12. 널포인터(null) 역참조

- **널포인터 역참조**는 포인터가 **null**을 가지면서 어떤 변수의 메모리를 **참조**하는 것이다.
- **널포인터 역참조**는 프로그램 **오류**에 해당한다.
- 해서, **널포인터 역참조**가 발생되지 않도록 프로그램을 작성해야 한다.

// 역참조(dereference)란? – 주소를 통해 특정 변수의 값이 저장된 메모리에 접근

```
void main() {
   int a = 5;
   int *p = &a;          //포인터 선언에 사용된 기호 *를 역참조연산자(간접연산자)라 함
   printf("%d\n", *p);   //출력 : 5
}
```

*p=&a;	• 포인터 p는 변수 a의 메모리 주소를 가진다. • **역참조연산자** *를 이용하여 변수 a의 메모리에 접근하여 값 **5**를 가져온다.

// 함수 malloc()

원형	void *malloc(size_t size);

- 함수 **malloc()**가 메모리 할당을 **성공**하면 : 할당한 메모리의 **첫번째 주소** 반환
- 함수 **malloc()**가 메모리 할당을 **실패**하면 : **null** 반환**(중요한 내용)**
- 함수 **malloc()**는 프로그램 실행 중에 **동적**으로 메모리를 할당한다.
- 동적 메모리 할당은 **힙**(heap) 영역에 메모리를 할당한다는 것이다.

----<널포인터 역참조가 발생 가능한 코드>----

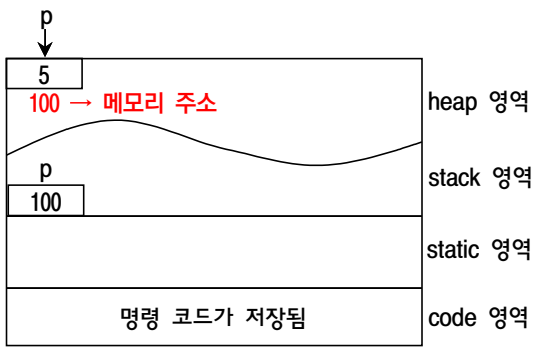
```
#include <stdio.h>
void main() {
   int *p;                          //포인터
   p = (int *)malloc(sizeof(int));  //힙할당, 할당할 메모리 공간이 없으면 null 반환
   *p = 5;                          //포인터 p가 null을 가지면 널포인터 역참조가 됨
   printf("%d\n", *p);              //출력 : 5
   free(p);                         //포인터 p가 가리키는 메모리 회수
}
```

그러면, **널포인터 역참조**가 발생되지 않도록 하려면 어떻게 해야 하는가?

```
----<널포인터 역참조 예방>-----------------------------------------------
 #include <stdio.h>
 void main()
 {
   int *p;                          //포인터
   p = (int *)malloc(sizeof(int));  //힙할당, 할당할 메모리 공간이 없으면 null 반환
   if(p==null) exit(0);             //포인터 p가 null을 가지면 프로그램 종료
   *p = 5;                          //포인터 p는 null을 가지지 않음
   printf("%d\n", *p);              //출력 : 5
   free(p);                         //포인터 p가 가리키는 메모리 회수
 }
-------------------------------------------------------------------------
```

• **널포인터 역참조**가 발생되는 것을 방지하려면 포인터 사용전에 if문으로 조사하면 된다.

```
  if(p==null) exit(0);             //포인터 p가 null을 가지면 프로그램 종료
```



· 변수 p는 스택 영역에 할당된다.

```
// 함수 malloc() 사용법
 char *p1;
 int *p2;
 p1 = (char *)malloc(50);
 p1 = "KOREA";
 printf("%s\n", p1);              //KOREA
 p2 = (int *)malloc(50*sizeof(int));
 p2[0] = 88;
 printf("%d\n", p2[0]);           //88
```

// 널포인터 역참조 오류를 이용한 공격과 예방

- 널포인터 역참조 **오류**는 포인터가 null을 가질 때 발생되는 오류이다.
- 널포인터 역참조 **오류**는 '객체는 null이 될 수 없다'는 가정을 위반했을 때 발생한다.
- 공격자가 의도적으로 널포인터 역참조를 발생시켜, 발생되는 오류를 이용하여 공격할 수 있다.
- 즉, 공격자는 널포인터 역참조 오류를 이용하여 어떤 공격을 계획할 수 있다.

----<안전하지 않은 코드>----------

```
void main()
{
  int *p;
  p = (int *)malloc(sizeof(int)); //메모리 할당이 불가능하면 함수 malloc()는 null을 반환
  *p = 5;                         //널 포인터 역참조 오류 발생 가능(포인터 p는 null이 될 수 있음)
  printf("%d\n", *p);             //출력 : 5
  free(p);                        //기억공간 회수
}
```

↓

↓널 포인터(null) 역참조 오류 예방 시큐어 코딩

↓

----<안전한 코드(시큐어 코딩)>----------

```
void main()
{
  int *p;
  p = (int *)malloc(sizeof(int)); //메모리 할당이 불가능하면 함수 malloc()는 null을 반환
  if(p==null)                     //널 포인터 역참조 오류 예방 시큐어 코딩
    exit(0);
  else                            //포인터 p가 null이 아닌 경우
    *p = 5;
  printf("%d\n", *p);             //출력 : 5
  free(p);                        //기억공간 회수
}
```

- **널포인터 역참조**가 발생되는 것을 방지하려면 포인터 사용전에 if문으로 조사하면 된다.

```
if(p==null) exit(0);              //포인터 p가 null을 가지면 프로그램 종료
```