考点16

程序的机器级代码表示

王道考研/CSKAOYAN.COM

31

考点16：程序的机器级代码表示

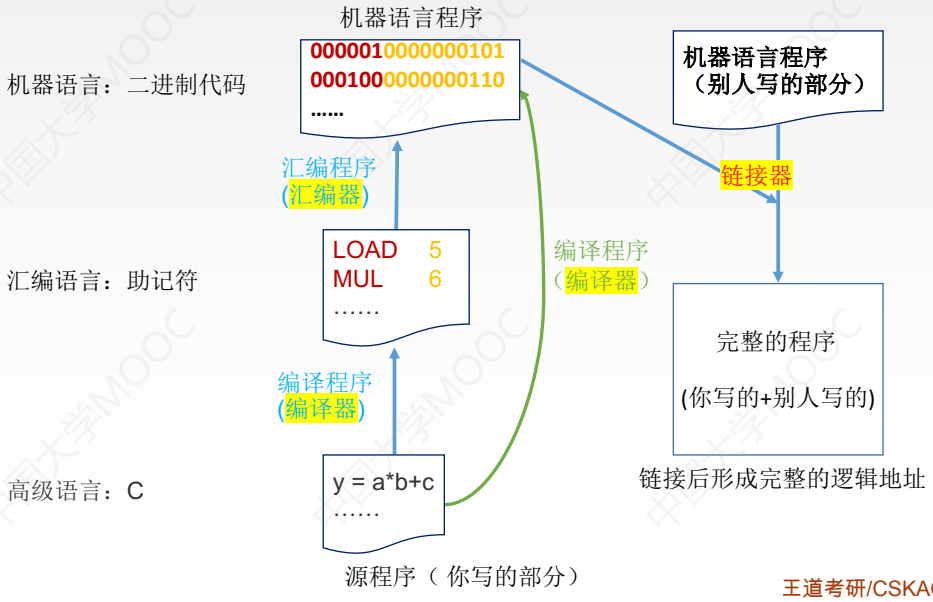
2009	2010	2011	2012	2013	2014	2015
2016	• 综合题44 2017	2018	• 综合题45 2019	2020	2021	2022

