**운영체제 HW1**

2017111683 남아현

1. 개요

이번 과제는 makefile을 이용해 stack, queue 프로그램을 작성 및 활용하여 주어진 문제를 푸는 것이다. 공백으로 구분된 한 라인을 입력으로 받아 strtok 함수를 사용해 구분하여 stack/queue에 저장하고 차례로 출력하도록 작성한다.

2. 프로그램 구조 설명

2.1 함수에 대한 설명

과제를 수행하기 위해서 우선 stack에 관한 함수인 create\_stack, push, pop을 만들고, queue에 관한 함수인 create\_queue, enqueue, dequeue를 만든다. 이후 과제를 수행하기 위해strtok의 기능과 함께 입출력 중간과정을 통째로 담은 strtok\_stack, strtok\_queue 란 함수를 각각 만들었다. 우선 stack은 STACK이라는 큰 구조체 안에data와 다음 노드를 가리키는 next라는 변수를 포함한 NODE라는 구조체를 담고있다. 즉, STACK 구조체 안에 NODE라는 구조체를 차근차근 쌓는 형태이다. create\_stack은 처음 STACK이라는 큰 틀을 만들기 위한 함수이다. push와 pop은 각각 NODE를 STACK에 넣고 꺼내는 함수이다. queue도 이와 비슷하게 create\_queue를 이용해 QUEUE라는 큰 구조체를 만들고 enqueue, dequeue라는 함수를 이용해 각각 NODE를 QUEUE에 넣고 꺼낸다.

strtok\_stack과 strtok\_queue는 입력 line을 받아 출력값을 printf하는 void함수이다. strtok함수를 이용하여 공백을 기준으로 숫자를 구분하고 그 숫자를 그대로 push한다. 또한, 공백은 이미 NULL으로 반환되었으므로 숫자 push앞에 다시 공백을 push하여 STACK에 공백과 숫자가 번갈아가며 쌓이도록 한다. 이후 line을 전부 push하면, STACK의 NODE갯수(즉, STACK의 count변수)가 0이 될 때까지 pop하여, 출력값이 출력되도록 한다. strtok\_queue역시 이와 같게 하되, queue는 입력 순서대로 출력되므로 숫자를 먼저 push하고 그 후 번갈아가며 공백을 push하는 것이 다른 점이다.

2.2 다이어그램

그림에서 파란색은 함수를, 주황색은 STACK과 QUEUE를 나타낸다. 노란색은 입력과 출력을 나타내며 화살표를 통해 관계를 정리해보았다.

