**운영체제 HW04**

2017111683 남아현

1. 개요

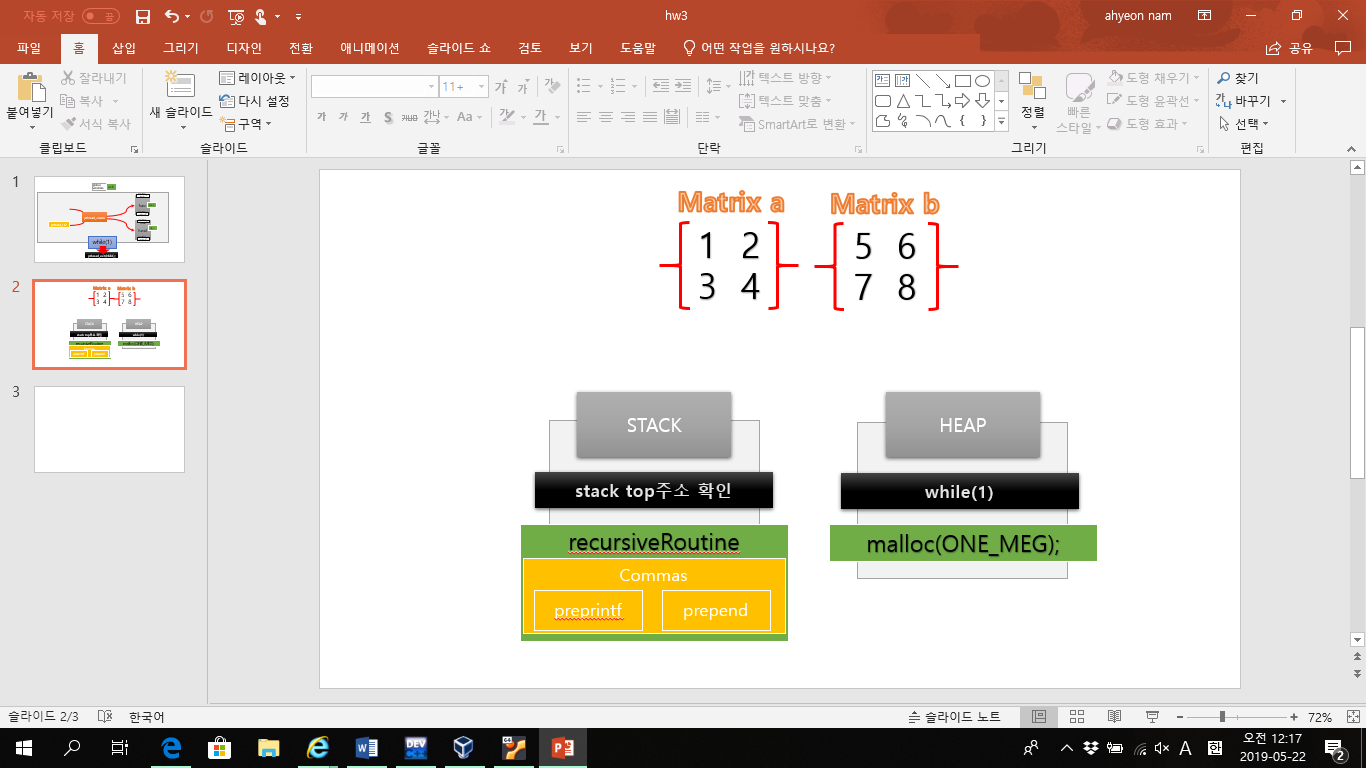
이번 과제는 stack과 heap이 공유하는 메모리 segment를 확인해보는 과제이다. Stack과 heap을 증가시키면서 충돌하는 지점을 찾고 stack과 heap 각각의 크기와 총 메모리 segment의 크기를 구한다. 또 stack과 heap의 크기를 달리하여 다른 지점에서 충돌시켜 처음 경우와 결과를 비교한다.

2. 프로그램 구조 설명

2.1 함수에 대한 설명

HEAP과 STACK을 각각 따로 확인 할 수 있도록 나눠 코드를 짰다. HEAP의 경우 while(1)문을 사용하여 메모리를 malloc으로 반복할 때마다 1메가바이트 증가시켜가며 할당했다. STACK의 경우 recursive하게 동작하도록 사용자정의함수를 만들었다. main함수에서는 처음에 commas함수를 이용해 stack의 top의 주소를 확인한 후 recursiveRoutine함수를 돌려 차지한 stack의 메모리를 확인했다.