历年考频：小题×0、综合题×2

王道考研/CSKAOYAN.COM

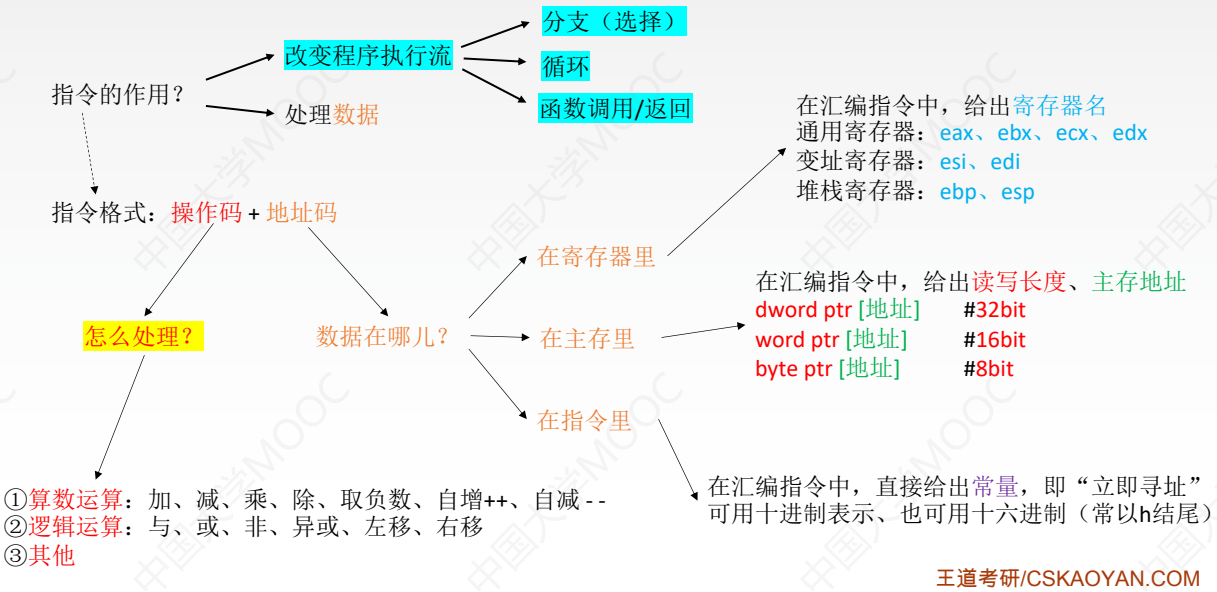
32

编译器、汇编器、链接器



33

汇编语言指令总结



34

常见的算数运算指令

destination：目的地（d 目的操作数）
source：来源地（s 源操作数）

目的操作数 d 不可以是常量

功能	英文	汇编指令	注释
加	add	add d,s	#计算d+s，结果存入d
减	subtract	sub d,s	#计算d-s，结果存入d
乘	multiply	mul d,s imul d,s	#无符号数d*s，乘积存入d #有符号数d*s，乘积存入d
除	divide	div s idiv s	#无符号数除法 edx:eax/s，商存入eax，余数存入edx #有符号数除法 edx:eax/s，商存入eax，余数存入edx
取负数	negative	neg d	#将d取负数，结果存入d
自增++	increase	inc d	#将d++，结果存入d
自减 - -	decrease	dec d	#将d--，结果存入d

王道考研/CSKAOYAN.COM

35

常见的逻辑运算指令

destination：目的地（d 目的操作数）
source：来源地（s 源操作数）

目的操作数 d 不可以是常量

功能	英文	汇编指令	注释
与	and	and d,s	#将 d、s 逐位相与，结果放回d
或	or	or d,s	#将 d、s 逐位相或，结果放回d
非	not	not d	#将 d 逐位取反，结果放回d
异或	exclusive or	xor d,s	#将 d、s 逐位异或，结果放回d
左移	shift left	shl d,s	#将d逻辑左移s位，结果放回d（通常s是常量）
右移	shift right	shr d,s	#将d逻辑右移s位，结果放回d（通常s是常量）

王道考研/CSKAOYAN.COM

36

其他指令

汇编指令	注释
mov d, s	#将源操作数s复制到目的操作数d所指的位置
jmp <地址>	# PC无条件转移至 <地址>
cmp a,b jxxx <地址>	#比较a和b两个数。本质上是执行 a-b，并生成标志信息 OF、ZF、CF、SF #若 cmp 指令比较的 a和b 满足条件，则PC转移至 <地址>
loop <地址>	# ecx--，若ecx!=0，则PC跳转至<地址>。通常用于实现计数型循环（循环n次）
test a,b	#将a、b按位与，并生成标志信息 OF、ZF、CF、SF、PF（奇偶校验位）

条件转移指令一般要和 cmp 指令一起使用

循环开始前用 ecx 记录剩余循环次数

同样可以用于条件转移（简要了解即可）

王道考研/CSKAOYAN.COM

37

if-else 分支结构常见代码框架

```
if (a>b){
  c=a;
} else {
  c=b;
}
```

“否”命题

\>

<=

if 部分的逻辑

else 部分的逻辑

jxxx 指令的条件是if条件取“否”

```
mov eax,7       #假设变量a=7,存入eax
mov ebx,6       #假设变量b=6,存入ebx
cmp eax,ebx     #比较变量a和b
jle NEXT        #若a≤b，转移到NEXT:
mov ecx,eax     #假设用ecx存储变量c，令c=a
jmp END         #无条件转移到END:
NEXT:
mov ecx,ebx     #假设用ecx存储变量c，令c=b
END:
```

if-else分支结构：

① cmp a,b
② jxxx <⑤>
③ if 部分的逻辑
④ jmp <⑥>
⑤ else 部分的逻辑
⑥ if-else完结

38

for/while 循环结构常见代码框架

```
int result = 0;
for(int i=1;i<=100;i++) {
  result +=i;
} //求 1+2+3+...+100
```

for/while循环结构：
①循环前的初始化
②cmp、jxxx (否命题)判断是否直接跳过循环？
③循环主体
④ cmp、jxxx (同循环条件)判断是否继续循环？

```
mov eax,0     #用 eax 保存 result，初值为0
mov edx,1     #用 edx 保存 i，初始值为1
cmp edx,100   #比较 i和100
jg L2         #若i>100，转跳到 L2 执行
L1:           #循环主体
add eax,edx   #实现 result +=i
inc edx       #inc 自增指令，实现 i++
cmp edx,100   #i和100
jle L1        #若 i<=100，转跳到 L1 执行
L2:           #跳出循环主体
```

