柔性数组

以可控的越界访问，来解决结构后续长度不定的情况

```
// ANSI C
struct my_st {
    int a;
    char buf[1];
};

// C99
struct my_st {
    int a;
    char buf[0];
};
```

pragma预处理

`pragma pack(n)` 以n为单位对齐
使用方法：

```
// 网络标准数据包对齐方式
#pragma pack(push)      // 保存当前对齐
#pragma pack(1)         // 设置对齐
...
#pragma pack(pop)       // 恢复对齐
```

共用体

属于语法糖

```
// 共用一个内存空间，空间大小为最大的数据类型的大小
union my_union {
    int a;
    double b;
    char c;
    ...
};

// 访问共用体的成员实质
union my_union my_u;
...
printf("%d\n", my_u.a);
printf("%d\n", *(int *)(&my_u));
```

共用体与结构体的区别

1. 共用体每个成员共同占用一份空间，大小为成员中最大的大小，结构体的大小要通过每个成员自身在内存中的对齐计算而得
2. 结构体各个成员赋值互不影响，独自占有内存空间，共用体某一成员的改变会覆盖处于内存起始位置的变量值（相互覆盖）
3. 结构体每个成员独有内存空间，按序排列，共用体成员共享内存空间，拥有相同的起始地址

枚举

```
// 默认从0开始每个成员递增1
enum my_enum {
    RED,        // 0
    GREEN,      // 1
    BLUE,       // 2
    ...
};

enum my_enmu = RED;

// 可以指定枚举成员的枚举值
enum my_enum {
    RED,        // 0
    GREEN = 2,  // 2
    BLUE,       // 3
    ...
};
```

枚举与宏的区别：

1. 枚举常量是实体中的一种，但宏不是实体
2. 枚举常量具有类型，但宏没有类型
3. 宏是预处理期进行文本替换，枚举要在编译期才能确定值