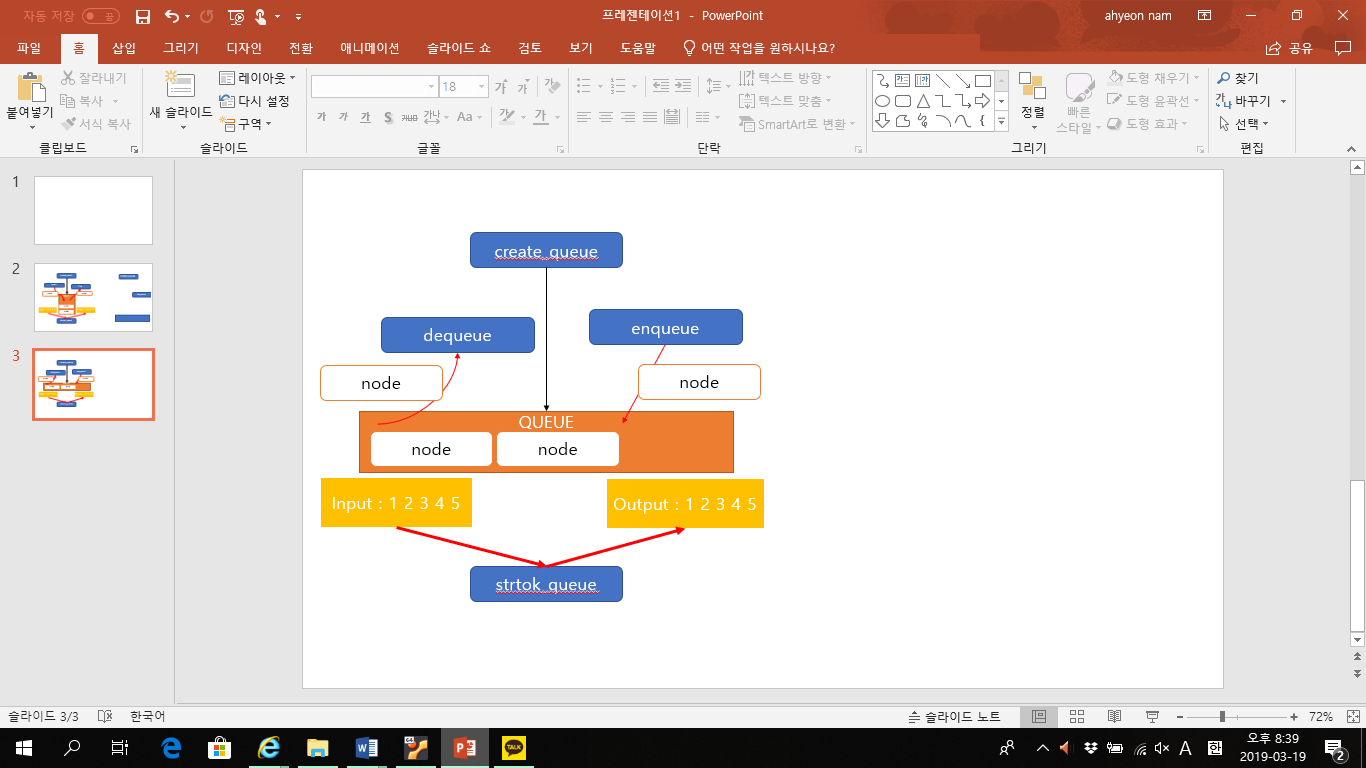
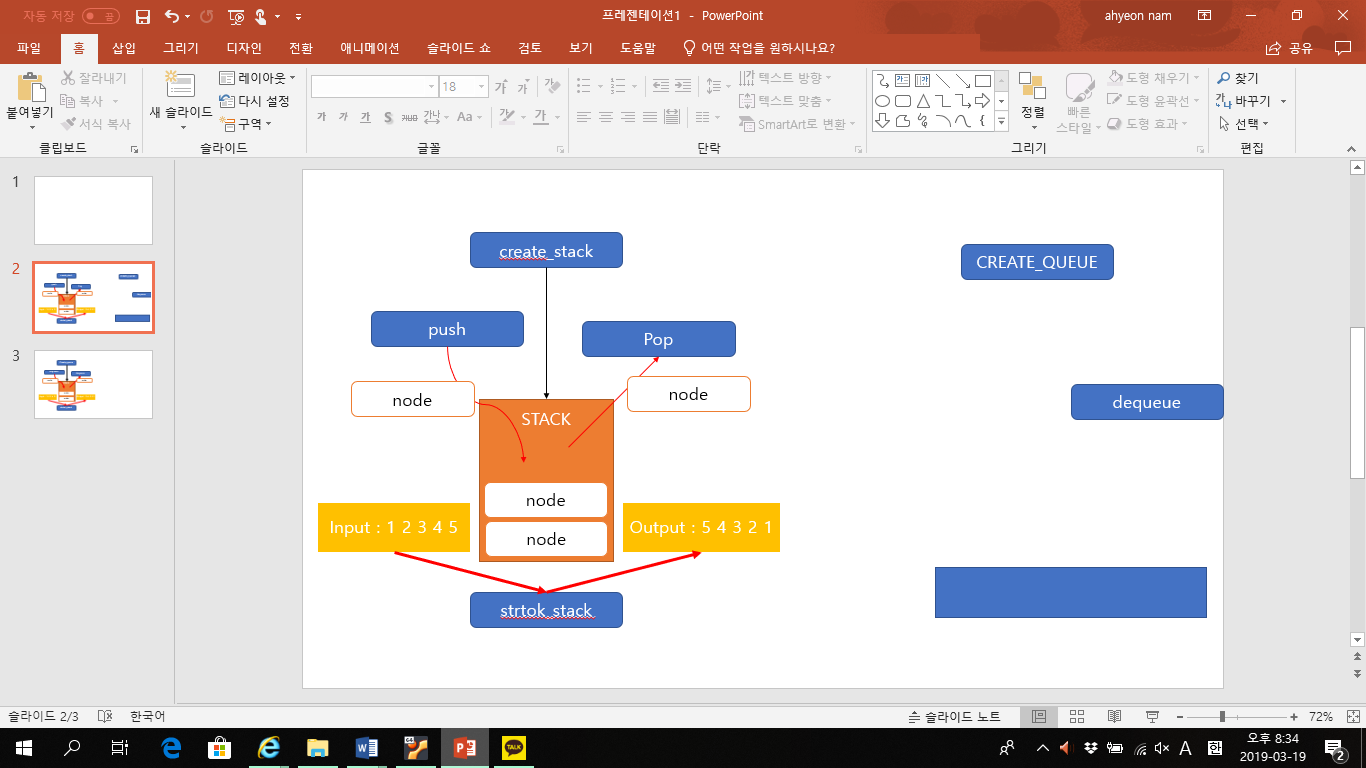