①循环前的初始化
②是否直接跳过循环？
③循环主体
④是否继续循环？

39

补充：loop指令实现循环

```
for(int i=500;i>0;i--) {
   做某些处理;
} //循环500轮
```

for/while循环结构：（loop指令实现）
①循环前 ecx 初始化
②循环主体
③ loop <地址>

```
mov ecx,500     #用ecx作为循环计数器
Looptop:        #循环的开始
...
做某些处理
...
loop Looptop    #ecx--，若ecx!=0，跳转到Looptop
```

使用loop 指令实现循环，必须以 ecx 作为循环计数器

等价于：
dec ecx
cmp ecx,0
jne Looptop

理论上，能用 loop 指令实现的功能一定能用条件转移指令实现

其他loop指令：loopx —— 如 loopnz, loopz
loopnz——当 ecx!=0 && ZF==0 时，继续循环
loopz——当 ecx!=0 && ZF==1 时，继续循环

40

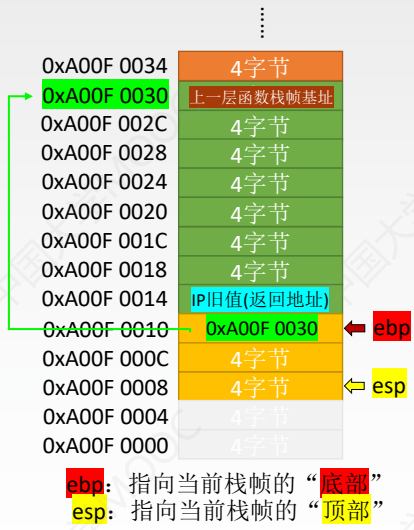
函数调用相关指令

函数调用指令：call <函数名>
①将IP旧值压栈保存（保存在函数的栈帧顶部）
②设置IP新值，无条件转移至被调用函数的第一条指令

返回指令：ret
从函数的栈帧顶部找到IP旧值，将其出栈并恢复IP寄存器

Push 🐶　#先让esp减4，再将 🐶 压入
Pop 🐴　#栈顶元素出栈写入 🐴，再让 esp加4

注1：🐶 可以是立即数、寄存器、主存地址
注2：🐴 可以是寄存器、主存地址



ebp：指向当前栈帧的“底部”
esp：指向当前栈帧的“顶部”

