**考试科目名称　计算机系统基础　（A卷）**

　2014—2015学年第 1 学期　教师 袁春风 路通 苏丰 唐杰 汪亮 考试方式：开卷

系（专业）　　计算机科学与技术 　　　　　　年级　　2013　　　　班级

学号　　　　　　　　　　　　　姓名　　　　　　　　　　　成绩

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题号 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 九 | 十 | 十一 | 十二 | 十三 |
| 分数 |  |  |  |  |  |  |  |  |  |  |  |  |  |

一个C语言程序有两个源文件：main.c和test.c，它们的内容如下图所示。

/\* test.c \*/

1 extern int a[];

2 #define N 4

3 int val=0;

4 int sum()

5 {

6 int i;

7 for (i=0; i<N; i++)

8 val += a[i];

9 return val;

10 }

/\* main.c \*/

1 #include <stdio.h>

2

3 int sum();

4 int a[4]={-1,-100,2, 3};

5 extern int val;

6 void main( )

7 {

8 val=sum();

9 printf(“sum=%d\n”,val);

10 }

假设在IA-32/Linux平台上用GCC编译驱动程序处理，main.c和test.c的可重定位目标文件名分别是main.o和test.o，生成的可执行文件名为test。回答下列问题或完成下列任务。

（ 提示：IA-32为小端方式，字长为32位，即sizeof(int)=4，虚拟地址空间中的只读数据和代码段、可读写数据段都按4KB边界对齐）

一、从C语言源程序到可执行文件test的转换需要经过哪些步骤？简要说明每个步骤的工作内容。**（4分）**

**参考答案： （略）**

二、已知数组a在虚拟空间中首址为0x080496dc，则0x080496e0到0x080496e3这4个单元内容依次是什么？若a改为float型（即main.c第4行的int改为float），则这4个单元内容依次是什么？**（6分）**

**参考答案：**

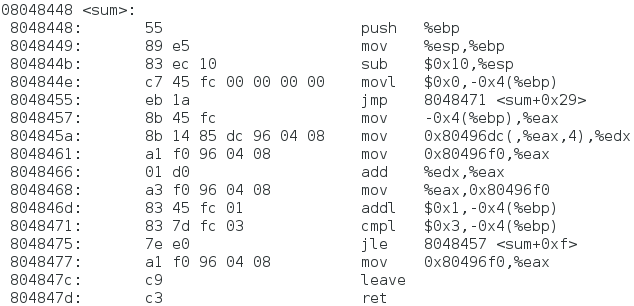
**在0x080496e0到0x080496e3这4个单元中存放的是-100，-100=-(64+32+4)=-1100100B，其对应的8位补码表示为10011100，在机器中的32位补码表示为FFFFFF9CH。因为IA-32是小端方式，因此，在0x080496e0到0x080496e3这4个单元的内容依次为：9CH、FFH、FFH、FFH。（2分）**

**若a改为float类型，则-100用IEEE 754单精度格式表示，因为-1100100B=-1.1001B×26，因此其机器数为1 10000101 100 1000 0000 0000 0000 0000，用十六进制表示为C2C80000H，因此，在0x080496e0到0x080496e3这4个单元的内容依次为：00H、00H、C8H、C2H。（4分）**

