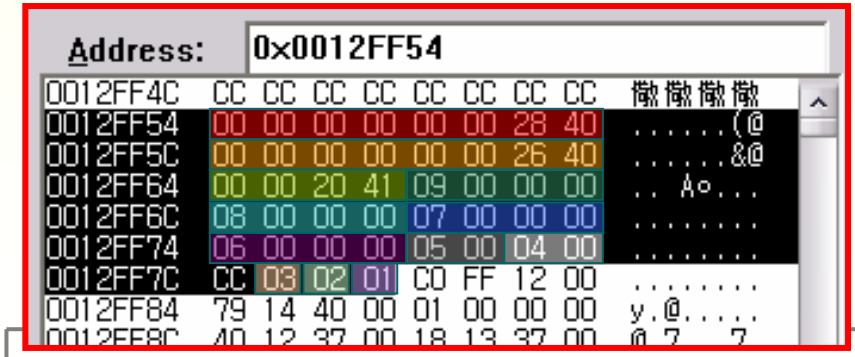
Data Types and Sizes

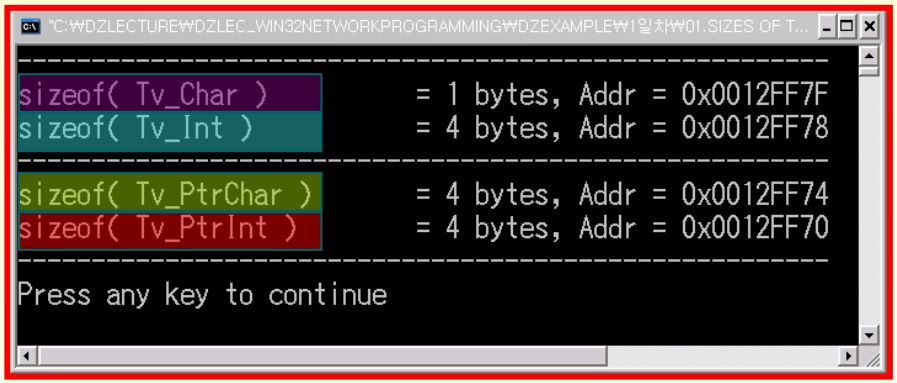
Type	Size
char, **unsigned char**, **signed char**	1 byte
short, **unsigned short**	2 bytes
int, **unsigned int**	4 bytes
long, **unsigned long**	4 bytes
float	4 bytes
double	8 bytes
long double	8 bytes

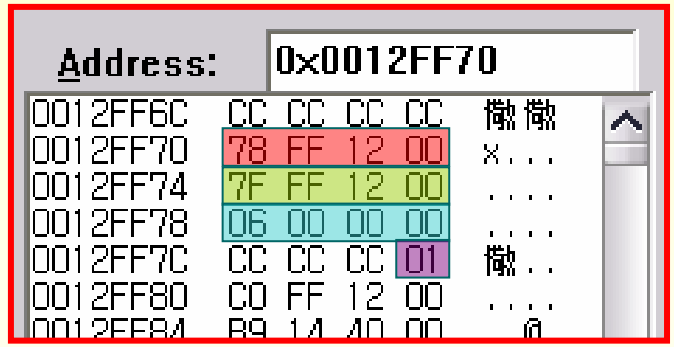

```
//local ----------------------------------------
    char              Tv_Char           = 1 ;
    unsigned char     Tv_UnsignedChar   = 2 ;
    signed char       Tv_SignedChar     = 3 ;

    short             Tv_Short          = 4 ;
    unsigned short    Tv_UnsignedShort  = 5 ;

    int               Tv_Int            = 6 ;
    unsigned int      Tv_UnsignedInt    = 7 ;

    long              Tv_Long           = 8 ;
    unsigned long     Tv_UnsignedLong   = 9 ;

    float             Tv_Float          = 10 ;

    double            Tv_Double         = 11 ;

    long double       Tv_LongDouble     = 12 ;
```



Pointer Types and Sizes

Type	Size
char*, unsigned char*, signed char*	4 bytes
short*, unsigned short*	4 bytes
int*, unsigned int*	4 bytes
long*, unsigned long*	4 bytes
float*	4 bytes
double*	4 bytes
long double*	4 bytes
user_struct*, user_class*, void*, etc*	4 bytes