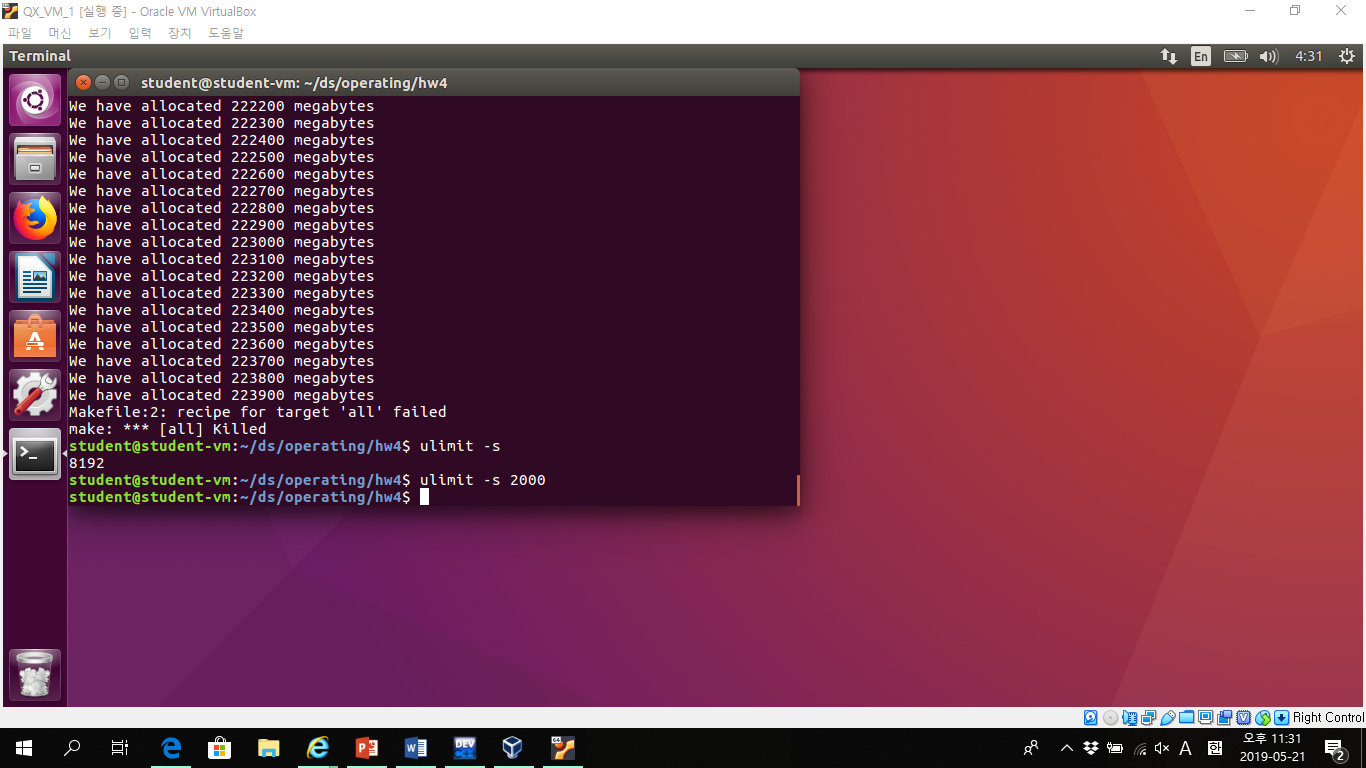
2.2 다이어그램



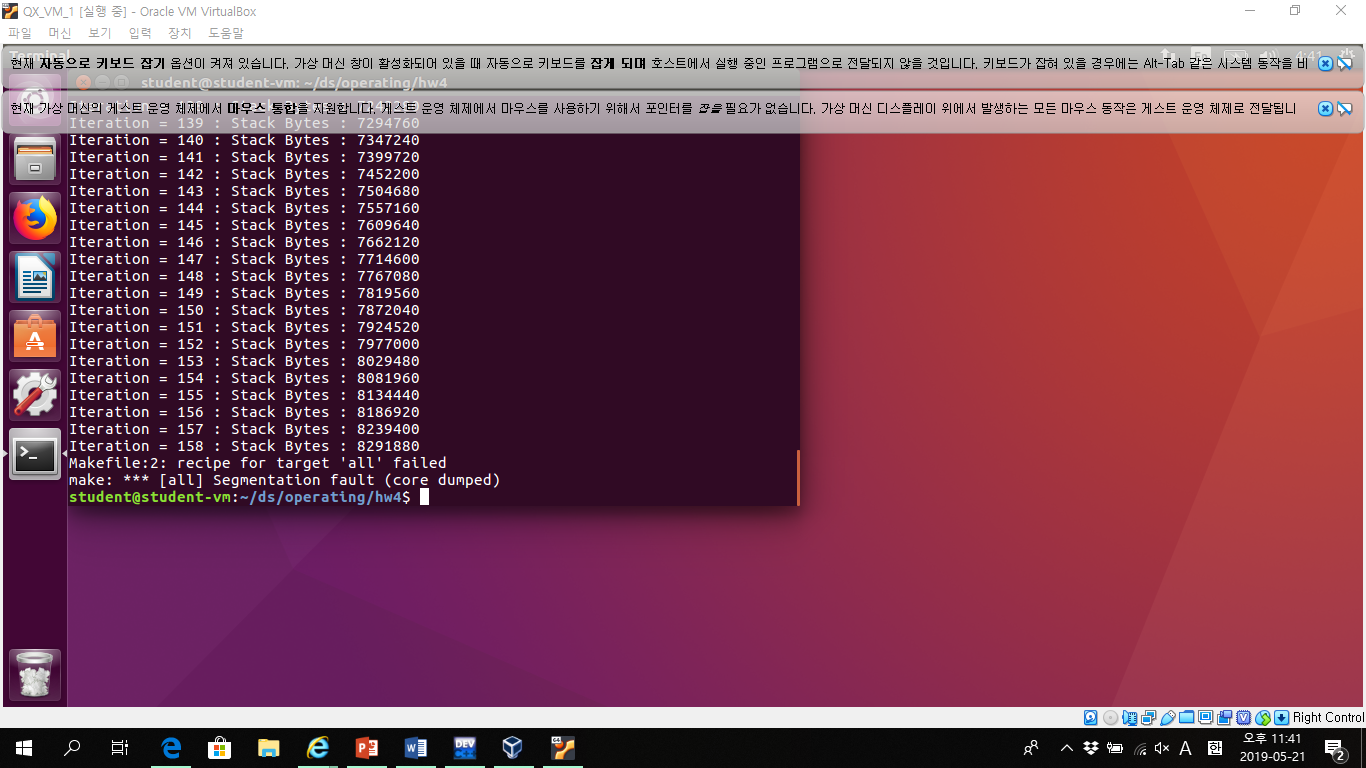
3. 실행 결과

1. ulimit -s 8192

<HEAP>

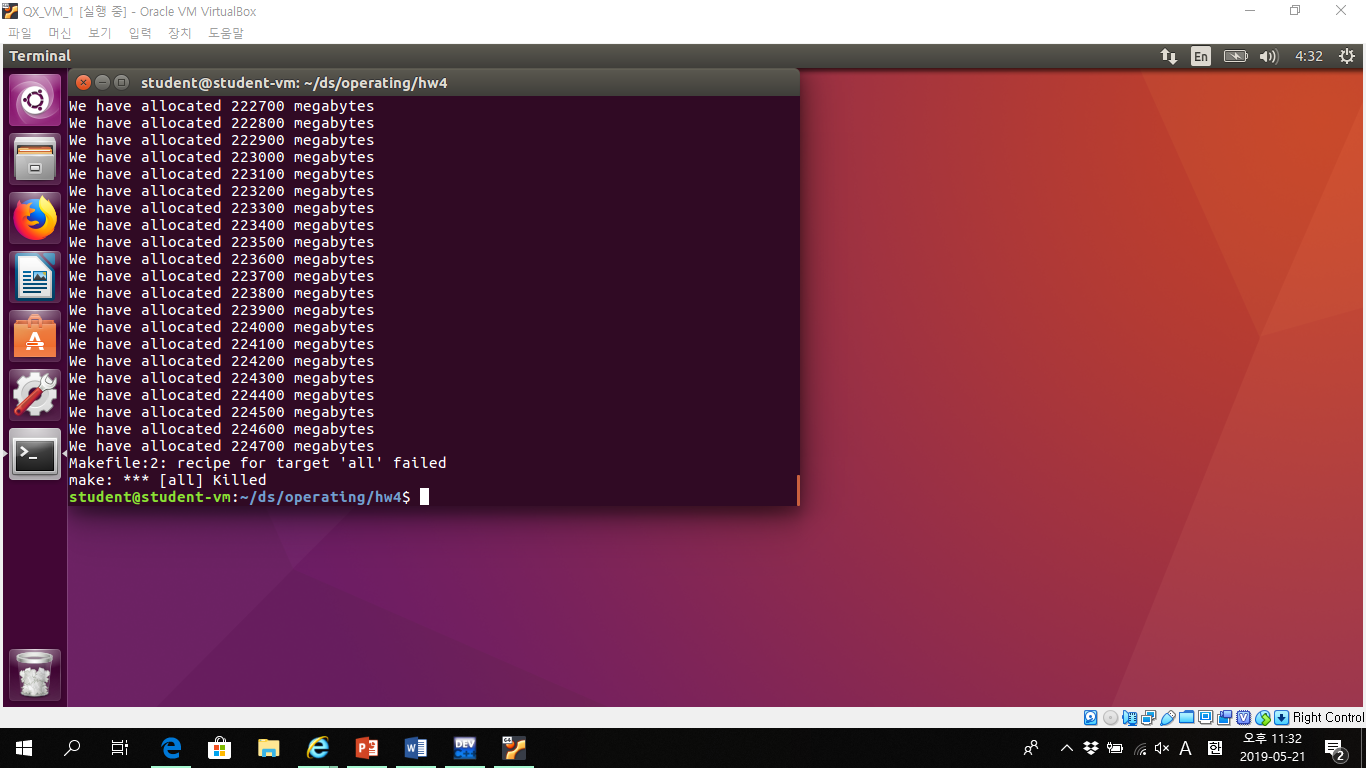


<STACK>

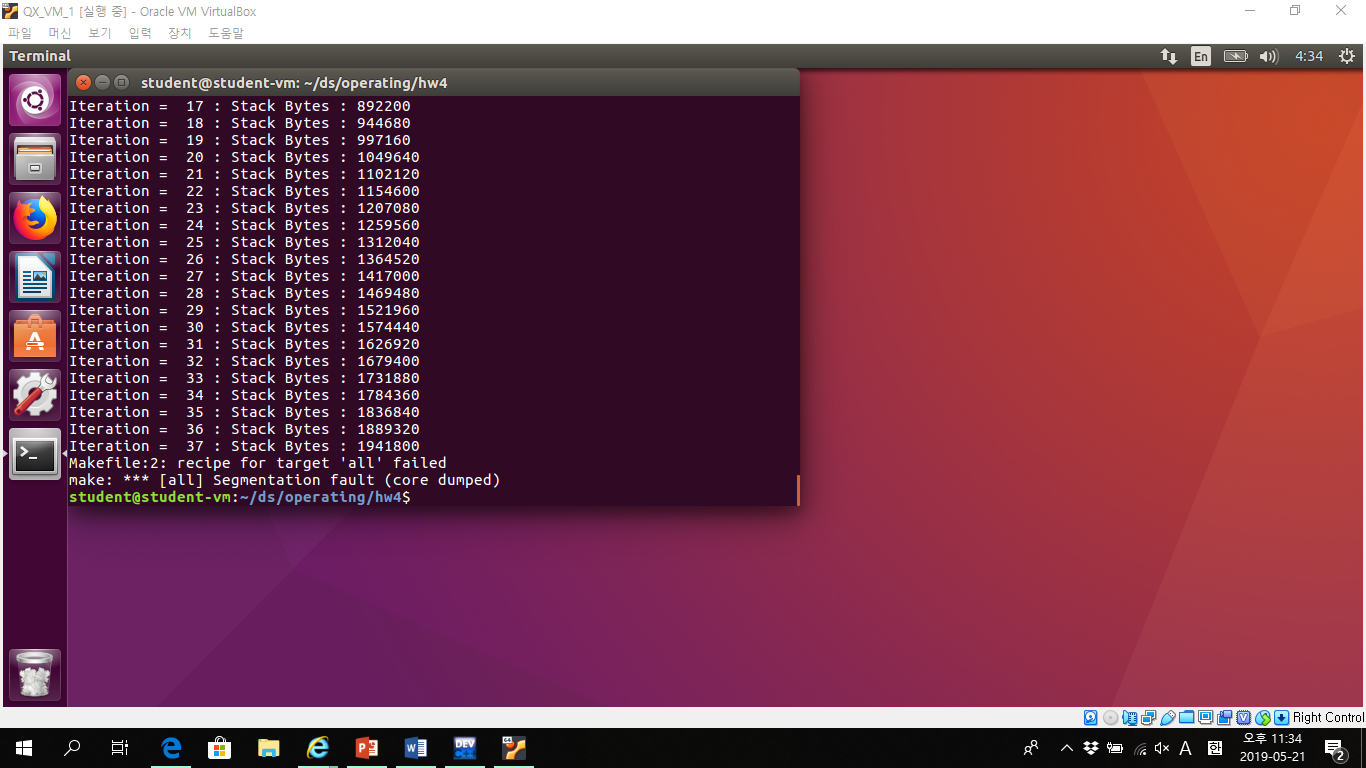


1. ulimit -s 2000

