1. 选择填空 1-20: B D A B D 6-10: B B B C D 11-15: C A B C A 16-20: D A C A B
2. 填空题

21：80H或10000000B 22：0.0001 1001 1001 1001 1001 1001 23：编译器/汇编程序

24：EAX/累加器 25：invoke ExitProcess, 0 26：4

27：XOR AX, 0000 0010 0000 0100B 或0204H 28：AND AX，0FH

29：返回地址 30：局部变量

1. 判断题

31-40： X X X√√ X X X X √

1. 简答题

|  |  |
| --- | --- |
| 41：  低地址  ESP --> | 现场寄存器ECX |
|  | 局部变量v |
| EBP--> | 主程序EBP |
|  | 返回地址 |
| 高地址 | 参数n |

42：在寄存器或全局变量传递参数时，两者没有区别

其他:宏是替换（虚实结合），子程序是堆栈传递参数（也支持寄存器、全局变量）

1）.参数声明：宏的参数没有类型。 而子程序的参数要声明类型 (1分)

2.）参数形式：宏的参数可以是常量、变量、表达式、寄存器、指令或部分指令等，很灵活；子程序的参数可以是传值或传地址。形式可以是常量、变量、常量表达式、寄存器 (1分)

3.）参数传递：宏是替换，用实参代替形参；子程序通过堆栈传递参数，按照倒序压入堆栈，而且返回后需要堆栈平衡 (1分)

4.）参数使用：宏体中直接使用参数名访问参数，对参数的改变直接影响到主程序。而子程序通过[EBP+8]、[EBP+12]、……等分别访问第1、2、……个参数，对传值参数的修改不影响主程序，对传地址的参数修改会影响主程序的值。(1分)

注：学生回答，意思有即可得分。

43： ADD EDX,8000 **0000**H 或 BT EDX，31

RCL EAX，1

RCL EDX，1

44: LEA ESI,S1

LEA EDI,S2

CLD ；缺少也不扣分

MOV ECX,10

REPZ CMPSB

45: 注：大致意思有即可，最后一行可没有

|  |  |  |
| --- | --- | --- |
|  | 传值 X | 传Y的地址 |
| 参数传递 | PUSH X | PUSH OFFSET Y |
| 参数使用 | 直接使用 X参与数据处理 | MOV 寄存器R，Y 采用【R+……】访问 |
| 变化影响 | 不影响主程序中的值 | 影响主程序中的值 |
| 本质 | 在堆栈中的拷贝 | 在堆栈中存放指针 |

1. 程序分析题

46：8A35H CF=0 OF=1 SF=1

47： ADDR arr[esi \* 4] DEC ECX arr[esi\*4+4]

XCHG eax,arr[esi\*4+4] POP ecx arr[esi\*4]

48： add ecx,3 ;计算本条件i对应的跳转表元素下标

**ja 03D3DAEh edx\*4**

**003D3DA4、003D3D98、003D3D8C、003D3DAE、003D3D68、003D3D74、003D3D80**

注： add ecx,3 这儿只要出现ecx就不扣分了。 JA/JG都不扣分。  
 最后的7个空，顺序错了也不扣分。

1. 编程题

49：设 全局dword变量 n为元素个数。 注：本题功能正确即可，不用一样

iq proc (proc/endp 1分)

push ebp (1分)

mov ebp,esp (1分)

push ebx

push edi (现场保护1分)

mov eax,0 ;插入了0个元素

.if n==16

jmp iqover

.endif ( 判断满返回0，正确得2分)

mov ebx,[ebp+8] (取arr地址，1分)

mov edi,[ebp+12] (取ip地址，1分)

add ebx,[edi] (计算当前队尾地址2分)

mov eax,[ebp+16] (取val值 1分)

mov [ebx],eax (插入队尾2分)

push edi ;左边的3句可替换为2句 ；调整队尾; 3个都给分吧

call incp ; add dword ptr[edi],4 ;inc dword ptr [EDI] ；2分

add esp,4 ; and dword ptr[edi],63 ;and dword ptr[edi],0FH

inc n (队列元素个数加1，1分)

mov eax,1 ;插入了一个元素 (返回值 1分)

iqover: pop edi

pop ebx (恢复现场1分)

leave (1分，可以用mov esp,ebp / pop ebp)

ret (1分)

iq endp

1. 附加题

50. 答： OS可以为每个任务设定一个时间片，利用时钟中断，当本任务时间片（中断次数到）到时，会暂停当前任务，由调度程序执行另一优先级最高的任务。从而实现多任务同时运行。（3分）

关键硬件设备的实时监控，可通过传感器采集其各类模拟信息，经过模数转换后，向CPU发出中断请求，CPU可实时响应本中断，在中断处理程序里完成数据采集和状态判断，并可对外发出控制信息，实现实时监控。不会由于CPU忙于其他任务导致关键硬件设备不能得到实时监控。（2分）

注：意思到即可，有自己创新想法直接满分

51. 答：现代Intel 64位CPU里有FPU和SSE指令处理系统。FPU采用80位的寄存器堆栈进行浮点处理，m尾数64位、e阶码15位、1个符号位S；可支持单精度、双精度、扩展双精度运算。而且现代CPU还有8个128位的XMM寄存器，支持SSE指令，可进行单精度、组合或标量双精度浮点运算。（1分）

但C语言浮点数的存储采用IEEE754标准格式，有3种浮点类型：float类型为32位（S1 e8 m23），double为64位（S1 e11 m52）,long double 为128位（S1 e15 m112）。这与计算机中FPU不符。另外由于我们经常用到的小数0.1等都不能用二进制精确表示，所以float、甚至double类型都会损失精度，当然位数越多精度越高。（1分）

因此建议C语言的浮点类型为80位和128位两种。C的浮点运算为FPU和SSE指令集支持的浮点运算即可。C的浮点运算库函数应以FPU和SSE的硬指令为基础进行其他运算或功能的实现。 (3分)

注：意思到即可，有自己创新想法直接满分