

应用案例

- `float f = 1.2f;` 如何将它对应的4个字节地址空间的数值以十六进制打印出来?

单片机中计量和运算单元是字节

- 数据在所有计算机中都是以二进制形式存储的，然后以**8个位**为一个单元称作“字节”，作为计量和运算处理单元，比如我们说一个文件大小是1MB，指的是1M字节；
- 在计算机中，**每个字节都有它对应的地址**，类似于门牌号，CPU通过地址来访问这个字节空间。

0x20000003	1	1	1	0	1	1	1	0
0x20000002	1	0	1	0	1	0	1	0
0x20000001	1	0	1	1	1	0	1	0
0x20000000	1	0	1	0	1	0	1	0

为什么引入数据类型

- 对于计算机硬件系统，二进制01数据没有任何意义，为了有效的组织管理这些数据，人为的规定了数据类型去解释这些二进制数据：

0x20000003	1	1	1	0	1	1	1	0
0x20000002	1	0	1	0	1	0	1	0
0x20000001	1	0	1	1	1	0	1	0
0x20000000	1	0	1	0	1	0	1	0

使用int 类型解释后代表数值：
-996519381

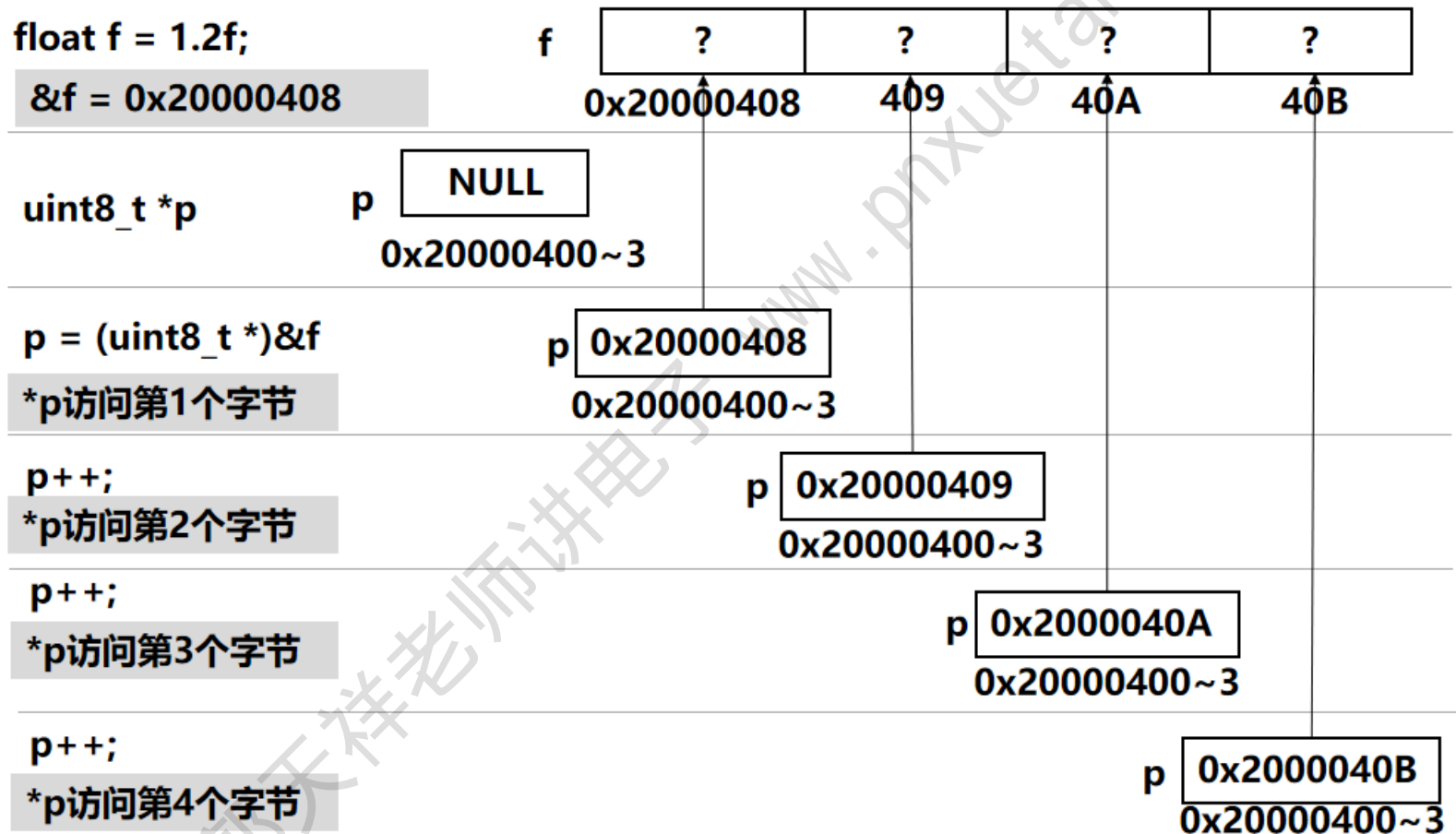
0x20000003	1	1	1	0	1	1	1	0
0x20000002	1	0	1	0	1	0	1	0
0x20000001	1	0	1	1	1	0	1	0
0x20000000	1	0	1	0	1	0	1	0

使用unsigned int 类型解释后代表数值：
3298447915

0x20000003	1	1	1	0	1	1	1	0
0x20000002	1	0	1	0	1	0	1	0
0x20000001	1	0	1	1	1	0	1	0
0x20000000	1	0	1	0	1	0	1	0

使用float类型解释后代表数值：
-1234.5678

➤ float f = 1.2f; 如何将它对应的4个字节地址空间的数值以十六进制打印出来?



应用案例

- float f = 1.2f; 如何将它对应的4个字节地址空间对应的数值以十六进制打印出来?

```
float f = 1.2f;  
uint8_t *p;  
p = (uint8_t *)&f;  
printf("*p = %#x.\n", *p);  
p++;  
printf("*p = %#x.\n", *p);  
p++;  
printf("*p = %#x.\n", *p);  
p++;  
printf("*p = %#x.\n", *p);
```

```
*p = 0x9a.  
*p = 0x99.  
*p = 0x99.  
*p = 0x3f.
```

应用案例

➤ `uint8_t value[4] = {0x9A, 0x99, 0x99, 0x3F};` 如何转换为float类型数值并打印?

0x9A	0x99	0x99	0x3F
0x20000478	479	47A	47B

● 分析:

1. 可以定义float *f, 用它来保存数组的首地址:

```
float *f = value; //此时f就保存了首地址478,
```

2. 但是此时由于类型不匹配, 需要强制类型转换:

```
float *f = (float *)value; //告诉编译器可以基于float类型去管理解释这4个字节内存空间的数据
```

3. 然后可以使用*f去访问这段内存空间了, 调用printf函数:

```
printf("*f = %.1f\n", *f);
```

THANK YOU!