```
#define FIELD_SIZEOF(t, f) (sizeof(((t*)0)->f))
```

FIELD_SIZEOF(t, f)返回的是结构内field的sizeof.

```
struct test_struct {
    int     test_1;
    char    test_2;
    char    *ptr;
};

    printk("%u-%u-%u\n", FIELD_SIZEOF(struct test_struct, test_1),
        FIELD_SIZEOF(struct test_struct, test_2),
        FIELD_SIZEOF(struct test_struct, ptr));
```

output:

```
4-1-4
```

---

offsetof(TYPE,MEMBER)获取struct内MEMBER field的偏移(offset)

```
#include <linux/stddef.h

struct test_struct {
    int     test_1;
    char    test_2;
    char    *ptr;
};

struct test2_struct {
    int     test_1;
    char    test_2;
    char    *ptr;
} __packed;

    printk("%u-%u-%u\n", offsetof(struct test_struct, test_1),
        offsetof(struct test_struct, test_2),
        offsetof(struct test_struct, ptr));

    printk("%u-%u-%u\n", offsetof(struct test2_struct, test_1),
        offsetof(struct test2_struct, test_2),
        offsetof(struct test2_struct, ptr));
```

output:

```
0-4-8
0-4-5
```

---

container_of(ptr, type, member)

由struct内的member field而获得该struct本身的指针。

```c
#include <linux/kernel.h>

/**
 * container_of - cast a member of a structure out to the containing structure
 * @ptr:    the pointer to the member.
 * @type:   the type of the container struct this is embedded in.
 * @member: the name of the member within the struct.
 *
 */
#define container_of(ptr, type, member) ({          \
    const typeof( ((type *)0)->member ) *__mptr = (ptr);    \
    (type *)( (char *)__mptr - offsetof(type,member) );})
```

sample:

```c
struct test2_struct {
    int     test_1;
    char    test_2;
    char    *ptr;
} __packed;

    struct test2_struct test_struc;
    struct test2_struct *p_test;

    pchar = &test_struc.test_2;
    p_test = container_of(pchar, struct test2_struct, test_2);
    BUG_ON(p_test != &test_struc);
```