<HEAP>



<STACK>



\*\*약 2kB

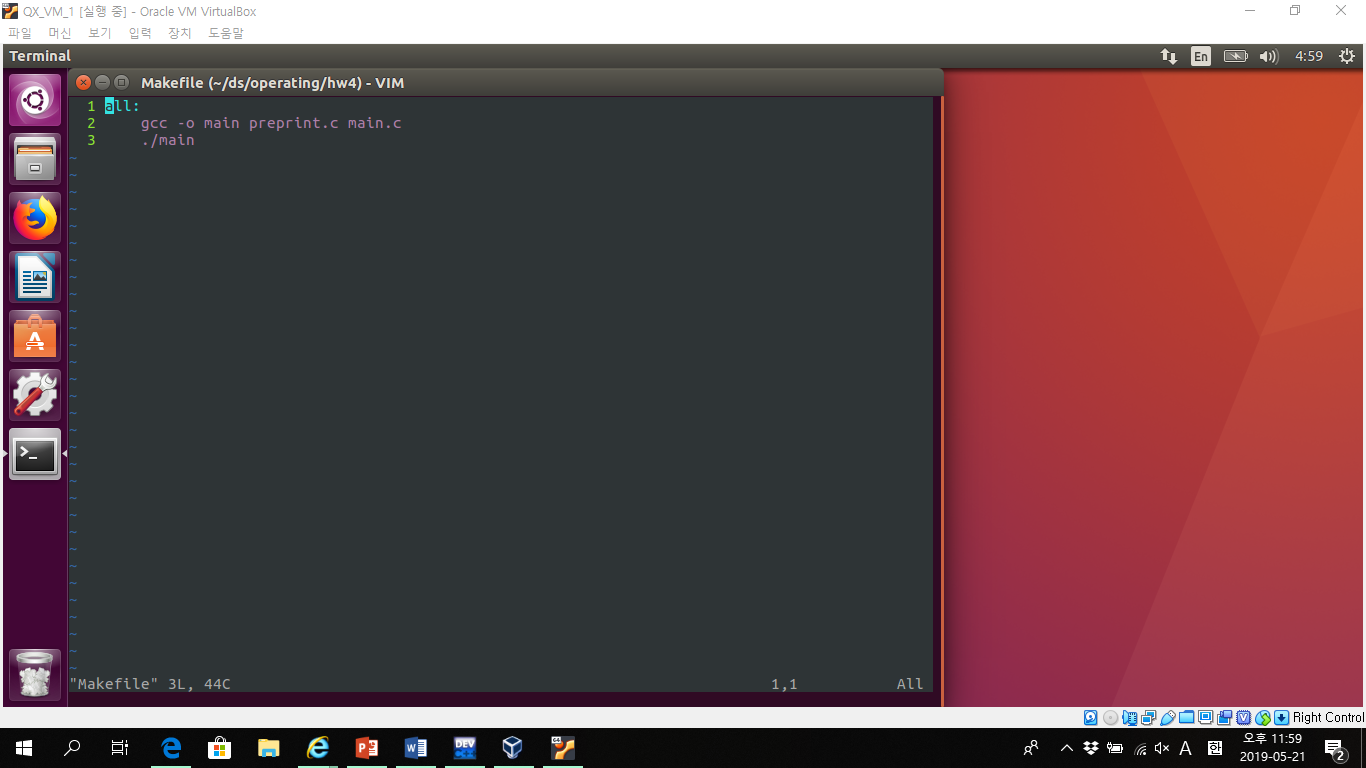
4. 고찰

HEAP의 경우 동적 할당 메모리를 늘리는 과정이 비교적 간단하게 느껴졌으나, stack의 경우 어렵게 느껴졌다. 특히 stack은 메모리를 스스로 관리한다는 지점에서 헷갈렸다. 필요하지 않은 것들에 대해 메모리를 다시 뺏는데 어떻게 해야 하나 고민이 많았다. 예제로 올라온 파일 역시 해석하기 어려웠는데, 차근차근 코드를 해석하고, 포인터를 활용하면 된다는 것을 깨달았다. 포인터에 대해 아직 부족한 부분이 많다는 것을 느꼈다. Heap에 대해서, 크기를 지정해 줄 경우 충돌이 잘 일어나지 않아 while(1)문으로 무한하게 돌려 충돌을 유도했는데 교수님께서 의도한 바 대로 잘 하고 있는건지 의문이 들었다. 하지만 malloc할당 사이즈를 늘려주거나 loop 크기를 늘려도 상황이 달라지지 않아 무한loop를 시도했다.