指令：enter 等价
push ebp　#保存上一层函数的栈帧基址（ebp旧值）
mov ebp,esp　#设置当前函数的栈帧基址（ebp新值）

指令：leave 等价
mov esp, ebp　#让esp指向当前栈帧的底部
pop ebp　#将esp所指元素出栈，写入寄存器ebp

王道考研/CSKAOYAN.COM

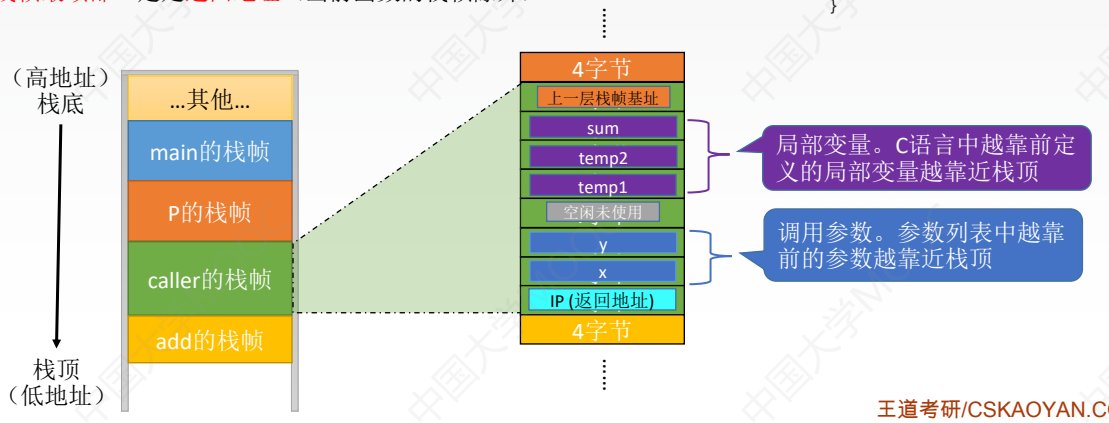
41

栈帧的内部结构

- 每个栈帧大小规定为 16B 的整数倍（当前函数的栈帧除外），因此栈帧内可能出现空闲未使用的区域。
- 通常将局部变量集中存储在栈帧底部区域
- 通常将调用参数集中存储在栈帧顶部区域
- 栈帧最底部一定是上一层栈帧基址（ebp旧值）
- 栈帧最顶部一定是返回地址（当前函数的栈帧除外）

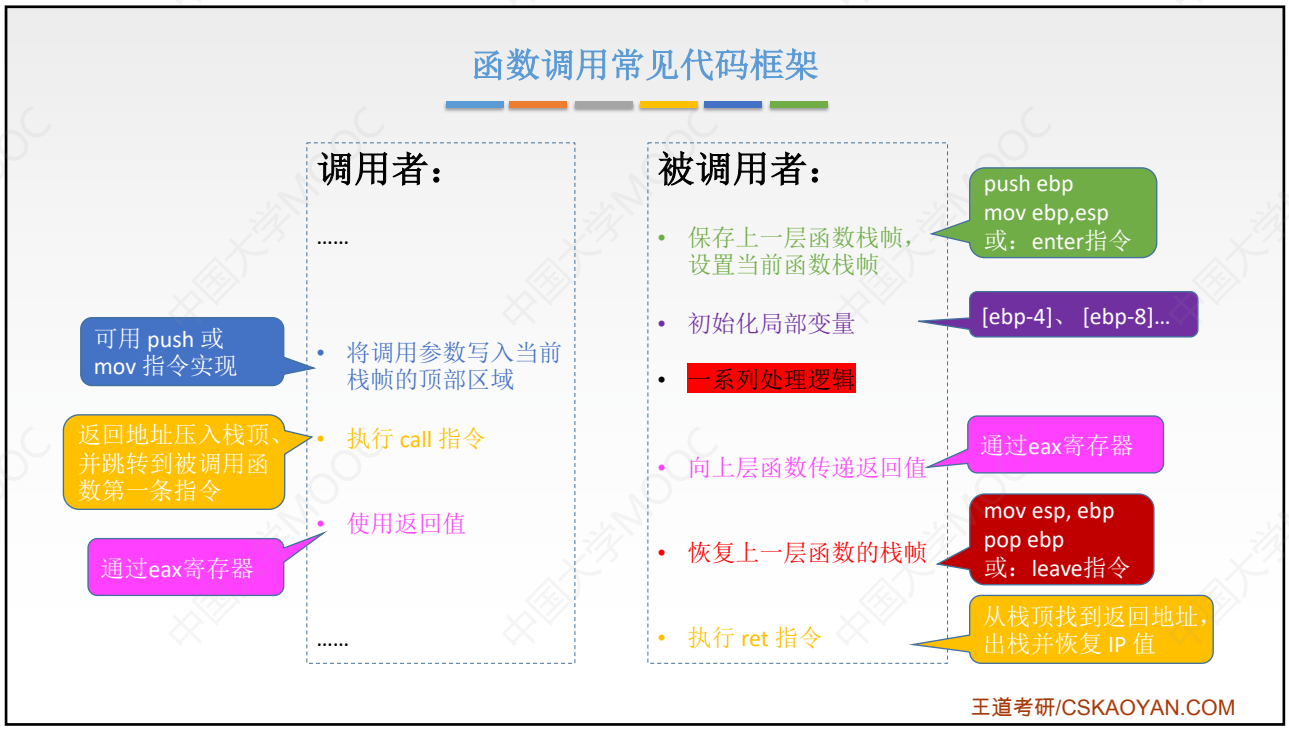
```
int caller(){
  int temp1=125;
  int temp2=80;
  int sum=add(temp1,temp2);
  return sum;
}

int add(int x, int y){
  return x+y;
}
```