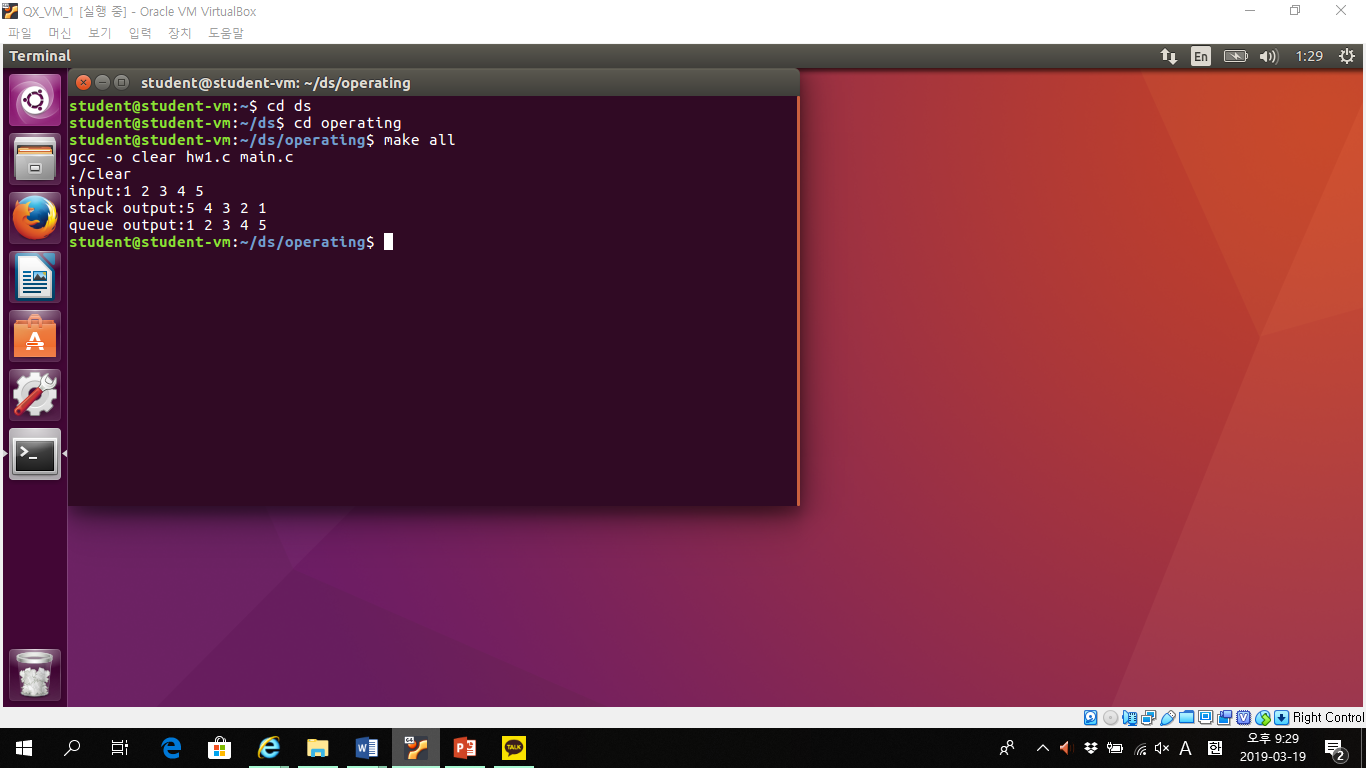


