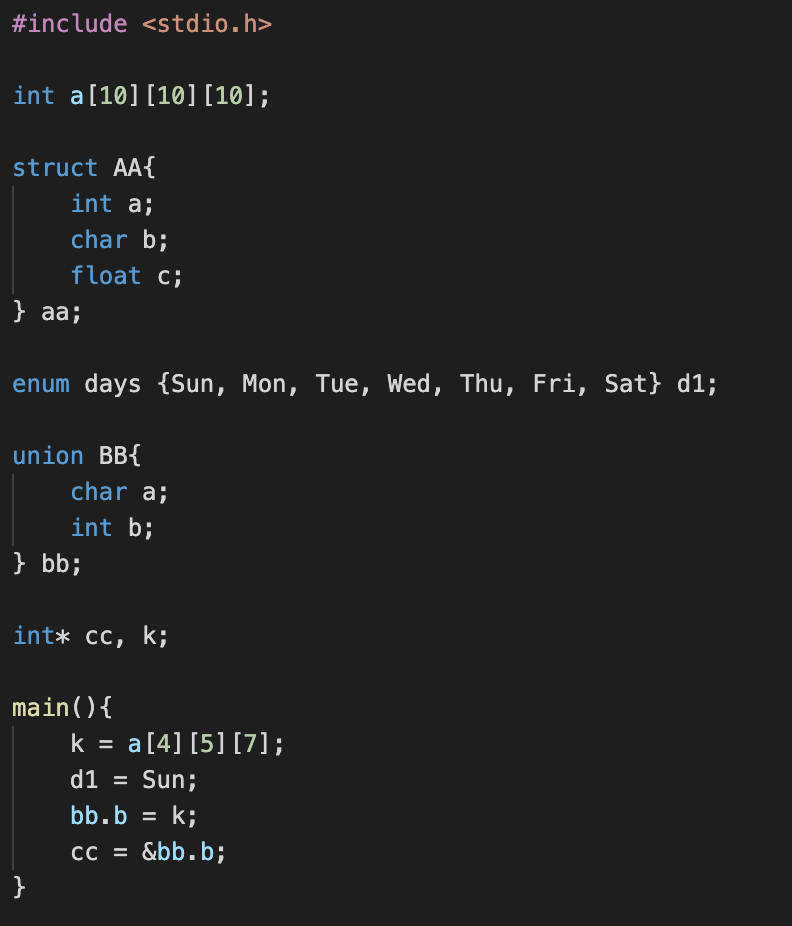
<Homework 3> Analyze the assembly code of the program “test.c”

20201635 컴퓨터공학과 전찬

test.c code는 아래와 같다.



<VScode로 작성된 test.c code>

위 코드를 통해서 전역 변수로 3차원 array a, 그리고 struct type AA와 변수 aa, enum type days와 변수 d1, union type BB와 변수 bb, int pointer cc, int 형 변수 k 가 존재함을 파악할 수 있다. 또한 main 부분에는 k, d1, bb.b, cc 등에 할당이 일어남을 파악할 수 있다.

수업에서 배운 내용을 바탕으로, 만약 array a의 시작 주소가 0 이라고 여기면 아래와 같은 형태로 저장될 것임을 파악할 수 있다.

1. a의 시작 주소는 0이며, 10 \* 10 \* 10 \* 4 = 4000 byte가 할당된다.

2. aa의 시작 주소는 4000이며, 4 + 4(1 + word alignment) + 4 = 12 byte 가 할당된다.

3. d1의 시작 주소는 4012이며, 4(int size) byte가 할당된다.

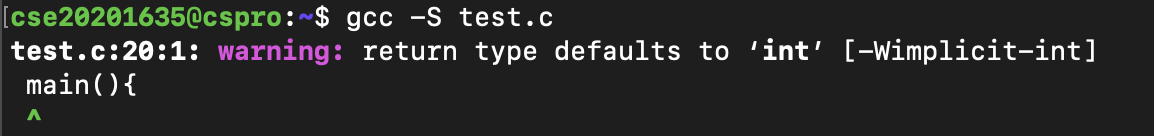
4. bb의 시작 주소는 4016이며, 4(char = 1, int = 4이므로 4) byte가 할당된다.

5. cc의 시작 주소는 4020이며, 8(64bit CPU) byte가 할당된다.

6. k의 시작 주소는 4028이며, 4byte가 할당된다.

