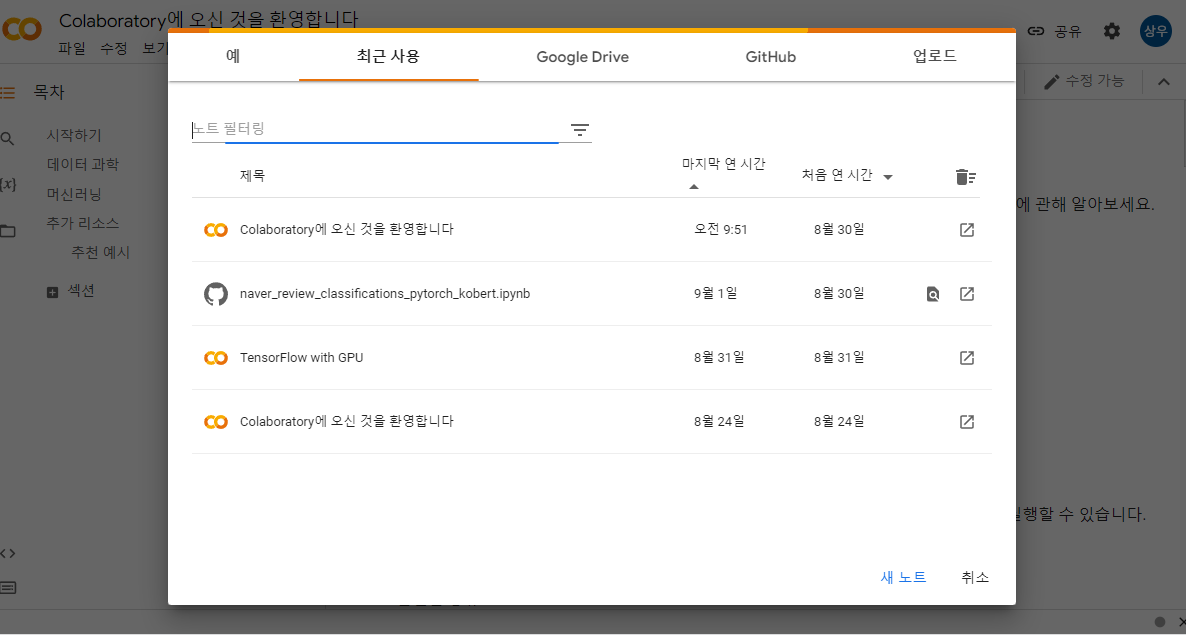
#Google Colab SetUp

과제를 들어가기 앞서, Google Colab 환경에서 진행하기 위해 다음과 같이 Google Colab 사이트에 접속하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 내부에 들어가 새 노트를 작성하여 과제를 할 수 있는 환경을 조성하였습니다.

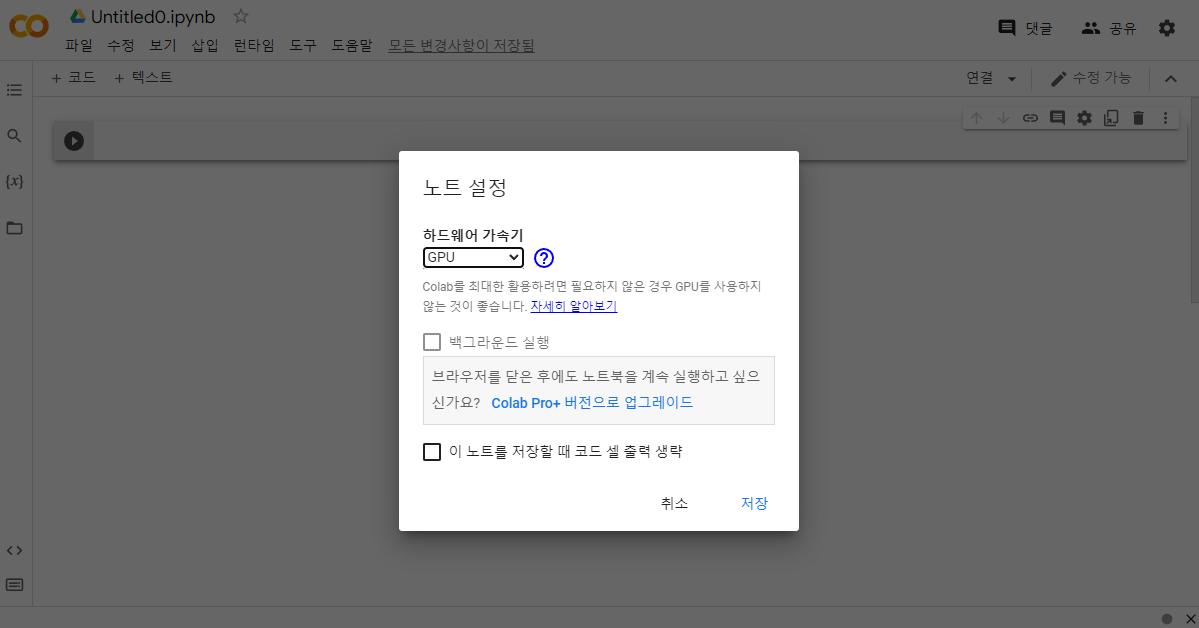


성공적으로 Untitled0.ipynb 노트가 생긴 것을 확인할 수 있습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 런타임- 런타임 유형 변경에 들어간 후 하드웨어 가속기를 GPU로 설정해주었습니다.