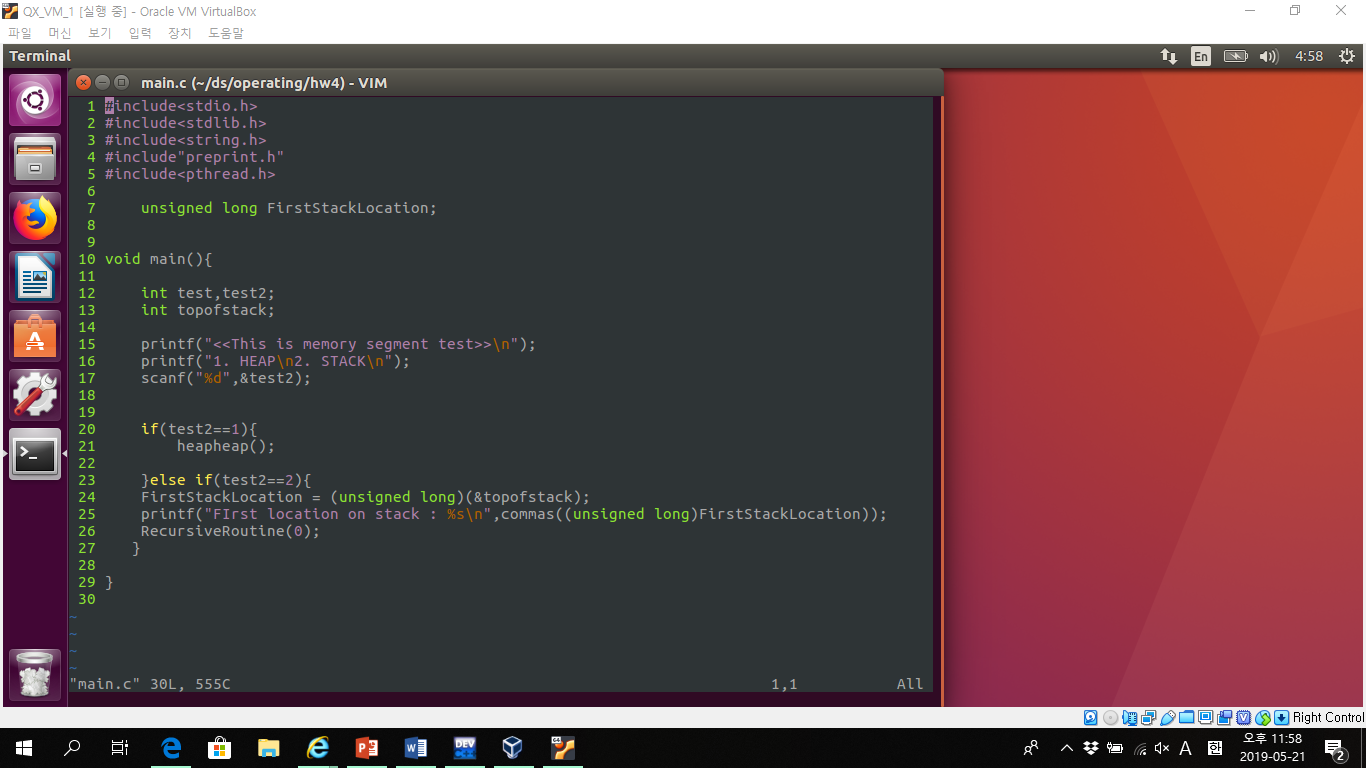
5. 프로그램 소스 파일

프로그램 소스 파일은 총 4개로 구성되어있다. Heap과 stack의 recursive함수를 담고있는 preprint.c, preprint.h, main함수를 담고있는 main.c, 그리고 Makefile이다.

