

C 어플리케이션구현 (Void Pointer)

동양미래대학교
컴퓨터정보공학부
2학년B반
20111448 이현호

①Void Pointer?(보이드포인터란?)

⇒(void*) 포인터란 형이 없는 포인터를 말한다.

C언어에서는 void는 '자료형이 없다', '자료형이 정해지지 않았다'라는 개념으로 사용한다.

보이드 포인터 같은 경우는 주소값만을 저장하기위해서사용한다.

주소값 자체는 모두 동일한 데이터형을 가지고있기 때문에,

어떠한 주소형 자료형이 되더라도 가진 주소값의 체제는 동일하다.

이때,void*로 선언을 하게되면 어떠한 자료형의 주소값이라도 형변환없이 대입이 가능하게 한다.

②Void Pointer 활용법

⇒보이드 포인터는 타입이 없는 포인터이기 때문에, 동시에 그 어떤 타입으로도 변환이 가능한 만능 포인터라고도 볼 수 있다.

주소값을 저장하는데 타입이 없기 때문에 캐스팅(Casting)을 통해서 타입을 바꾸어줄 수 있다.

즉,없지만 그 어떤 포인터로도 변화가 가능한 만능 포인터가 된다.

예를 들어 void 포인터 변수에 92라는 값이 들어있다고 가정했을 때, 포인터를통해 주소를 찾아가면 그곳에는 무엇인가가 들어있을 것이다. 정수일지,문자일지,문자열인지,배열인지,혹은 구조체가 들어있을수도있다.

하지만 우리는 아무것도 알수가없다. 왜냐하면 우리는 포인터를 통해서 주소값을 따라가고,

그 따라간곳에 값을 식별해내기 위해서는 포인터의 타입이 정해져있어야 하기 때문이다.

int*라면 그곳에 정수값이 있을것이고, char*라면 그곳에 문자값이 있을 것이다. 하지만 void*는 순수하게 주소값만 가지고 있을뿐, 그곳에 가봐야 어떠한 대상도 가르킬수 없게 된다.

그러나, 앞서 얘기했듯이 void*는 형변환(Casting)이 가능하기 때문에 이것은 곧 없음과 동시에 모든 것이 될수 있다는 아이러니한 포인터가 된다.

임의의 타입을 가리키는 void라는 키워드는 클래식 C에는 존재하지 않았다. 기존에는 메모리의 한지점을 가리키는 용도로 1바이트 단위로 이동되는 char*를 대신 사용했었다. 그러나 이렇게 되면 char*형에 다른 타입의 포인터를 대입할때마다 캐스트의 연산자가 필요해서 무척 불편했으며, char*가 정말 char형 변수를 가리키는 것과 임의의 타입을 가리키는 것이 구분되지않아 실수의 가능성도 높았다. 그래서 ANSI C표준에서 Void라는 타입을 추가했고 임의 타입의 포인터가 필요할 때 void*를 쓰기로했다.

③Void pointer 주의사항

⇒ ❶ *연산자를 사용할 수 없다.

void형 포인터는 주소를 값으로 갖는다, 하지만 *를 사용해서 값을 액세스 하지는 못한다.

이유는 void*는 데이터타입을 가지고있지않기 때문이다.

int는 4byte,char 1byte이다. 우리가 변수를 선언할 때 그사실을 알수가 있는데,

void로 선언을 하면 데이터의 크기를 알수 없으므로, 포인터가 가르키는 위치에서 몇 바이트를 읽어야할지, 또 읽어낸 비트를 어떤식으로 해석해야할지 모르기 때문이다.

❷ 증감 연산자를 쓸 수 없다.

대상체의 타입이 정해져 있지 않기 때문에 증감 연산자 또한 곧바로 사용할 수 없다.

또한, 정수값과 바로 가감 연산을 하는것도 허용되지 않는다. 대상체의 크기를 모르기 때문에 얼마만큼 이동해야할지 모르는 것이다. 연산을 하기위해서는 데이터의 주소값을 저장하고있다가 형변환(Casting)을 진행한 후에 가능하다.

④Void pointer 활용예

```
//보이드포인터 활용예(보이드포인터 변수에 다른 원소 형식의 포인터를 대입해보기)
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int array[5] = { 1,2,3,4,5,};
    char name[5 + 3] = "Hyeon";
    void *vp = 0;
    vp = array; //int 형을 원소로 하는 포인터를 대입
    printf("vp: %p arr: %p \n", vp,array);
    vp = name; //char 형을 원소로 하는 포인터를 대입
    printf("vp: %p name: %p \n", vp,name);
    return 0;
}
```

선택 Microsoft Visual Studio 디버깅 콘솔

vp: 00AFFC9C arr: 00AFFC9C
vp: 00AFFC8C name: 00AFFC8C

C:\Users\HOBRO\source\repos\Pointer1\ 이 창을 닫으려면 아무 키나 누르세요.

```
//보이드포인터 활용예(보이드포인터 변수에 다른 원소 형식의 포인터를 대입해보기)
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int dongyang = 20111448;
    char name = 'H';
    void *vp = 0;
    vp = dongyang; //int 형을 원소로 하는 포인터를 대입
    printf("vp: %p arr: %p \n", vp,dongyang+1);
    vp = name; //char 형을 원소로 하는 포인터를 대입
    printf("vp: %p name: %p \n", vp,name+1);
    return 0;
}
```

선택 Microsoft Visual Studio 디버깅 콘솔

vp: 0132E058 arr: 0132E059
vp: 00000048 name: 00000049

C:\Users\HOBRO\source\repos\Pointer1\ 이 창을 닫으려면 아무 키나 누르세요.