

例 1:

```
struct data {  
    int c;  
    long a;  
    char b;  
};
```

1: 结构体存储是按 **单元** 的来的, 以结构体内**最大的 类型**为单元, linux 下 单元 的大小固定是 4个字节, 可能和cpu 有关

例一中

该结构体最大的类型 是 long 类型, 所以每个单元 占 8个字节, 总共 24个字节,
当 把 long 换成 int, 最大类型 是 int 类型, 每个单元占 4个字节, 总共 12 个字节,

2: 优化

结构体是顺序 按单元存储的, 在例一中 ,

每个单元是 8个字节, int 占 4个字节 需要一个单元, 该单元剩余 4个字节, 放不下 long a, 所以需要重新分配一个单元, 最后 char b 也需要重新分配一个单元。

如果改成

```
struct data {  
    int c;  
    char b;  
    long a;  
};
```

这样 c 和 b 可以共用一个单元, 总共只需要两个单元就好了。