Figure . stack 다이어그램 Figure 2. Queue 다이어그램

3. 실행 결과



4. 고찰

과제에 본격적으로 임하기 전에 stack과 queue가 제대로 동작하는지 확인 후 과제에 임해야 했는데 그러지 않아 오류를 범했다. 실제로 dequeue과정에서 NODE간의 연결이 제대로 넘겨지지 않았다. (enqueue->head=temp->next 과정을 생략했다.) queue->head가 가리키는 것은 원래의 queue->head 그대로였고 그 NODE는 free가 되었다. 결국 NODE간의 연결이 끊기는 현상이 발생했다. 당연히 segmentation error가 떴고 이를 한참이나 해결하지 못하다가 결국 함수 코드를 하나하나 살펴가며 그림 그리며 겨우 해결했다.

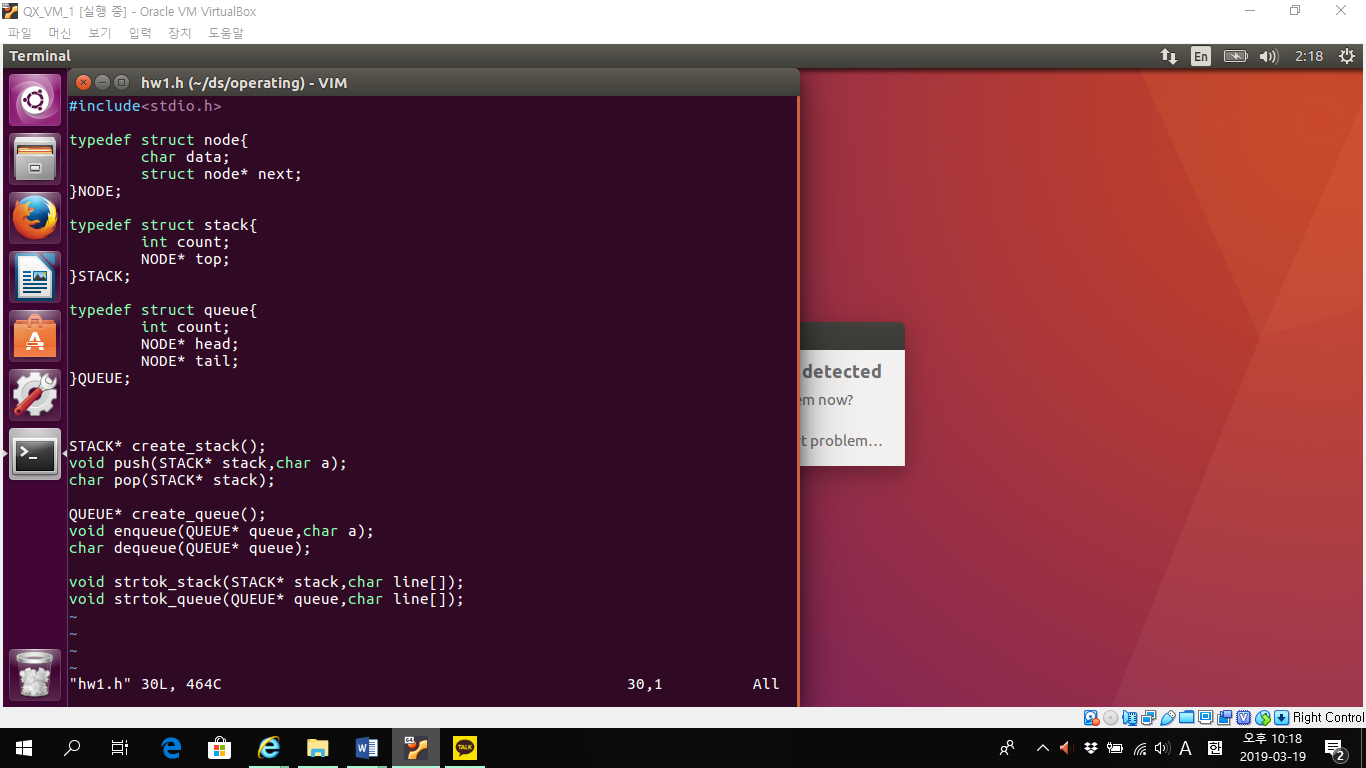
strtok함수를 활용할 때 그 원리를 제대로 이해하지 않은 채 코드를 짜서 에러가 생겼다. 특히 line이 strtok를 한 번 거치면 구분자는 NULL로 반환된다는 사실을 제대로 인지하지 못해 같은 line을 stack, queue 둘 다에 적용시키니 line을 두번째로 사용하는 경우 결과가 제대로 출력되지 않았다. Strtok를 거치고 공백이 없어진다는 사실을 알지 못한 것이다. 결국 line, line2를 따로 만들고 각각 strtok\_stack과 strtok\_queue에 적용했다.

사실 처음에는 strtok\_stack함수와 strtok\_queue함수를 처음에는 만들지 않으려 했다. Main함수에 직접 쓰려 했는데 함수를 쓰지 않으니 main함수가 너무 길어지고 헷갈리며 지저분해졌다. 또한 에러 파악에도 시간이 훨씬 많이 걸렸다. 이에 strtok\_stack과 strtok\_queue함수를 만들기로 결정했다. 이번 HW1를 통해 가장 크게 느낀 점은 main함수를 최대한 간단히 만들수록 에러를 잡아내기 쉬우며 웬만하면 사용자지정함수를 만드는게 편리하다는 것이다

