|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2021级信息安全02** | | | **姓名** | **于璐** |
| **实验题目** | X86汇编子程序 | | | | | |
| **实验时间** | **2023/10/28** | | **实验地点** | **DS3305** | | |
| **实验成绩** |  | | **实验性质** | **☑验证性 □设计性 □综合性** | | |
| 教师评价：  ☑算法/实验过程正确； ☑源程序/实验内容提交； ☑程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； ☑报告规范；  其他：  评价教师签名： 杨吉云 | | | | | | |
| 1. 实验目的 2. 验证X86汇编子程序结构 3. 验证X86汇编子程序参数传递方式运算 4. 验证X86汇编子程序运行时堆栈的变化 | | | | | | |
| 1. 实验项目内容 2. 编写两个数的加法子程序，参数传递方式为寄存器传参、变量传参数、变量地址传参数，在主程序传递参数调用子程序，记录堆栈变化情况。 3. 编写子程序实现对数组求和、平均值，在主程序中传递数组地址和数组个数调用子程序，记录堆栈变化情况。 4. 编写子程序将其中字母大小写反转、将其中的数字转换为二进制存储到数组中。主程序定义字符串为“asdterDXRT345\*!@as4”，以及存储数组，通过传递字符串地址和数组地址调用子程序，记录堆栈变化情况； | | | | | | |
| 1. 实验过程或算法（源程序）   1.  寄存器传参:  111  变量传参数:  222  变量地址传参数:  333  2.  .386  .model flat,stdcall  .stack  include windows.inc  include user32.inc  include kernel32.inc  includelib user32.lib  includelib kernel32.lib  .data  arrayA dd 10000h,20000h,30000h,30000h,20000h,10000h  sum dd ?  ave dd ?  .code  sumArray proc  push edx  push ebp  mov ebp,esp  mov ecx,[ebp+12]  mov esi,[ebp+16]  mov ebx,ecx ;ebx暂存平均值  mov eax,[esi] ;eax存数组的和  sub ecx,1  L1:  add esi,4  add eax,[esi]  loop L1  mov ecx,ebx ;将数组长度存到ecx  mov ebx,eax ;数组和在ebx  mov edx,0  div ecx ;平均值在eax  pop ebp  pop edx  ret  sumArray endp  main proc  mov esi,offset arrayA  push esi  mov ecx,lengthof arrayA  push ecx  invoke sumArray  mov sum,ebx  mov ave,eax  invoke ExitProcess,0    main endp  end main  3.  .386  .model flat,stdcall  .stack  include windows.inc  include user32.inc  include kernel32.inc  includelib user32.lib  includelib kernel32.lib  .data  stringA BYTE "asdterDXRT345\*!@as4"  arrayS BYTE 20 dup(?)  .code  stringRe proc  push ebp  mov ebp,esp  mov esi,[ebp+16] ;esi指向的是字符串  mov ecx,[ebp+12]  mov edi,[ebp+8] ;edi指向数组    ;mov esi,offset stringA  ;mov ecx,lengthof stringA  ;mov edi,offset arrayS  sub esi,type stringA  L1:  ;循环遍历一遍  add esi,type stringA  mov al,[esi]  cmp al,'a' ;有可能是小写  jae L2  cmp al,'A' ;65-90是大写字母的范围  jae L3  cmp al,'0' ;是数字  jae L4  loop L1  L2:  cmp al,'z'  ja L5  ;如果是小写  sub al,32  mov [esi],al  ;add esi,type stringA  loop L1  L3:  cmp al,'Z'  ja L5  add al,32  ;如果是大写  mov [esi],al  ;add esi,type stringA  loop L1  L4:  cmp al,'9'  ja L5  ;如果是数字  sub al,'0'  mov [edi],al  add edi,type arrayS  loop L1  L5:  loop L1  ret  stringRe endp  main proc  mov esi,offset stringA  push esi  mov ecx,lengthof stringA  push ecx  mov edi,offset arrayS  push edi  invoke stringRe  invoke ExitProcess,0    main endp  end main | | | | | | |
| 1. 实验结果及分析和（或）源程序调试过程   实验结果:  **1.计算10000+20000**  **1.1寄存器传参**  堆栈区域:  寄存器堆栈  运行前内存:  寄存器传递1  运行后内存:  寄存器传递2  观察到和30000被写入到了内容中  **1.2变量传参数**  堆栈区域:  观察到变量的值10000h和20000h被压入堆栈  压入值  运行前内存:  1运行前内容  运行后内存:  1运行后你内存  观察到和30000被写入到了内容中  **1.3变量地址传参数**  观察到变量在内存中的地址403000和403004依次被压入堆栈中,同时用到寄存器ebx和edx,将这两个寄存器的值也压入堆栈中  3_堆栈1  执行完pop edx之后:  popedx之后  执行完pop ebx之后:  popebx  子过程执行完之后,退出程序:  tuichu  对于堆栈中的值,压入堆栈,栈顶指针减小,pop执行之后,栈顶指针增加  运行前内存:  3_运行前内存  运行后内存:  1111111  观察到和30000h被写入到了内容中,程序执行符合预期!  **2.将数组地址和数组个数分别push到栈中**  1111111111111111  \*\*\*\*\*\*\*\*\*\*\*\*\*\*程序执行\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  运行前内存中的内容:  运行前内存中只有初始化的数组的六个值,为了方便观察结果,我把它们设置为:  1  运行前在内存中的.data数据如下:  1  运行后内存中的内容:  2  观察到程序运行之后,和C0000h和平均值20000h均成功写入内存,程序执行正确  **3.第三题堆栈变化情况**  将字符串地址,字符串长度以及保存二进制数字的数组的地址压入堆栈中  堆栈  程序执行前内存中的内容,内存中的值为字符串:asdterDXRT345\*!@as4  1  程序执行后内存中的内容:  2  **观察到内存中,所有的字母大小写均发生了改变,大小写字母转换完成(右侧红框标出)**  所有的四个数字3454也变为二进制的形式存储在了内存地址403013开始的数组位置,**在内存中存储为03h,04h,05h,04h,以二进制的方式存储在了数组里面,程序执行与预期一致**  **源程序调试过程遇到的问题及解决方案:**   * 在第2题求平均值的过程中,发生了溢出错误,程序无法执行,将隐含的被除数的高32位edx寄存器中值压入栈中,并将edx寄存器的值清零,然后再调制执行除法(和除以数组长度)就可以运行了,在子过程退出时执行pop edx. | | | | | | |