三、使用“objdump –d test”得到sum函数的反汇编结果如下，从反汇编结果可看出IA-32是CISC还是RISC？为什么？**（2分）**

**参考答案：CISC。（1分）**

**因为指令的长度参差不齐，有一个、两个、三个、四个字节等不同长度。（1分）**



四、根据sum函数反汇编结果画出其栈帧，要求分别用EBP和ESP标示栈帧底部和顶部并标出i的位置。**（4分）**

**ESP**

**sum的栈帧**

**EBP**

EBP在P中的旧值

*i*

-4

-8

-12

**参考答案：**

五、cmpl指令的执行将会影响EFLAGS寄存器

中哪些常用标志？当i=4时，sum函数中

cmpl指令的执行结果将如何影响下条jle指

令？**（10分）**

**参考答案：**

**cmpl指令通过做减法来生成标志信息，其执行将影响EFLAGS寄存器中的OF、CF、ZF和SF这几位条件标志位。（4分）**

**当i为4时，cmpl指令中的操作数M[R[ebp]-4]实际上就是4，因此，cmpl指令实际上是在以下电路中实现“4减3”的功能，也即，在电路中X=0000 0004H，Y=0000 0003H，Y’=FFFF FFFCH，Sub=1，因此，result=0000 0001H，OF=0，CF=0，ZF=0，SF=0，执行到jle指令时，根据指令功能得知是带符号整数比较，因为OF=SF，因而是大于关系，不满足jle指令的“小于或等于”关系，因而不会跳转到循环体内执行，而是跳出循环执行。（6分）**

Sub

加法器

0

1

*Y*

*Y*

*n*

*n*

*n*

*n*

*n*

*X*

零标志（ZF）

溢出标志（OF）

进位输出C

结果（result）

进位输入

*Y*’

MUX

六、地址0x804845a处的mov指令中，源操作数采用什么寻址方式？其中，EAX寄存器存放的是哪

个变量？为何比例因子为4？如何计算源操作数的有效地址？源操作数的访问过程需要经过哪些步骤？（要求从有效地址计算开始进行简要说明，包括何时判断及如何判断TLB缺失、缺页和cache缺失等，在300字以上。）**（20分，若能结合题目中给出的具体例子清楚描述IA-32/Linux中的地址转换过程，则额外加10分）**

**参考答案：**

**采用“比例变址+位移量”寻址方式。（1分）**

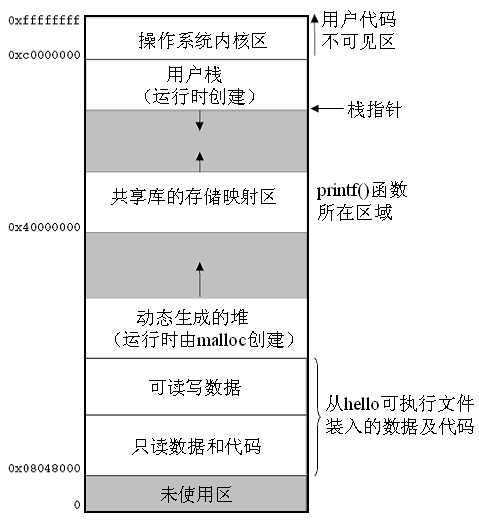
**EAX中存放的是i。（1分）**

**比例因子为4是因为a的每个元素占4B，且IA-32按字节编址。（1分）**

**有效地址EA=R[eax]\*4+0x80496dc=i\*4+0x80496dc（其中0x80496dc是数组a的首地址）（1分）**

**（略）**

七、画出test的一个进程对应的虚拟地址空间。要求根据sum函数的反汇编结果，给出只读数据和代码段的起始地址、可读写数据段的起始地址，并说明符号a、val、sum分别定义在哪个段内。**（10分）**

**参考答案：**

**0x08048000**

**0x08049000**

**根据sum反汇编结果可知只读数据和代码段的起始地址、可读写数据段的起始地址分别是0x8048000和0x8049000。（7分）**

**因为符号a和val都是初始化的全局变量，所以被定义在可读写数据段中，sum是函数名，所以被定义在只读数据和代码段。（3分）**

八、已知页大小为4KB，主存地址位数为32位。假设数据cache的数据区大小为32KB，采用4路组相联映射方式，主存块大小为64B，则主存地址如何划分？若main.c中数组a有100个元素被初始化，test.c中的N定义为100，即for循环访问的数组元素为a[0]~a[99]，执行sum函数前，数组a未调入cache但已调入主存，数组a在主存的首地址为0x406dc，则数组a的所有元素将被复制到cache的哪些组中？在执行sum函数过程中访问数组a的cache命中率是多少？**（10分）**

