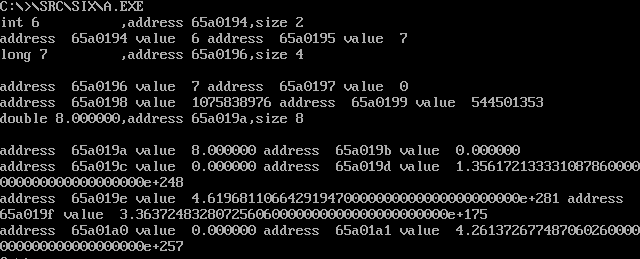
# 02020930\_综合研究研究报告

## 1

（1）写一个C程序，打印int、long、 double型变量所占的字节数、地址、各个字节的地址和内容。

int ia = 6;
  
long la = 7;
  
double da = 8;
  
main() {
  
 int i;
  
 printf("int %d ,address %lx,size %d\n", ia, (long)&ia,
  
 sizeof(int));
  
 for (i = 0; i < sizeof(int); ++i) {
  
 printf("address %lx ", (long)&ia + i);
  
 printf("value %d ", \*((&ia) + i));
  
 }
  
 printf("\n");
  
   
 printf("long %ld ,address %lx,size %d\n", la, (long)&la,
  
 sizeof(long));
  
 for (i = 0; i < sizeof(long); ++i) {
  
 if (i % 2 == 0)
  
 printf("\n");
  
 printf("address %lx ", (long)&la + i);
  
 printf("value %ld ", \*((&la) + i));
  
 }
  
 printf("\n");
  
 printf("double %lf,address %lx,size %d\n", da, (long)&da, sizeof(double));
  
 for (i = 0; i < sizeof(double); ++i) {
  
 if (i % 2 == 0)
  
 printf("\n");
  
 printf("address %lx ", (long)&da + i);
  
 printf("value %lf ", \*((&da) + i));
  
 }
  
}int ia = 6;
  
long la = 7;
  
double da = 8;
  
main() {
  
 int i;
  
 printf("int %d ,address %lx,size %d\n", ia, (long)&ia,
  
 sizeof(int));
  
 for (i = 0; i < sizeof(int); ++i) {
  
 printf("address %lx ", (long)&ia + i);
  
 printf("value %d ", \*((&ia) + i));
  
 }
  
 printf("\n");
  
   
 printf("long %ld ,address %lx,size %d\n", la, (long)&la,
  
 sizeof(long));
  
 for (i = 0; i < sizeof(long); ++i) {
  
 if (i % 2 == 0)
  
 printf("\n");
  
 printf("address %lx ", (long)&la + i);
  
 printf("value %ld ", \*((&la) + i));
  
 }
  
 printf("\n");
  
 printf("double %lf,address %lx,size %d\n", da, (long)&da, sizeof(double));
  
 for (i = 0; i < sizeof(double); ++i) {
  
 if (i % 2 == 0)
  
 printf("\n");
  
 printf("address %lx ", (long)&da + i);
  
 printf("value %lf ", \*((&da) + i));
  
 }
  
}

* 结果
* 

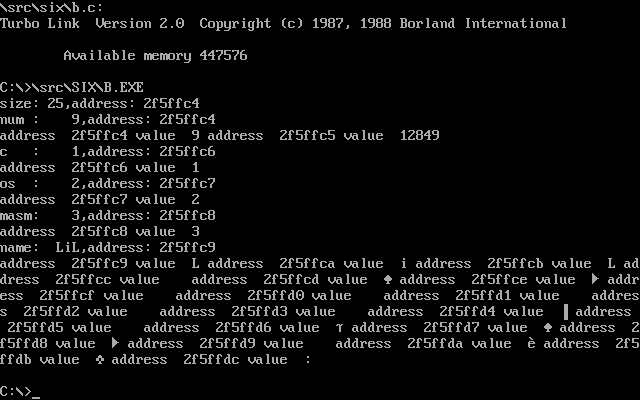
## 2

（2）写一个C程序，打印stu型变量所占的字节数、地址、各数据项地址、内容和各个字节的内容。

typedef struct
  
 {
  
 int num；
  
 unsigned char c；
  
 unsigned char osi
  
 unsigned char masm；
  
 char name［20］
  
 }stu；

提示:研究 sizeof的用法；可将任何一个变量的存储空间，看做一个数组。

typedef struct {
  
 int num;
  
 unsigned char c;
  
 unsigned char os;
  
 unsigned char masm;
  
 char name[20];
  
} stu;
  
   
main() {
  
 int i = 0;
  
 stu s;
  
 s.num = 9;
  
 s.c = '1';
  
 s.os = '2';
  
 s.masm = '3';
  
 s.name[0] = 'L';
  
 s.name[1] = 'i';
  
 s.name[2] = 'L';
  
 s.name[3] = '\0';
  
   
 printf("size: %d,address: %lx\n", sizeof(s), (long)&s);
  
 printf("num : %d,address: %lx\n", s.num, (long)&(s.num));
  
 for (i = 0; i < sizeof(int); ++i) {
  
 printf("address %lx ", (long)&(s.num) + i);
  
 printf("value %d ", \*(&(s.num) + i));
  
 }
  
 printf("\n");
  
 printf("c : %c,address: %lx\n", s.c, (long)&(s.c));
  
 for (i = 0; i < sizeof(char); ++i) {
  
 printf("address %lx ", (long)&(s.c) + i);
  
 printf("value %c ", \*(&(s.c) + i));
  
 }
  
 printf("\n");
  
 printf("os : %c,address: %lx\n", s.os, (long)&(s.os));
  
 for (i = 0; i < sizeof(char); ++i) {
  
 printf("address %lx ", (long)&(s.os) + i);
  
 printf("value %c ", \*(&(s.os) + i));
  
 }
  
 printf("\n");
  
 printf("masm: %c,address: %lx\n", s.masm, (long)&(s.masm));
  
 for (i = 0; i < sizeof(char); ++i) {
  
 printf("address %lx ", (long)&(s.masm) + i);
  
 printf("value %c ", \*(&(s.masm) + i));
  
 }
  
 printf("\n");
  
 printf("name: %s,address: %lx\n", s.name, (long)&(s.name));
  
 /\* printf("%d", sizeof(char \*)); \*/
  
 for (i = 0; i < 20; i++) {
  
 printf("address %lx ", (long)&(s.name) + i);
  
 printf("value %c ", \*((\*(&s.name)) + i));
  
 }
  
 printf("\n");
  
}

* 结果
* 

## 3

（3）写一个程序，这个程序的运行结果反映如下主题参数的存储空间与局部变量的存储空间，在函数运行后收回。

long \*f(int a);
  
long res[2];
  
long p;
  
long l;
  
main() {
  
 long \*a = f(5);
  
 p = a[0];
  
 l = a[1];
  
   
 printf("parm %d ,address %lx\n", \*(long \*)a[0], p);
  
 printf("local %d ,address %lx\n", \*(long \*)a[1], l);
  
}
  
   
long \*f(int parm) {
  
 int local = 1;
  
 local += parm;
  
 printf("parm %d address %lx\n", parm, (long)&parm);
  
 printf("local %d address %lx\n", local, (long)&local);
  
   
 res[0] = (long)&parm;
  
 res[1] = (long)&local;
  
   
 /\* printf("parm %d ,address %lx\n", \*(long \*)res[0], res[0]);
  
 printf("parm %d ,address %lx\n", \*(long \*)res[1], res[1]); \*/
  
 return res;
  
}

* 结果
* 