

2.2 结构体大小计算_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 2.2 结构体大小计算涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

2. 结构体大小计算

```
struct student
{
    char name[5];
    int id;
    short score;
};
```

打印其大小，发现和我们实际思考的不一样。

思考：为什么会出现字节对齐这种现象？
因为各个硬件平台对内存空间的访问不一样。
一些特定的平台只能从特定的地址开始。还有就是提高效率

普通变量的大小

char	short	int	long	float	double
1	2	4	4	4	8

位置偏移的大小

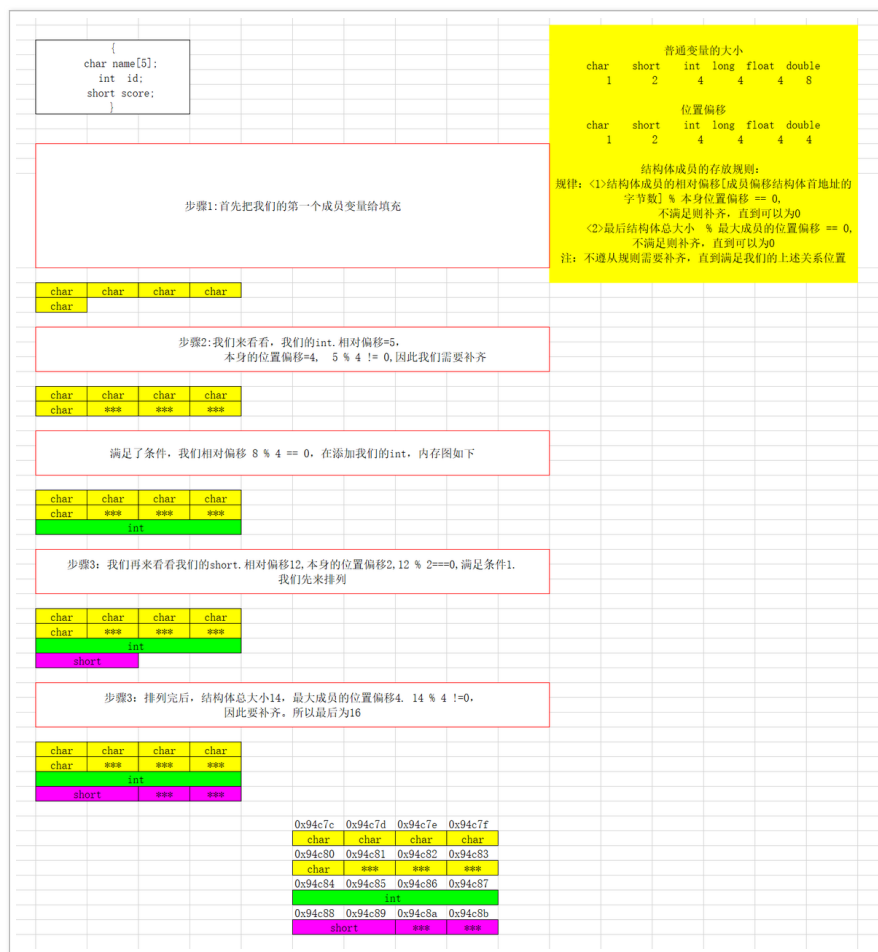
char	short	int	long	float	double
1	2	4	4	4	4

结构体成员的存放规则：

规律：<1>结构体成员的相对偏移[成员偏移结构体首地址的字节数] % 本身位置偏移 == 0，
不满足则补齐，直到可以为0

<2>最后结构体总大小 % 最大成员的位置偏移 == 0, 不满足则补齐，直到可以为0

注：不遵从规则需要补齐，直到满足我们的上述关系位置



```
#include <stdio.h>

struct student
{
    char name[5];
    int id;
    short score;
};

void output_student(struct student *sp)
{
    printf("sizeof(sp) = %d\n",sizeof(sp));
    printf("NAME\tID\tSCORE\n");
    printf("%s\t%d\t%d\n",sp->name,sp->id,sp->score);
}

int main()
{
    struct student st = {"jack",1,80};

    printf("sizeof(st) = %d\n",sizeof(st));

    output_student(&st);
    return 0;
}

sizeof(st) = 16
sizeof(sp) = 4
NAME    ID    SCORE
jack    1      80

struct student
{
    int id;
    short score;
    char name[5];
};
```