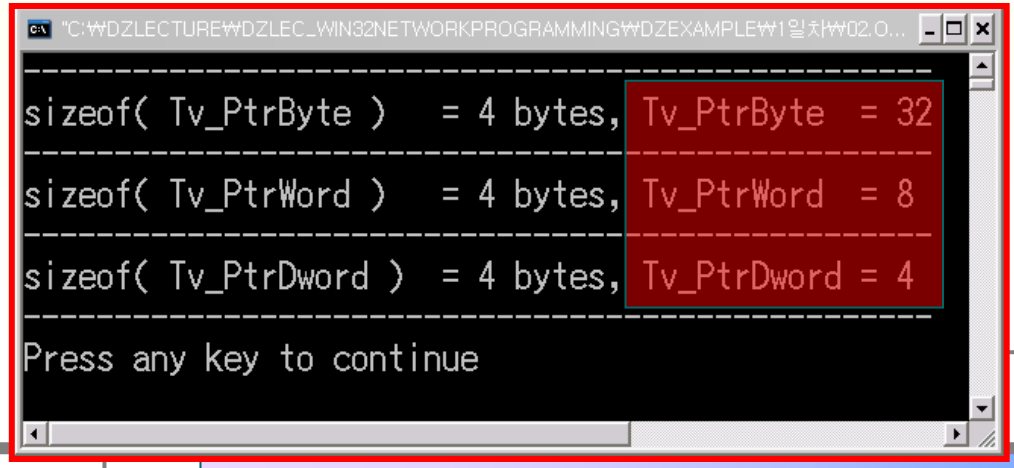





```
//local ----------------------------------------
    char        Tv_Char     = 1 ;
    int         Tv_Int      = 6 ;
    char*       Tv_PtrChar  = &Tv_Char ;
    int*        Tv_PtrInt   = &Tv_Int ;
```

Pointer 연산

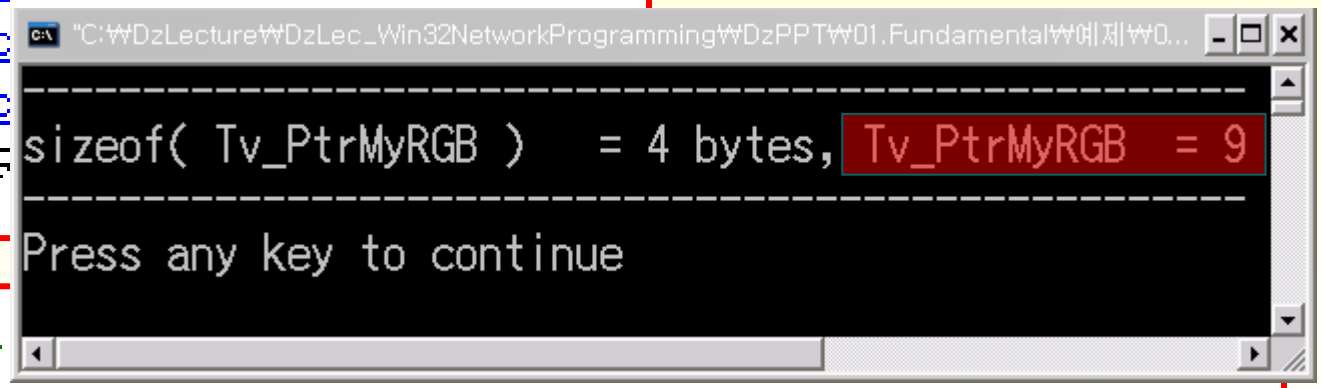
```
//local ----------------------------------------
    BYTE*           Tv_PtrByte          = 0 ;
    WORD*           Tv_PtrWord          = 0 ;
    DWORD*          Tv_PtrDword         = 0 ;

//code -----------------------------------------
    Tv_PtrByte  = Tv_PtrByte + 2 * 16 ;
    Tv_PtrWord  = Tv_PtrWord + 4 ;
    Tv_PtrDword = Tv_PtrDword + 1 ;
```



Structure Pointer

```
// structure ----------------
typedef struct St_myRGB
{
    unsigned char    Blue ;
    unsig
    unsig
}*Ptr_myR
```

St_myRGB는 char형 이 3개로 구성
즉, 3바이트 의 구조체
위의 구조체에 +3을 해주면
3+3+3 = 9가되는것이다




```
//local -
    Ptr_myRGB        Tv_PtrMyRGB          = 0 ;

//code ----------------------------------------
    Tv_PtrMyRGB = Tv_PtrMyRGB + 3 ;
```