**参考答案：**

**数据cache共有32KB/64B=512行，共有512/4=128个cache组。因此，32位主存地址中，块内地址占6位，组索引占7位，高19位是标记。（3分）**

**数组a中有100个元素，每个元素占4个字节，因此，a共占400B，400B/64B=6.25。其首地址为0x406dc，因此，其最后一个元素的地址为0x406dc+396=0x406dc+0x18c=0x40868。（3分）**

**将0x406dc展开后为100 000 0011011 011100，将0x40868展开为100 000 0100001 101000，因此，可以看出a中的元素依次映射到cache的第11011B=27组、28组、…、100001=33组，共7组。因此，共有7次未命中，命中率为93/100=93%。（4分）**

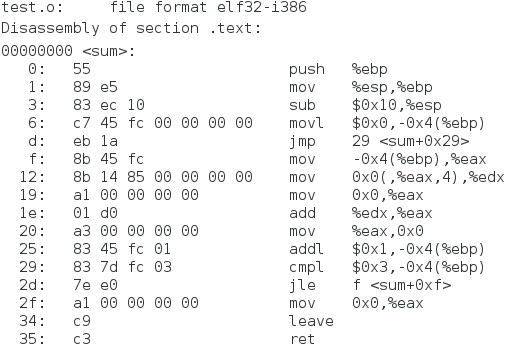
九、填写下表中各标识符的情况，说明每个标识符是否出现在test.o的符号表（.symtab节）中，如果是的话，进一步说明定义该符号的模块是main.o还是test.o、该符号的类型是全局、外部还是本地符号、该符号出现在test.o中的哪个节（.text、.data或.bss）。**（6分）**

**参考答案：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 标识符 | 在test.o的符号表中？ | 定义模块 | 符号类型 | 节 |
| a | **是** | **main.o** | **外部** | **----** |
| val | **是** | **test.o** | **全局** | **.data** |
| sum | **是** | **test.o** | **全局** | **.text** |
| i | **否** | **----** | **----** | **----** |
| N | **否** | **----** | **----** | **----** |

**（i和N各占1分，其余每3格占1分，共6分）**

十、使用“objdump –d test.o”得到sum函数的反汇编结果如下。对照在可重定位文件test.o和可执行文件test中的两个sum函数的反汇编结果，说明在哪些指令中进行了重定位（可在相应指令下方划线或给出相应指令所在的位移量）。**（4分）**

**参考答案：（一个占1分）**

十一、为什么在main.c的开头需加“#include <stdio.h>”？ 为什么main.c中没有定义printf()函数，也没它的原型声明，但main()函数引用它时没有发生错误？为什么printf( )函数中未指定输出文件名，但执行test程序后会在屏幕上显示字符串？**（4分）**

**参考答案：**

**因为main.c中使用了标准I/O函数printf()，这个函数的原型说明在stdio.h中，所以应该在main.c的开头加“#include <stdio.h>”。（1分）**

**因为在main.c的开头加了“#include <stdio.h>”，因此，在预处理后，printf()函数的原型说明被嵌入到了main.c中，因此，在main函数中引用printf函数不会发生错误。（2分）**

**因为printf函数隐含的输出文件名是stdout，即标准输出，因此，执行test程序后会在屏幕上显示字符串。（1分）**

十二、main函数中的printf语句所对应的指令为“call 8048300”。简述从执行该指令开始到在屏幕上显示出“sum=-96”为止的大概过程。要求300字以上。**（10分，若能结合题目中给出的具体例子清楚描述IA-32/Linux中的异常/中断处理机制，则额外加10分）**

**参考答案：**

**（略）**

十三、已知test程序用于解决求和问题。请运用计算机系统层次结构概念，简要说明如何使用通用计算机系统解决求和问题。与使用普通计算器进行求和的方法比较，使用通用计算机系统解决问题的最大特点是计算机系统采用什么工作方式？**（10分）**

**参考答案：**

**（略）**