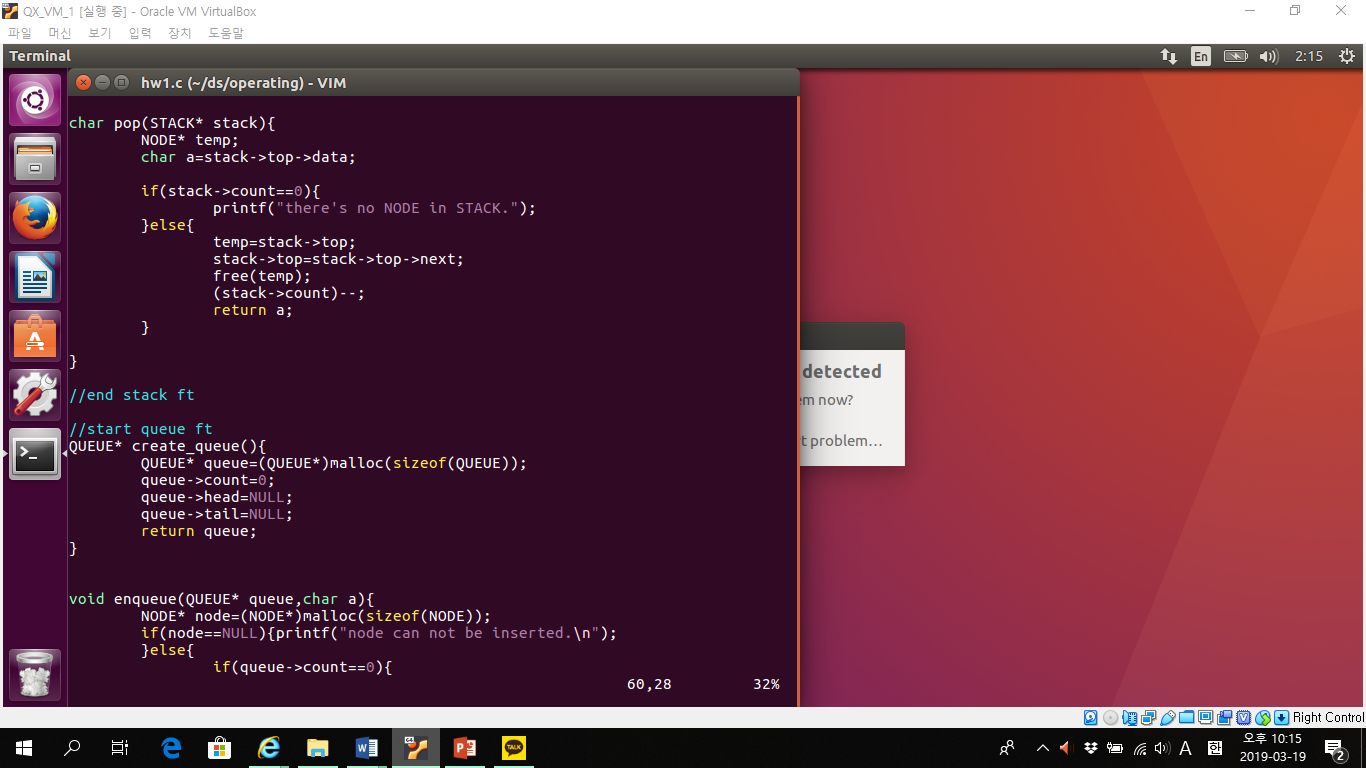
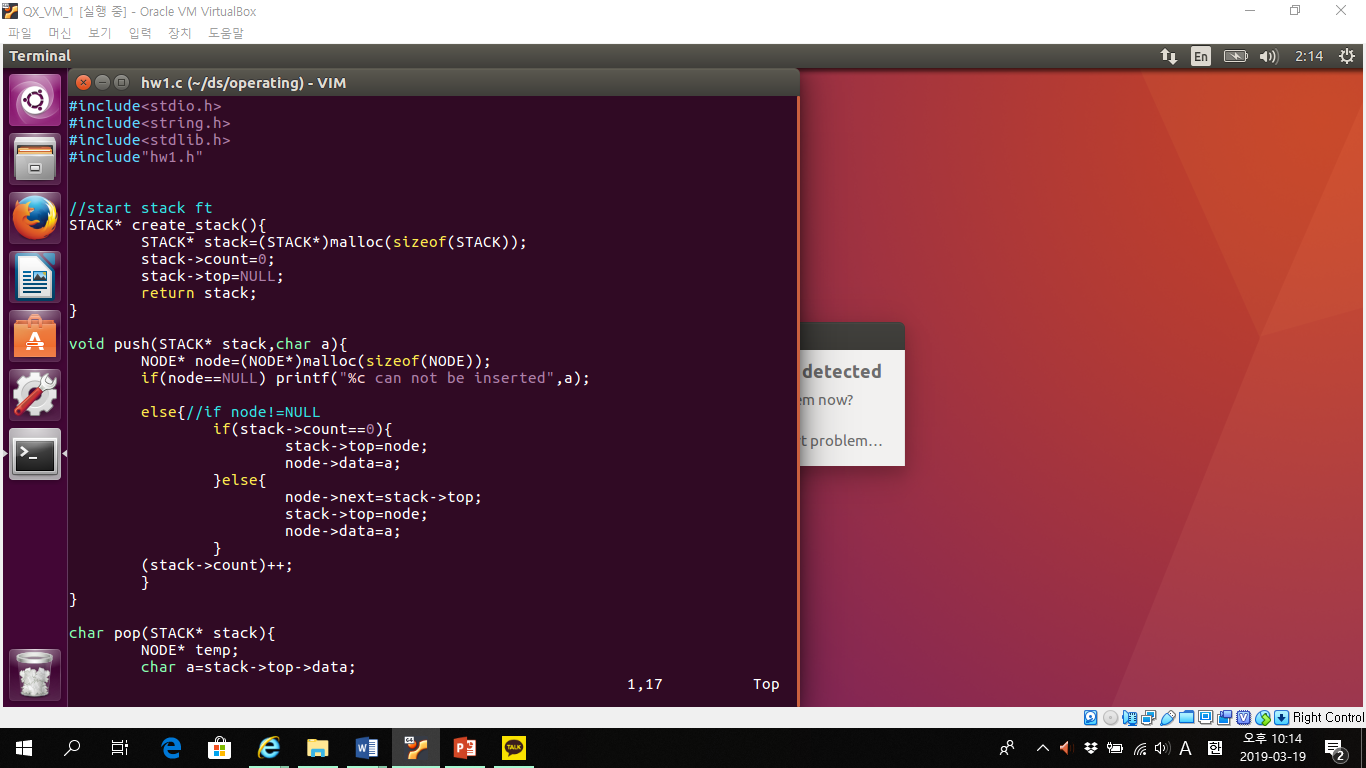
5. 프로그램 소스 파일