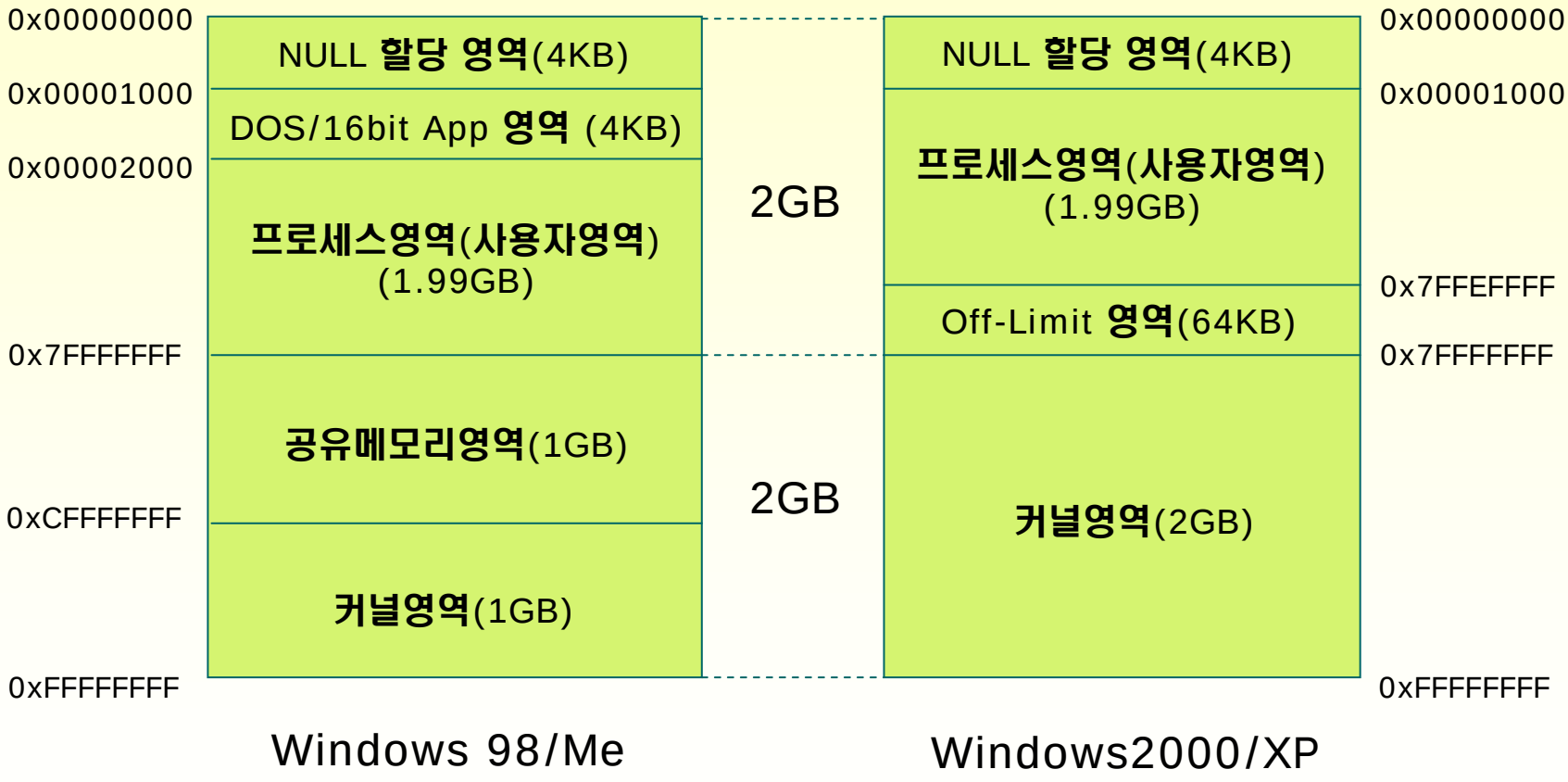
DEBUGGING

- Debugging == JOB of Finding BUGs!
- in MSVC++
  - F5 – Start Debug
  - F10 – Step Over
  - F11 – Step Into
- Debug Window
  - watch, variables, registers, memory, call stack, etc
- Debugging Tools
  - DEBUGVIEW (App)
  - OutputDebugString Win32API

예제

- DebuggingTest

32비트 윈도우 가상메모리 구조



4GB, Multi-Tasking Enable?

- PHYSICAL MEMORY/VIRTUAL MEMORY

- Virtual Memory Paging
- Memory-Mapping

- Scheduler
- Round-Robbin
- Context Switching(or Preemption)

Fundamental

argc, argv

```
void main (int argc, char *argv[])
{
  for (int i= 0 ; i < argc ; ++i)
  {
     printf (“명령줄 %d = %s\n”, i,
  argv[i]) ;
  }
}
```

argc, argv 예제 in Linux

```
#include <stdio.h>
#include <stdlib.h>

int  main(int argc, char** argv)
{
      int i;
      for(i=0; i<argc ; ++i)
      {

printf("Commandline %d=%s\n",i,argv[i]);
      }
}
```