

1 结构体成员的内存对齐

请按照要求，产生可执行文件，并给出每一个步骤的截图。以下要求使用 Linux 或者 Windows 的 GCC 工具。

1.1 测试一

以下包括代码和操作。

1.1.1 main1.c 文件

```
1 #include <stdio.h>
2
3 typedef struct _person0 {
4     char* firstName;
5     char* lastName;
6     char* title;
7     unsigned int age;
8 } Person;
9
10 typedef struct _person1 {
11     char* firstName;
12     char* lastName;
13     char* title;
14     unsigned short age;
15 } AltPerson1;
16
17 typedef struct _person2 {
18     char* firstName;
19     char* lastName;
20     char* title;
21     unsigned char age;
22 } AltPerson2;
23
24 int main()
25 {
26     printf("size of char*: %d\n", sizeof(char*));
27     printf("size of unsigned int: %d\n", sizeof(unsigned int));
```

```
28     printf("size of unsigned short: %d\n", sizeof(unsigned
        short));
29     printf("size of unsigned char: %d\n", sizeof(unsigned
        char));
30     printf("size of Type Person: %d\n", sizeof(Person));
31     printf("size of Type AltPerson1: %d\n", sizeof(AltPerson1));
32     printf("size of Type AltPerson2: %d\n", sizeof(AltPerson2));
33
34     Person p0;
35     AltPerson1 p1;
36     AltPerson2 p2;
37
38     printf("addr of p0: %p\n", &p0);
39     printf("addr of age in p0: %p\n", &p0.age);
40     printf("addr of p1: %p\n", &p1);
41     printf("addr of age in p1: %p\n", &p1.age);
42     printf("addr of p2: %p\n", &p2);
43     printf("addr of age in p2: %p\n", &p2.age);
44     return 0;
45 }
```

1.1.2 步骤

1. 根据以上代码产生可执行文件，并运行得到输出结果。
2. 一个 Person 类型的变量占多少字节？它的各个成员变量加起来一共占多少字节？然后，请针对 AltPerson1 和 AltPerson2，回答相应问题。
3. p0 的地址和 p1 的地址相差多少？p1 的地址和 p2 的地址呢？
4. p0、p1 和 p2 占据的空间相连吗？如果不相连，请给出间距。
5. p0 的地址和 p0.age 的地址相差多少？p1 的地址和 p1.age 的地址呢？p2 和 p2.age 呢？这其中存在什么规律？

1.2 测试二

以下包括代码和操作。

1.2.1 main2.c 文件

```
1  #include <stdio.h>
2
3  typedef struct _person0 {
4      unsigned int age;
5      char* firstName;
6      char* lastName;
7      char* title;
8  } Person;
9
10 typedef struct _person1 {
11     unsigned short age;
12     char* firstName;
13     char* lastName;
14     char* title;
15 } AltPerson1;
16
17 typedef struct _person2 {
18     unsigned char age;
19     char* firstName;
20     char* lastName;
21     char* title;
22 } AltPerson2;
23
24 int main()
25 {
26     printf("size of char*: %d\n", sizeof(char*));
27     printf("size of unsigned int: %d\n", sizeof(unsigned int));
28     printf("size of unsigned short: %d\n", sizeof(unsigned
29         short));
30     printf("size of unsigned char: %d\n", sizeof(unsigned
31         char));
32     printf("size of Type Person: %d\n", sizeof(Person));
33     printf("size of Type AltPerson1: %d\n", sizeof(AltPerson1));
34     printf("size of Type AltPerson2: %d\n", sizeof(AltPerson2));
35
36     Person p0;
37     AltPerson1 p1;
```

```
36     AltPerson2 p2;  
37  
38     printf("addr of p0: %p\n", &p0);  
39     printf("addr of firstName in p0: %p\n", &p0.firstName);  
40     printf("addr of p1: %p\n", &p1);  
41     printf("addr of firstName in p1: %p\n", &p1.firstName);  
42     printf("addr of p2: %p\n", &p2);  
43     printf("addr of firstName in p2: %p\n", &p2.firstName);  
44     return 0;  
45 }
```