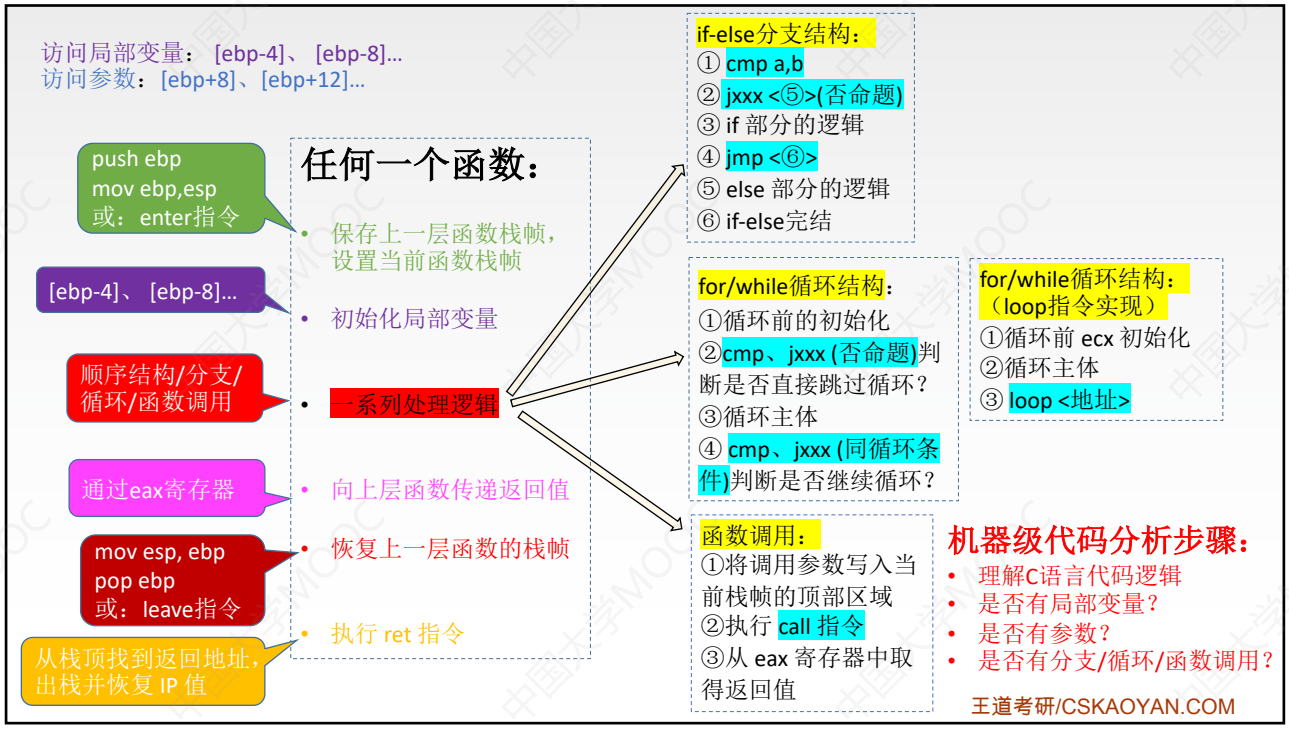


王道考研/CSKAOYAN.COM

42


函数调用常见代码框架
调用者：
.....
将调用参数写入当前栈帧的顶部区域
执行 call 指令
使用返回值
.....
可用 push 或 mov 指令实现
返回地址压入栈顶、并跳转到被调用函数第一条指令
通过eax寄存器
被调用者：
保存上一层函数栈帧，设置当前函数栈帧
初始化局部变量
一系列处理逻辑
向上层函数传递返回值
恢复上一层函数的栈帧
执行 ret 指令
push ebp
mov ebp,esp
或：enter指令
[ebp-4]、 [ebp-8]...
通过eax寄存器
mov esp, ebp
pop ebp
或：leave指令
从栈顶找到返回地址，出栈并恢复 IP 值
王道考研/CSKAOYAN.COM

