

포인터의 크기는 운영체제의 처리단위에 따라 크기가 달라진다.  
예를 들면 64비트 운영체제의 경우 한 처리단위가 64비트인 것이다.  
그래서 포인터는 64비트를 바이트로 환산할 경우 8바이트 이므로  
64비트 운영체제의 포인터 크기는 8바이트 이다.  
32비트 운영체제의 포인터 크기는 64비트 운영체제의 반인  
4바이트 이다. 컴퓨터는 포인터 방식으로 데이터를 처리 하므로  
처리단위가 8바이트이면 포인터 크기도 8바이트인 것이다.  
자료형의 크기에 상관없이 운영체제의 처리단위에 따른다.

진법 변환

10101110

2진수 -> 10진수

128 64 32 16 8 4 2 1 --> 175

1 0 1 0 1 1 1 0

2진수 -> 16진수

84218421 --> AE

10101110

333

10진수 -> 2진수

$2^8=256$

$2^6=64$

$2^3=8$  ,  $2^2=4$  ,  $2^0=1$

합이 333 2진수로 표현하면

->1001001101

10진수 -> 16진수

333을 16진수로 바로 변환하지는 못하고

2진수로 바꾸어서 진행해야한다.

8421 8421 8421

2진수가 0100 1001 1101이므로

--> 0x49D이다

8F23

16진수 -> 2진수

8 F 2 3

8421 8421 8421 8421

1000 1111 0010 0011

16진수 -> 10진수

2진수가

$2^{15}$   $2^{14}$   $2^{13}$   $2^{12}$   $2^{11}$   $2^{10}$   $2^9$   $2^8$   $2^7$   $2^6$   $2^5$   $2^4$   $2^3$   $2^2$   $2^1$   $2^0$   
32768 16384 8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1  
1 0 0 0 1 1 1 1 0 0 1 0 0 0 1 1이므로  
32768+2048+1024+512+256+32+2+1 36643이다.(맞는지는 모르겠습니다 )

rbp를 push하면 임의의 주소가 rbp에 할당된다.

그리고 동시에 rsp가 생기고 rsp는 rbp의 크기를

주소값으로 갖는다. 그리고 mov하여 rsp값을

rbp1에 복사해서 rbp1의 임의의 주소를 초기화하여

rbp2의 값을준다. 그리고 rsp는 sub로 값을 증가시켜

공간을 넓히고 그 공간안에 값을 넣는다.

그 값은 함수를 호출했을 때 함수안에서 사용된다.