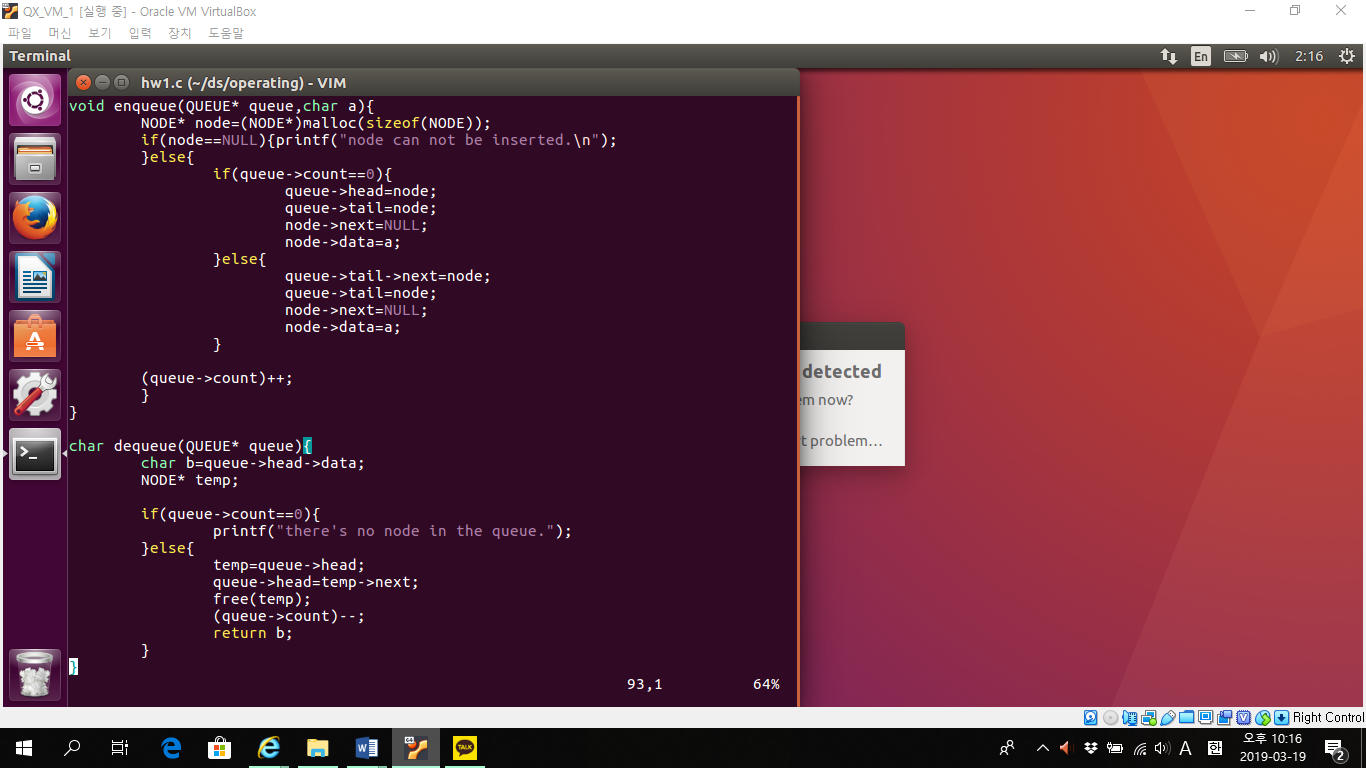
프로그램 소스 파일은 총 4개로 구성되어 있다. 사용자정의 헤더파일인hw1.h, 사용자정의함수를 담고있는 hw1.c, main함수를 담고있는 main.c, 그리고 Makefile이다.