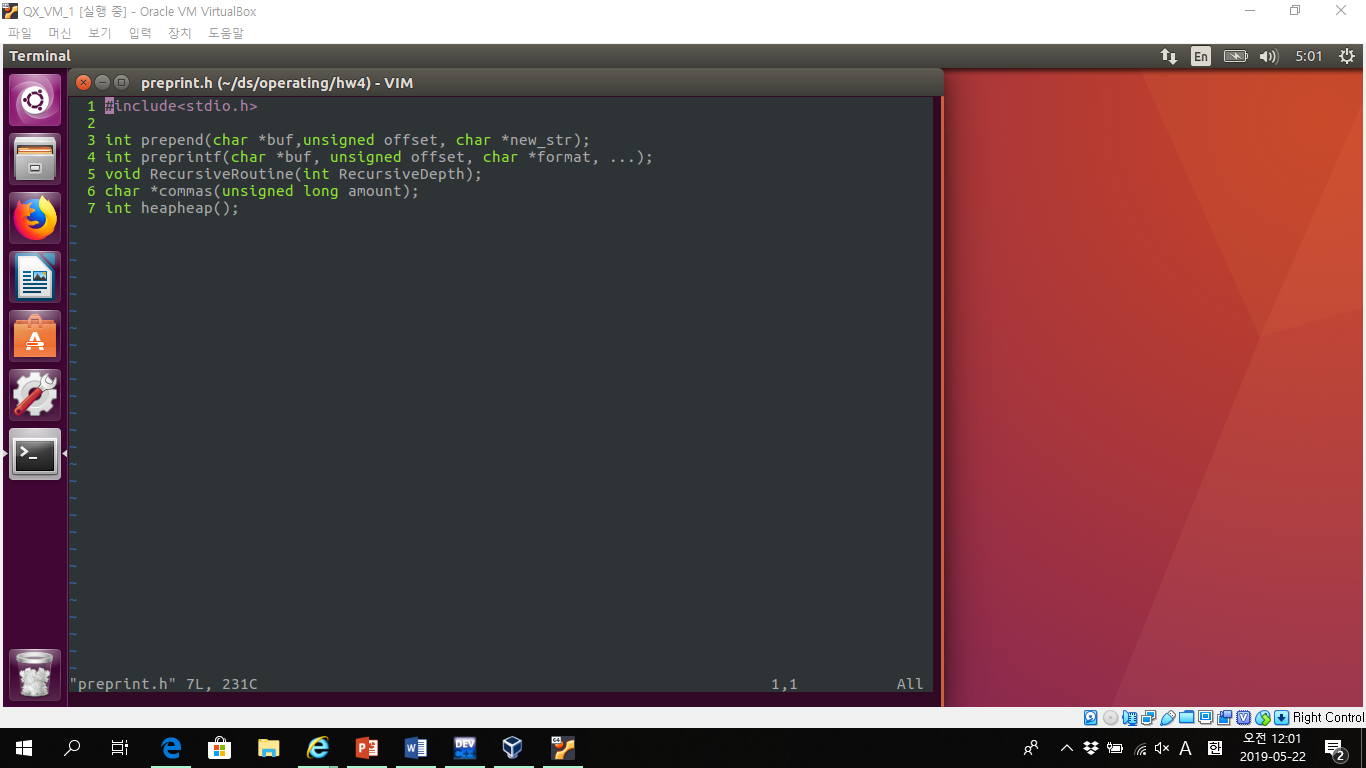
<Makefile>



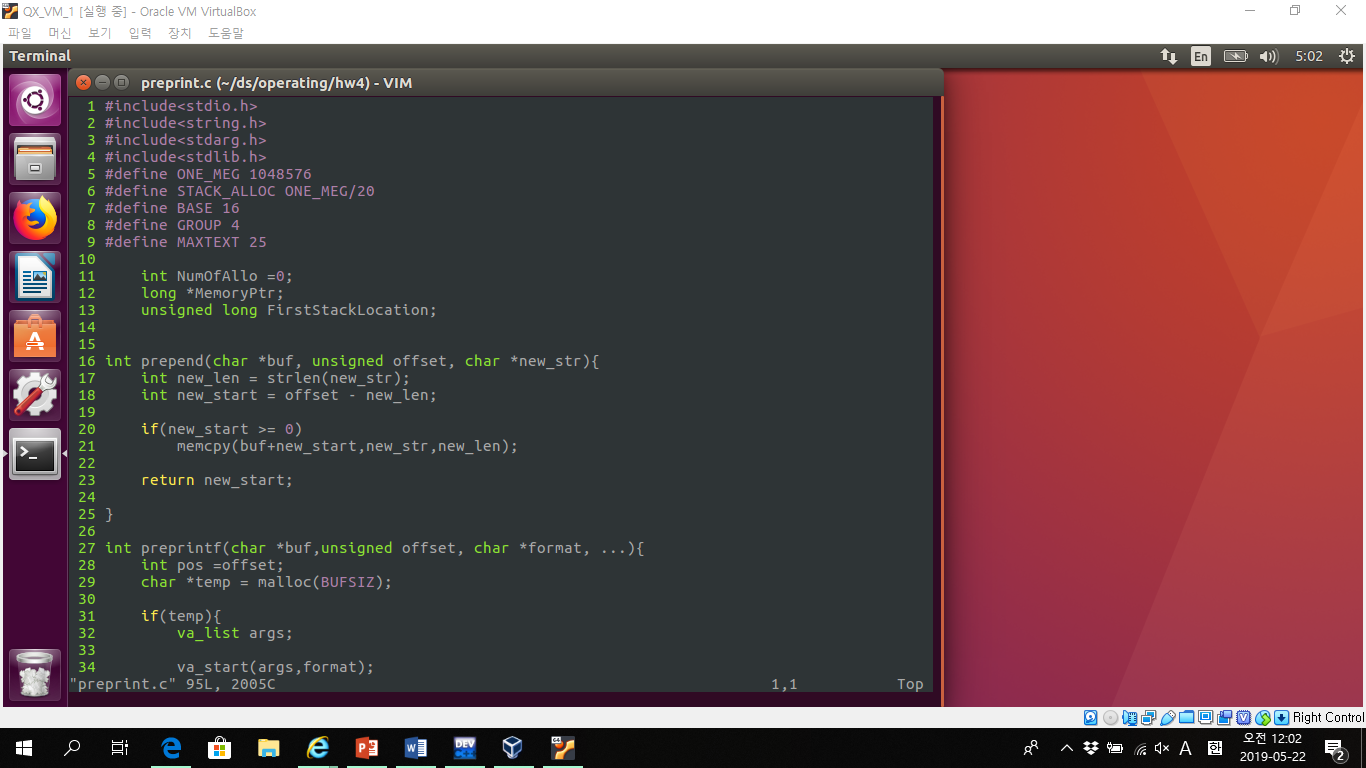