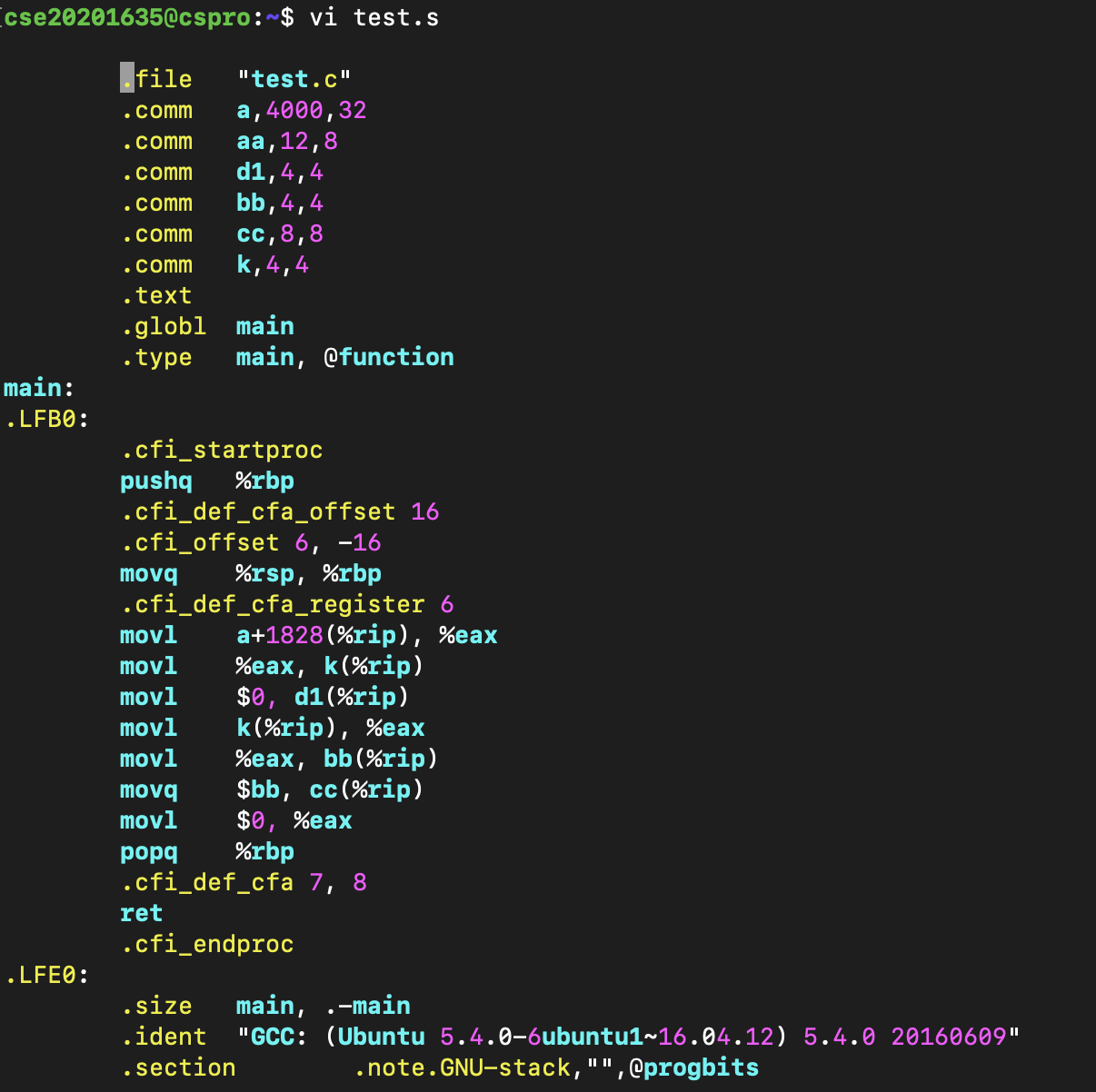
또한 이를 바탕으로 main() 부분에 존재하는 code 또한 분석해볼 수 있는데, k = a[4][5][7]; 부분은 a + 1828( = 4\*(4\*10\*10 + 5\*10 + 7)) 주소에 존재하는 data를 k 에 복사하는 형태, d1 = Sun;은 d1에 0을 복사하는 형태, bb.b = k;는 bb에 k를 복사하는 형태, cc = &bb.b;는 cc에 bb.b의 주소값을 복사하는 형태가 됨을 파악할 수 있다.

따라서 이를 linux server에서 gcc -S test.c command를 통해 assembly code 형태로 변환할 수 있으며, 이는 아래와 같다.



<linux에서 gcc -S test.c command 입력 결과>

위와 같이 warning이 존재하는데, 이는 main() 함수를 int main(){ ~~ return 0; } 을 해주지 않아서 발생하는 것이기 때문에, 분석할 때에 차이는 존재하지 않는다. 따라서 위 command를 토대로 만들어진 test.S file의 내용은 아래와 같다.



<test.s code>

위 내용을 살펴보면 기본적으로 file의 이름이 test.c인 것을 파악할 수 있다. 또한 많이 사용되는 명령어로는 .comm, movq, movl 이 존재하는데, 이는 각각 아래와 같다.

1. .comm : name, size, alignment 인자를 받으며, data 영역에 size byte 만큼을 할당한다. 할당된 data는 name을 통해 접근할 수 있다. alignment 인자는 optional이며, size에 따라 바뀌며, 기본적으로 최대 값은 32 이다.

2. movq : source, destination 을 인자로 받으며, source에 존재하는 data를 destination으로 옮기는 형태이다. 이때 q는 quadword로 8byte만큼 옮길 수 있다.

3. movl : movq와 동일하나 4byte만큼 옮길 수 있다.



<test.s code 분석>

또한 이를 토대로 아래 항목도 분석할 수 있는데,

#1. storage allocation

.comm 부분이 allocation을 수행하며, %rbp, %rsp, %rip 등을 활용해서 storage에 접근하는 형태임을 파악할 수 있다.

#2. referencing mechanism

cc = &bb.b가 assembly code에서 실제로는 movq $bb, cc(%rip) 와 같이 일어남을 토대로, referencing 하는 것도 값을 기준으로 다룬다는 것을 파악할 수 있다.