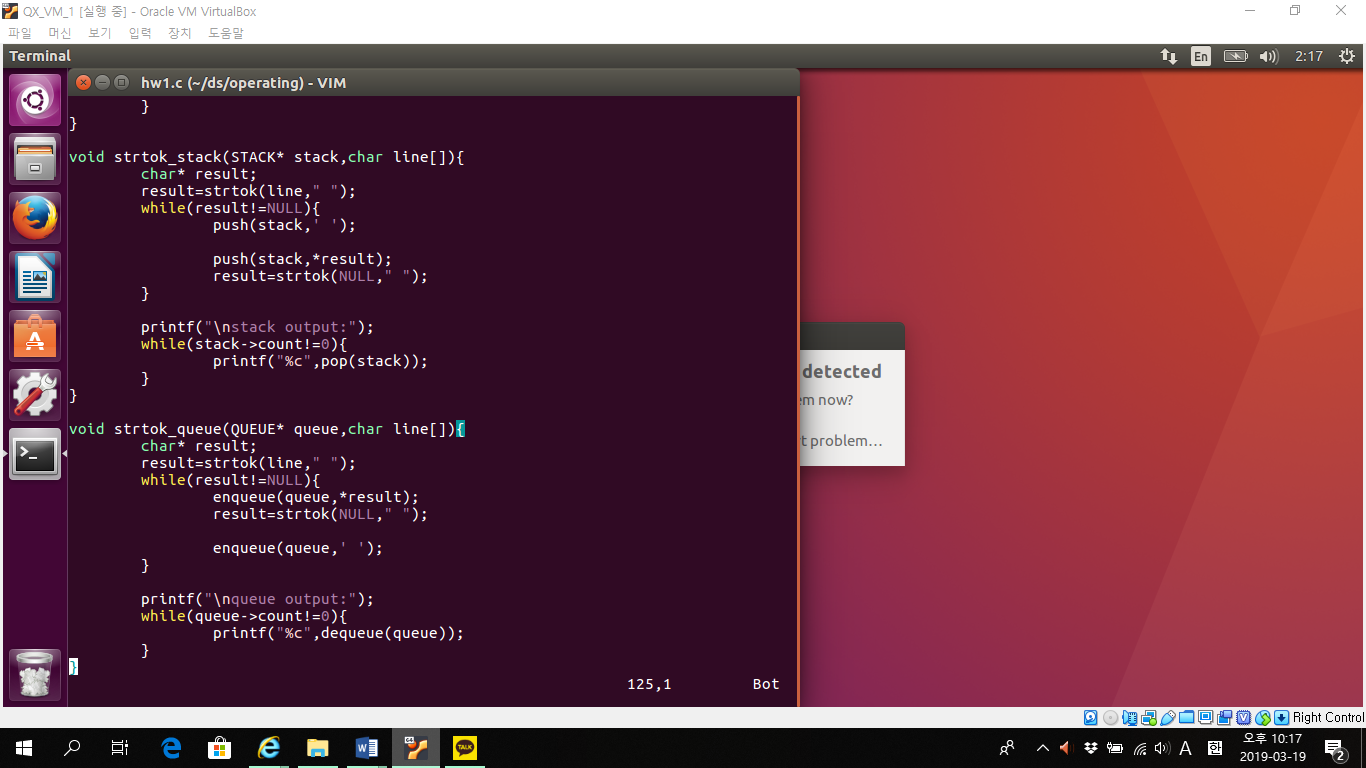
<hw1.h파일(사용자정의 헤더파일)>



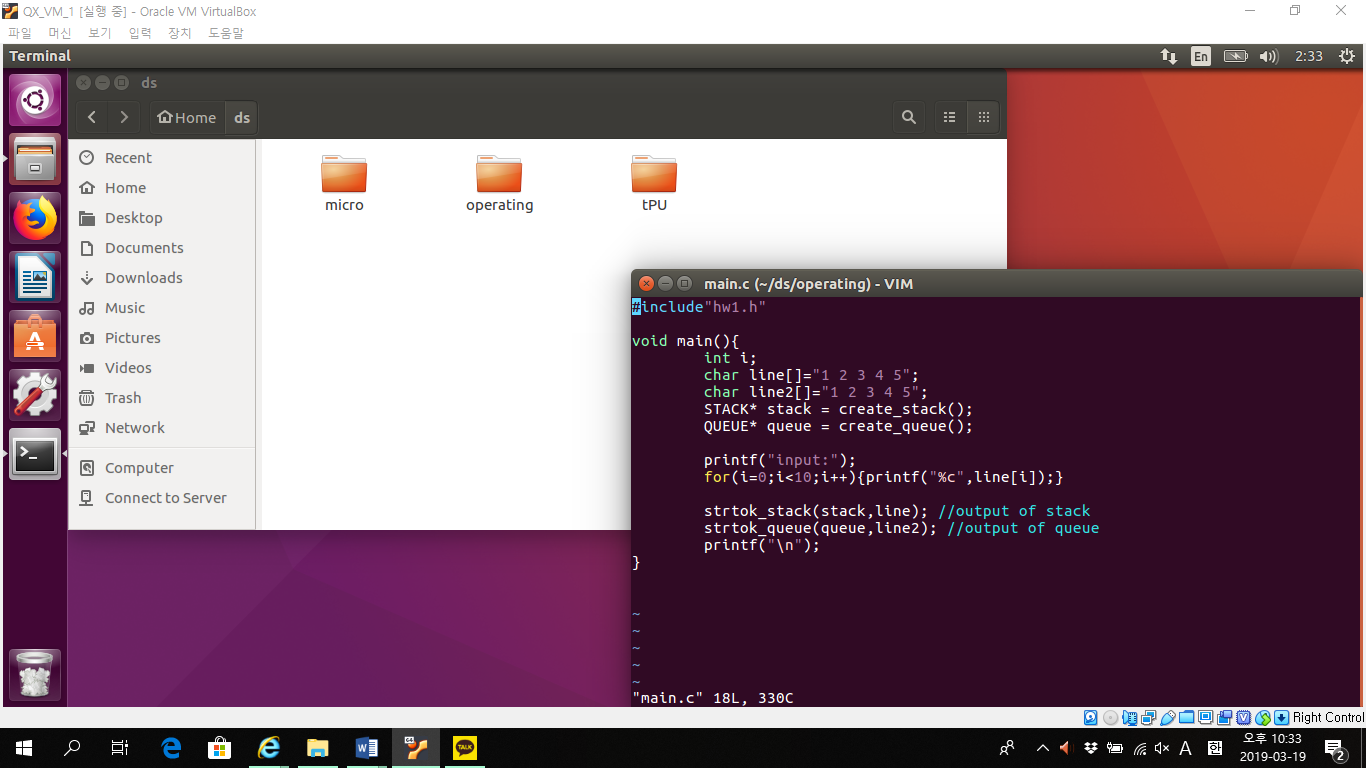
<hw1.c파일(사용자정의 함수파일)>