<main.c>

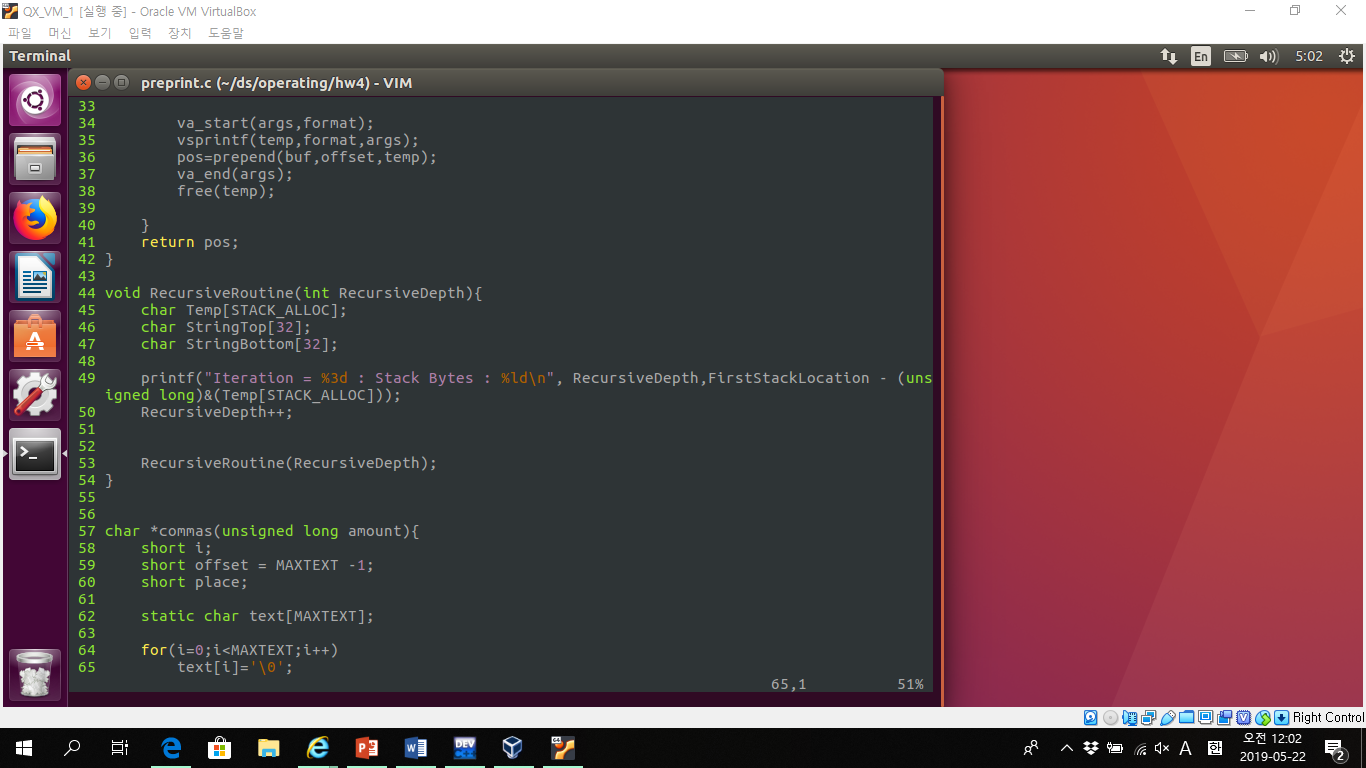


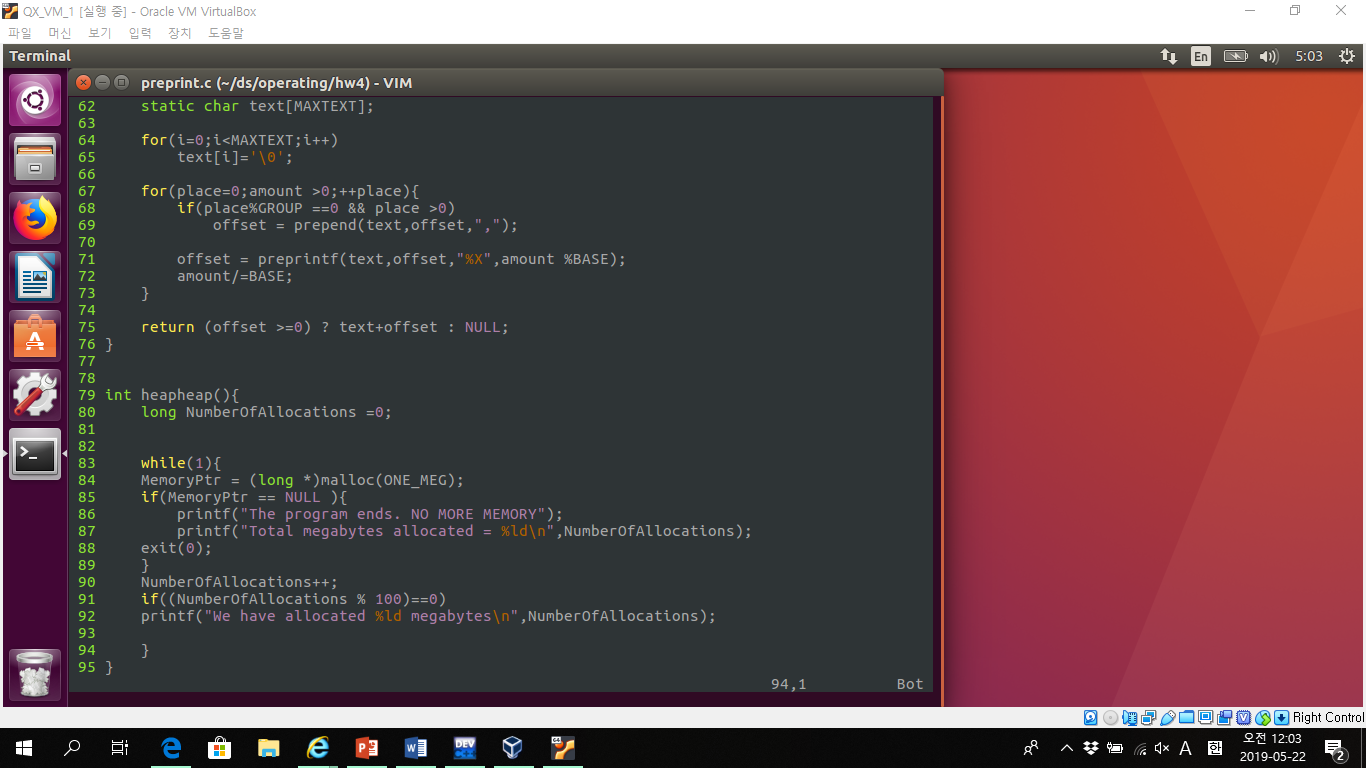
<preprint.h>



<preprint.c>







6. 자료 출처

<https://blog.naver.com/awes0megod/221296305196> atoi함수