1.2.2 步骤

1. 根据以上代码产生可执行文件，并运行得到输出结果。
2. 一个 Person 类型的变量占多少字节？它的各个成员变量加起来一共占多少字节？然后，请针对 AltPerson1 和 AltPerson2，回答相应问题。
3. p0 的地址和 p1 的地址相差多少？p1 的地址和 p2 的地址呢？
4. p0、p1 和 p2 各自占据连续的存储空间吗？说明理由。
5. p0 的地址和 p0.firstName 的地址相差多少？p1 的地址和 p1.firstName 的地址呢？p2 和 p2.firstName 呢？这其中存在什么规律？

1.3 测试三

以下包括代码和操作。

1.3.1 main3.c 文件

```
1 #include <stdio.h>  
2  
3 typedef struct _person0 {  
4     char* firstName;  
5     char* lastName;  
6     unsigned int age;
```

```
7     char gender;
8     char* title;
9 } Person;
10
11 typedef struct _person1 {
12     unsigned int age;
13     char* firstName;
14     char* lastName;
15     char gender;
16     char* title;
17 } AltPerson1;
18
19 typedef struct _person2 {
20     char* firstName;
21     char* lastName;
22     unsigned int age;
23     char* title;
24     char gender;
25 } AltPerson2;
26
27 int main()
28 {
29     printf("size of char*: %d\n", sizeof(char*));
30     printf("size of unsigned int: %d\n", sizeof(unsigned int));
31     printf("size of unsigned char: %d\n", sizeof(unsigned
32         char));
33     printf("size of Type Person: %d\n", sizeof(Person));
34     printf("size of Type AltPerson1: %d\n", sizeof(AltPerson1));
35     printf("size of Type AltPerson2: %d\n", sizeof(AltPerson2));
36
37     Person p0;
38     AltPerson1 p1;
39     AltPerson2 p2;
40
41     printf("in p0,\n");
42     printf("addr of firstName: %p, ", &p0.firstName);
43     printf("addr of lastName: %p, ", &p0.lastName);
44     printf("addr of age: %p, ", &p0.age);
45     printf("addr of gender: %p, ", &p0.gender);
```

```
45     printf("addr of title: %p", &p0.title);
46     printf("\n");
47
48     printf("in p1,\n");
49     printf("addr of age: %p, ", &p1.age);
50     printf("addr of firstName: %p, ", &p1.firstName);
51     printf("addr of lastName: %p, ", &p1.lastName);
52     printf("addr of gender: %p, ", &p1.gender);
53     printf("addr of title: %p", &p1.title);
54     printf("\n");
55
56     printf("in p2,\n");
57     printf("addr of firstName: %p, ", &p2.firstName);
58     printf("addr of lastName: %p, ", &p2.lastName);
59     printf("addr of age: %p, ", &p2.age);
60     printf("addr of title: %p", &p2.title);
61     printf("addr of gender: %p, ", &p2.gender);
62     printf("\n");
63     return 0;
64 }
```

1.3.2 步骤

1. 根据以上代码产生可执行文件，并运行得到输出结果。
2. 一个 Person 类型的变量占多少字节？它的各个成员变量加起来一共占多少字节？然后，请针对 AltPerson1 和 AltPerson2，回答相应问题。
3. p0 的地址和 p1 的地址相差多少？p1 的地址和 p2 的地址呢？
4. p0、p1 和 p2 各自占据连续的存储空间吗？说明理由。
5. 请画出 p0、p1 和 p2 内部各个成员变量的内存分布（需要在同一张图里面直观显示全部变量的占据的内存）。

1.4 实验报告写作要求

1. 步骤详细；

2. 表述简明;
3. 图文并茂;
4. 逻辑流畅。