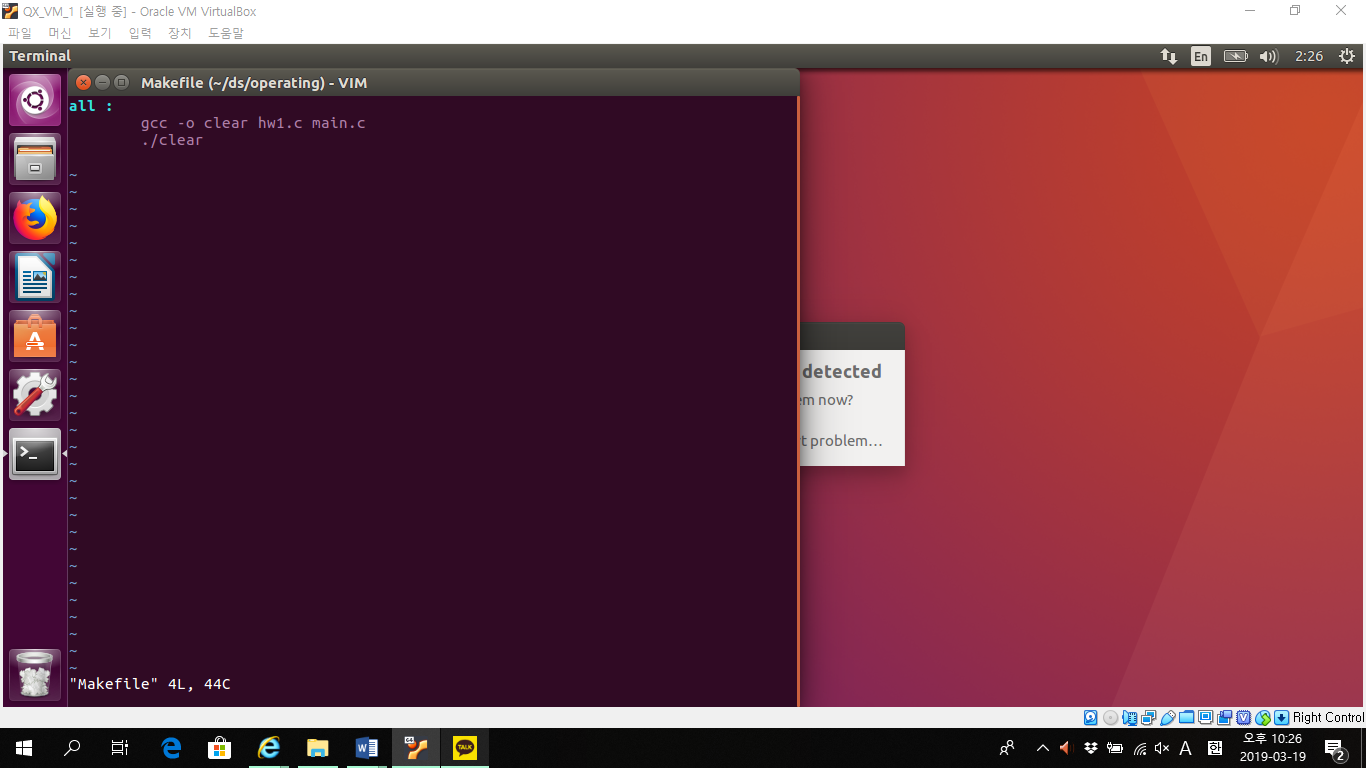




<main.c파일>



<Makefile>



6. 자료 출처

https://blog.naver.com/sharonichoya/220490957748