**实训日志**

|  |  |  |
| --- | --- | --- |
| 学校：宝鸡文理学院 | 专业：电子科学与技术 | 学号：201691074036 |
| 日期：2018/9/5 | 星期：星期三 | 姓名：刘超飞 |
| 内容：（课程主要内容，代码/作业，收获，感悟，建议等）   1. 课程内容   day3:  位运算：  按位取反：~  每一位都进行取反运算，0变为1，1变为0  7 & = ~(1 << 2)  按位与：&  每一位都进行与，同为1则为1，否则为0  4 & 7  0000 0100  0000 0111  0000 0100  4 & 2  0000 0100  0000 0010  0000 0000  按位或：|  每一位进行或，有一个为1，则为1，否则为0  4 | 7  0000 0100  0000 0111  0000 0111  4 | 2  0000 0100  0000 0010  0000 0110  按位异或：^  每位进行异或，相同为0，不同为1  4 ^ 2 = 6  4 ^ 7 = 7  左移：<<  1 << 34  int i = 1;  i << 34;  右移：>>  87  1000 0111  0000 0111  7按位取反：  1000 0000  0000 0111 0  0100 0000  0000 0111 0  0010 0000  0000 0111 0  0001 0000  0000 0111 0  0000 1000  0000 0111 0  0000 0100  0000 0111 1  0000 0010  0000 0111 1  0000 0001  0000 0111 1  二进制形式输出：  for(i = 7; i >= 0; i--)  {  if((a & (1 << i)) > 0)  {  printf("1");  }  else  {  printf("0");  }  }  puts("");  0000 0111 >> 7 0 & 1 0  0000 0111 >> 6 0 & 1 0  0000 0111 >> 5 0 & 1 0  0000 0111 >> 4 0 & 1 0  0000 0111 >> 3 0 & 1 0  0000 0111 >> 2 1 & 1 1  0000 0111 >> 1 11 &01 1  0000 0111 >> 0 111 &001 1  for(i = 7; i >= 0; i--)  {  printf("%d", a >> i &1);  }  puts("");  //printf("%d\n", 1 << 34);/\*warning: left shift count >= width of type [enabled by default]\*/ 输出结果0  char i = 1;  //printf("%d\n", i << 34);/\*warning: left shift count >= width of type [enabled by default]\*/ 输出结果4  i = 34;  printf("%d\n", 1 << i);输出结果4  char n = 1;  printf("%d\n", n << i);输出结果4  凡是左移操作中，有变量存在，当左移位数超过类型的大小时，会对其类型大小取余数，然后移动余数位，如果  其移位的位数为字面量，则编译时，输出warning。  7 >> 2;  -7 >> 2;  0000 0111 >> 2 0000 0001  1000 0111  1111 1000  1111 1001 >> 2 1111 1110  右移，空出来的位补符号位，我们把这种右移叫（算术右移）  数组：是连续存储的基本数据类型相同的一些数构成的构造数据类型。  数组的特性：  1.存储区域是连续  2.所有元素的数据类型相同  3.数组的大小 = 元素数据类型所占字节数 \* 元素个数  4.可以通过下标访问每一个成员  5.下标从0开始，到n-1结束  数组的定义：  <元素的数据类型><数组名>[<元素个数>];  int arr[5];  数组的初始化：  <元素的数据类型><数组名>[<元素个数>] = {value};  >全部初始化：  int arr[5] = {1,2,3,4,5};  int arr[] = {1,2,3};  >部分初始化：  int arr[5] = {1,2};  >全部初始化为0  int arr[5] = {0};  数组的操作：  >数组的遍历：  p[i] <====>\*(p+i) 等价的  >数组的排序：  数组名的含义：  int arr[] = {1,2,3};  printf("arr:%p\*arr:%d\n",arr,\*arr);  int \*p = &arr[0];  printf("p:%p\*p:%d\n",p,\*p);    arr:是数组的首地址  arr:是第一个元素的地址  arr:是一个地址常量  arr:也能代表整个数组，它的数据类型是int [3], sizeof(arr) = 12, （当数组名不作为函数参数时）  当数组名作为函数参数时，它代表的是数组的首地址(指针)。  在32bit机器上，指针占4个字节。  arr = 4；  数组作为函数参数：  int print\_arr(int arr[5]， int size)  {  }  int main()  {  int arr[5] = {1,2,3};  print\_arr(arr);  }  char arr[5] = {1,2,4};  字符串是一种能特殊的字符数组。  while(\*str)  {  putchar(\*str);  str++;  }  putchar:是输出一个字符  作业：  1.实现一个strlen函数  2.实现字符串分割  二、作业  #include <stdio.h>  #include <string.h>  int main()  {  char str[]= "crose fire!";  carr\_str(str);  carr1\_str(str);  printf("\n");  }  int carr\_str(char str[],int size)  {  int len= 0;  len = strlen(str);  printf("%d\n",len);  return 0;  }  int carr1\_str(char str[])  {  while(\*str)  {  putchar(\*str);  str++;  }  }  今天的课程内容主要有数组的定义到运用，数组作函数的参数使用及其用法……这些概念理解起来十分困难，老师一次又一次的讲解依然有些地方不是很明确。但是在接下来的敲代码过程中，从遇到难题再到解决难题这中间充分体现了理论概念的重要性，经过自己的独立思考和老师的点拨勉勉强强才写好了程序，其困难程度好比愚公移山。但是通过自己不懈的努力愚公最后还是移走了泰山。  经过今天的学习我才真正的体会到了学习C语言其中的苦楚，因此接下来更要做好充足的准备迎接下一个困难——长风破浪会有时，直挂云帆济沧海。学习C语言只要有百折不挠，不耻下问的精神应该能够精通这门枯燥又不乏乐趣的课程吧！？ | | |