

Topic 10: Declarations

typedef

- typedef is not as define
 - typedef int WORD; <-> #define WORD int
- How about this declaration
 - typedef int LWORD[2];
 - LWORD has a structure of two int
 - LWORD is an alias of two integer spaces
 - Common use in an embedded system for declaring fix-sized general structure
 - Security key (ex. 128 bits)
 - Packet header (ex. 160 bits)

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

General header

```
typedef char buf[2];
```

```
int main(void)
```

```
{
```

```
    buf K[3];
```

K can be considered as a 2-D array [3][2]

```
    buf *p;
```

```
    printf("sizeof K: %d\n", sizeof(K));
```

sizeof K :6

```
    printf("reference of K[0]:%p, K[1]:%p, K[2]: %p\n", &K[0], &K[1], &K[2]);
```

reference of K[0]:0xbfbfebe2, K[1]:0xbfbfebe4, K[2]: 0xbfbfebe6

```
    K[0][0]=0x10; K[0][1]=0x20; K[1][0]=0x30;
```

```
    K[1][1]=0x40; K[2][0]=0x50; K[2][1]=0x60;
```

```
    p = K;
```

```
    printf("%x, %x, %x, %x\n", *p, **p, *p[0], (*p)[0]);
```

bfbfebe2, 10, 10, 10

```
    printf("%x %x %x %x %x\n", (*p)[1], *p[1], **(p+1), *(p+1)[1], (*(p+1))[1]);
```

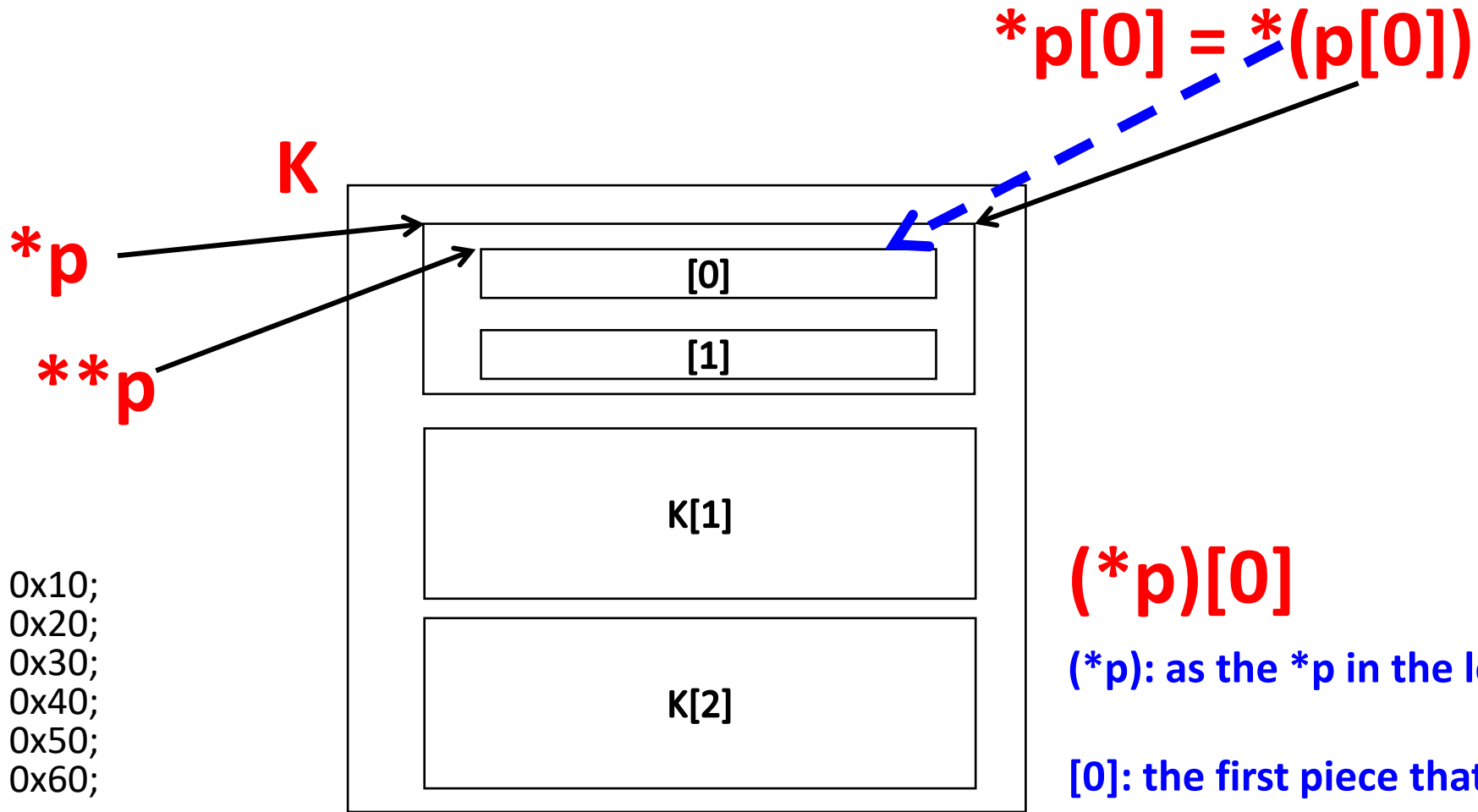
```
    return 0;
```

```
}
```

```

K[0][0] = 0x10;
K[0][1] = 0x20;
K[1][0] = 0x30;
K[1][1] = 0x40;
K[2][0] = 0x50;
K[2][1] = 0x60;

```



(*p)[1], *p[1], **(p+1), *(p+1)[1], (*(p+1))[1]

typedef

- ▶ To declare a function pointer

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

```
typedef void (*Fun)(void);
```

```
int main(void)
{
    Fun generalFun;

    generalFun = FunX;
    generalFun();

    generalFun = FunY;
    generalFun();
}
```

```
void FunX(void){ printf("Hello ");}
void FunY(void){ printf("World!");}
```

Function pointer is commonly used in protocol design

→ especially when there are multiple protocol interfaces.

W16-assignment

- Implement a number calculator, which add two positive integers

	1	2	3
+	2	4	1

	3	6	4

```
ryande-MacBook-Pro-2:TEST ryan.pan$ ./a.out
```

```
Please give a number less than 4: 1234
```

```
Please give another number less than 4: 1234
```

```
The add result: 2468
```

```
ryande-MacBook-Pro-2:TEST ryan.pan$ ./a.out
```

```
Please give a number less than 4: 123
```

```
Please give another number less than 4: 12
```

```
The add result: 135
```

```
ryande-MacBook-Pro-2:TEST ryan.pan$ ./a.out
```

```
Please give a number less than 4: 9999
```

```
Please give another number less than 4: 2
```

```
Overflow!
```

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

#define MAX 4
typedef char my_int[MAX];

void my_add (return result, input1, input2);
void my_number_print (input);
void trans_from_string (return input, user's input string);
```

You need to perform add based on the data structure my_int

Do not translate user input to an integer

You need to implement the procedure of ADD as you learned in elementary school

Final project

- Three to four students as a group
- Give me your group members today
- I will announce the project details next week
- Final project will be demonstrated by YouTube video with length no less than 20 minutes
 - Explain your design, flows, data structures, all details about your program, and teamwork