정상적인 작동을 확인하기 위해 명령어를 입력, NVCC 설치 확인을 진행하였습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

#Code 작성

텍스트이(가) 표시된 사진

자동 생성된 설명

위는 해당 과제에서 제시되고 Google Colab 환경에서 실행되어야 하는 내용을 담은 cpp 코드이다. 위 코드를 확인하면 void test함수는 a배열의 i번째 내용에 1을 더해 b배열 내의 i번째에 넣는 함수이다. Main에선 a와 b에 int가 SIZE, 즉 10개 만큼 들어갈 수 있는 만큼 Malloc을 통해 메모리를 할당될 것이며, 반복문을 통해 a엔 0,1,2…,9가 들어가게 될것이고, b엔 0만이 들어가게 될 것이다. 이후 test함수를 통해 b에 a값을 기반으로 1,2,3,…,10이 들어가며 이를 모두 출력하게 될 것이다. 출력 후, free 명령어를 통해 할당된 메모리를 제거해주고 종료합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

%%writefile을 통해 colab 환경에서 Assignment1cuda.cu파일로서 다룰 수 있도록 한다.

SIZE를 10으로 지정하고 test함수도 동일하게 적용하였습니다.(i는 현재 thread의 x좌표를 합니다.) 이후 host에서 사용할 a,b와 device에서 사용할 d\_a와 d\_b를 선언하고 int가 SIZE만큼 들어갈 수 있도록 메모리를 할당합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 cudaMalloc을 이용, d\_a와 d\_b에 cuda에서 사용할 수 있도록 cudaMalloc을 통해 할당합니다. 인자로서 Device에서 사용할 변수(d\_a, d\_b)와 할당될 사이즈(int 사이즈 \* 배열 수)를 기재하였습니다. 이후 반복문을 통해 a에 0,1,2…,9를 b에는 모두 0을 넣어주었습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 cudaMemcpy를 이용, d\_a에 a를, d\_b를 sizeof(int) \* SIZE만큼 복사합니다. 이때 CPU에서 GPU, 즉 Host에서 Device로 복사하는 것이므로 3번째 인자는 cudaMemcpyHostToDevice가 됩니다. 이후 test <<<1, SIZE >>>(d\_a ,d\_b)를 이용 1부터 SIZE까지 d\_a와 d\_b에 대하여 test함수를 실행시킵니다. 해당 결과로 d\_b에는 d\_a의 값에 1을 더한 1,2,3…,10이 들어가게 될것입니다. 이후 cudaMemcpy를 통해 b에 d\_b의 내용을 sizeof(int) \* SIZE만큼 복사합니다. 이때는 GPU에서 CPU로, 즉 Device에서 HOST로 복사하므로 cudaMemcpyDeviceToHost가 인자로 들어가게 됩니다.

이후 printf로 b의 모든 내용을 출력해줍니다. 이후 할당된 a와 b의 메모리를 해제하고 cudaFree를 이용해 d\_a와 d\_b에 할당된 메모리 또한 해제해주고 함수를 종료합니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

위는 cpp파일의 실행모습이다. 예상대로라면 google colab에서도 같은 결과가 나올 것이다.

#실제 적용

텍스트이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

위에서 작성된 코드를 적용하였다. 맨 위에 적힌 코드에 의해 Assignment1cuda.cu가 생기게 된다.

테이블이(가) 표시된 사진

자동 생성된 설명

이후 nvcc -o를 이용 cuda capability를 sm\_35 버전으로 컴파일 할 수 있도록 실행파일을 생성하며 이때 이름을 Assignment1cuda로 지정해주었다.

이렇게 생긴 실행파일을 !./Assignment1cuda를 통해 실행하자, 예상했던 결과와 같게 나오는 것을 확인할 수 있다.

고찰

GPU 컴퓨팅 Assignment1을 진행하면서 google colab 환경에서 위와 같이 변수 및 컴파일 등을어떻게 다뤄야 하는지 진행하며 이해해 볼 수 있었습니다. 해당 과제를 바탕으로 malloc이나 memcpy, free 등 메모리와 관련된 다양한 부분에서 기존의 c언어와 유사한 부분이 많이 존재함을 확인할 수 있었습니다. 동시에 이후에 꾸준히 사용될 google colab에 대하여 어느정도 접근해볼 수 있던 좋은 기회였던 것 같습니다.