43


访问局部变量： [ebp-4]、 [ebp-8]...
访问参数： [ebp+8]、[ebp+12]...
任何一个函数：
保存上一层函数栈帧，设置当前函数栈帧
初始化局部变量
一系列处理逻辑
向上层函数传递返回值
恢复上一层函数的栈帧
执行 ret 指令
push ebp
mov ebp,esp
或：enter指令
[ebp-4]、 [ebp-8]...
顺序结构/分支/循环/函数调用
通过eax寄存器
mov esp, ebp
pop ebp
或：leave指令
从栈顶找到返回地址，出栈并恢复 IP 值
if-else分支结构：
① cmp a,b
② jxxx <⑤>(否命题)
③ if 部分的逻辑
④ jmp <⑥>
⑤ else 部分的逻辑
⑥ if-else完结
for/while循环结构：
①循环前的初始化
②cmp、jxxx (否命题)判断是否直接跳过循环？
③循环主体
④ cmp、jxxx (同循环条件)判断是否继续循环？
for/while循环结构：（loop指令实现）
①循环前 ecx 初始化
②循环主体
③ loop <地址>
函数调用：
①将调用参数写入当前栈帧的顶部区域
②执行 call 指令
③从 eax 寄存器中取得返回值
机器级代码分析步骤：
理解C语言代码逻辑
是否有局部变量？
是否有参数？
是否有分支/循环/函数调用？
王道考研/CSKAOYAN.COM

44

```
int f1(int n){
  if (n>1)
    return n*f1(n-1)
  else
    return 1;
}
```

访问参数：[ebp+8]、[ebp+12]...

if-else分支结构：
① cmp a,b
② jxxx <⑤>(否命题)
③ if 部分的逻辑
④ jmp <⑥>
⑤ else 部分的逻辑
⑥ if-else完结

函数调用：
①将调用参数写入当前栈帧的顶部区域
②执行 call 指令
③从 eax 寄存器中取得返回值

2019年真题

45.（16 分）已知 $f(n)=n!=n\times(n-1)\times(n-2)\times\cdots\times2\times1$，计算 $f(n)$ 的 C 语言函数 f1 的源程序（阴影部分）及其在 32 位计算机 M 上的部分机器级代码如下：

```
int  f1(int n){
1    00401000   55              push  ebp
     ...        ...             ...
     if(n>1)
11   00401018   83 7D 08 01     cmp  dword ptr [ebp+8],1
12   0040101C   7E 17           jle  f1+35h (00401035)
     return n*f1(n-1);
13   0040101E   8B 45 08        mov  eax, dword ptr [ebp+8]
14   00401021   83 E8 01        sub  eax, 1
15   00401024   50              push  eax
16   00401025   E8 D6 FF FF FF  call  f1 ( 00401000)
     ...        ...             ...
19   00401030   0F AF C1        imul  eax, ecx
20   00401033   EB 05           jmp  f1+3Ah (0040103a)
     else return 1;
21   00401035   B8 01 00 00 00  mov  eax,1
     }
     ...        ...             ...
26   00401040   3B EC           cmp  ebp, esp
     ...        ...             ...
30   0040104A   C3              ret
```

其中，机器级代码行包括行号、虚拟地址、机器指令和汇编指令，计算机 M 按字节编址，int 型数据占 32 位。请回答下列问题：

王道考研/CSKAOYAN.COM

47

43.（13 分）已知 $f(n)=\sum_{i=0}^{n}2^i=2^{n+1}-1=\overbrace{11\cdots1}^{n+1位}B$

```
int f1(unsigned n){
    int sum=1, power=1;
    for(unsigned i=0;i<=n-1;i++){
        power *= 2;
        sum += power;
    }
    return sum;
}
```

访问局部变量：[ebp-4]、[ebp-8]...
访问参数：[ebp+8]、[ebp+12]...

for/while循环结构：
①循环前的初始化
②cmp、jxxx (否命题)判断是否直接跳过循环？
③循环主体
④ cmp、jxxx (同循环条件)判断是否继续循环？

2017年真题

44.（10 分）在按字节编址的计算机 M 上，题 43 中 f1 的部分源程序（阴影部分）与对应的机器级代码（包括指令的虚拟地址）如下：

其中，机器级代码行包括行号、虚拟地址、机器指令和汇编指令。请回答下列问题。

```
     int f1( unsigned n)
1    00401020   55          push ebp
     ......     ......      ......
         for( unsigned i=0; i<= n -1; i++)
     ......     ......      ......
20   0040105E   39 4D F4    cmp dword ptr [ebp-0Ch],ecx
     ......     ......      ......
             power * = 2;
     ......     ......      ......
23   00401066   D1 E2       shl  edx,1
     ......     ......      ......
         return sum;
     ......     ......      ......
35   0040107F   C3          ret
```

王道考研/CSKAOYAN.COM

48