

1 数组的内存分布测试

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 自然对齐的内存分布

以下包括代码和操作。

1.1.1 main1.c 文件

```

1  #include <stdio.h>
2  int main()
3  {
4      int a[4];
5      short b[2];
6      char c[2];
7      printf("c[0]\\t\\t\\tc[1]\\t\\t\\tb[0]\\t\\t\\tb[1]\\n");
8      printf("%p\\t%p\\t", &c[0], &c[1]);
9      printf("%p\\t%p\\n", &b[0], &b[1]);
10     printf("a[0]\\t\\t\\ta[1]\\t\\t\\ta[2]\\t\\t\\ta[3]\\n");
11     printf("%p\\t%p\\t", &a[0], &a[1]);
12     printf("%p\\t%p\\n", &a[2], &a[3]);
13     printf("sizeof(int): %d\\n", sizeof(int));
14     printf("sizeof(short): %d\\n", sizeof(short));
15     printf("sizeof(char): %d\\n", sizeof(char));
16     return 0;
17 }

```

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`int`、`short`和`char`型变量分别各占多少字节？
3. 代码中的数组 `a`、`b`、`c` 各占多少字节？它们的首元素地址分别是什么？它们的其余元素地址呢？

4. 请根据上面的结果，画出相应的内存分布示意图（需要在同一张图里面直观显示全部变量以及数组的占据的内存）。

1.2 非自然对齐的内存分布

以下包括代码和操作。

1.2.1 main2.c 文件

```
1 #include <stdio.h>
2 int main()
3 {
4     int a[4];
5     char c[2];
6     short b[2];
7     printf("b[0]\\t\\t\\tb[1]\\t\\t\\tc[0]\\t\\t\\tc[1]\\n");
8     printf("%p\\t%p\\t", &b[0], &b[1]);
9     printf("%p\\t%p\\n", &c[0], &c[1]);
10    printf("a[0]\\t\\t\\ta[1]\\t\\t\\ta[2]\\t\\t\\ta[3]\\n");
11    printf("%p\\t%p\\t", &a[0], &a[1]);
12    printf("%p\\t%p\\n", &a[2], &a[3]);
13    printf("sizeof(int): %d\\n", sizeof(int));
14    printf("sizeof(short): %d\\n", sizeof(short));
15    printf("sizeof(char): %d\\n", sizeof(char));
16    return 0;
17 }
```

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int、short和char型变量分别各占多少字节？
3. 代码中的数组 a、b、c 各占多少字节？它们的首元素地址分别是什么？它们的其余元素地址呢？
4. 请根据上面的结果，画出相应的内存分布示意图。

1.3 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。