Передаваемые извне   
переменные:

**push** offset XBuf

**xor** **eax**,**eax**

**mov** **al**,**byte** ptr **[**K**]**

**push** offset rand

**push** offset X

**push** offset Sel

**call** Skreshiv@16

*;СКРЕЩИВАНИЕ*

 Skreshiv proc Sel:**dword**, X:**dword**, rand:**dword**, XBuf:**dword**

local K:**byte**

local addrXBuf:**dword**

local lrand:**dword**

local numA:**DWORD**

Массив Xi ( ссылка на 1-ый эл в **ESI** )

8F E2 46 FD F1 07 A3 90 9D DC E4 61 84 57 B9 95 D4 9B BE 38 D7 04 2D C3 A1 23 6F EB 4D A9 23 51 51 C3 7F 03 29 B2 8F CF 96 B6 0B 11 D0 54 3A 74 92 1B

local numM:**DWORD**

Массив Sel состоит из 20 результатов селекции ( ссылка на 1-ый эл в **EDI** )

07 00 06 07|02 09 03 04|00 00 04 04|04 06 09 02

06 06 04 00|

local exchvar1:**dword**

local exchvar2:**dword**

local loop5:**dword**

**pusha**

Адресация для доступа к одному элементу X

Одна из пяти хромосом первого родителя

**xor** **ebx**,**ebx**

**mov** **bl**, **byte** ptr **[edi**+**ecx**\*2**]**

**mov** **eax**,5

**mul** **bl**

**add** **eax**,loop5

**mov** **bl**, **byte** ptr **[esi**+**eax]**

**mov** **byte** ptr **[**exchvar1**]**, **bl**

**mov** **esi**,**dword** ptr **[**XBuf**]**

Адресация для доступа ко второму элементу X  
Одна из пяти хромосом второго родителя

**mov** **bl**, **byte** ptr **[edi**+**ecx**\*2+1**]**

**mov** **eax**,5

**mul** **bl**

**add** **eax**,loop5

**mov** **bl**, **byte** ptr **[esi**+**eax]**

**mov** **byte** ptr **[**exchvar2**]**, **bl**

**mov** addrXBuf, **esi**

**mov** numA, 48271 *;numA=48271*

**mov** numM, 2147483647 *;numM=2147483647*

**mov** **esi**, **dword** ptr **[**X**]** *;esi=массив X*

**mov** **edi**, **dword** ptr **[**Sel**]** *;edi=массив Sel*

**mov** **edx**,**dword** ptr **[**rand**]** *;edx=rand*

**mov** **edx**,**dword** ptr **[edx]**

**mov** lrand,**edx** *;lrand=edx=rand*

**mov** K,**al** *;K=al= значение должно быть меньше N ( К - сколько генерировать потомков )*

Цикл на K (0..K-1) через ECX

**mov** **ecx**,0 *;ecx=0*

a1:

**mov** loop5,0

Цикл на 5 (0..4) по количеству X в уравнении  
Через переменную loop5

lp5:

**xor** **ebx**,**ebx** *; загрузка байтов для скрещивания в exchvar1 и exchvar2*

В этом цикле могут формироваться различные маски только след. видов:  
{ [0111 1111] , [0011 1111] , [0001 1111] , [0000 1111] , [0000 0111] , [0000 0011] , [0000 0001] }

В рассматриваемом примере сформируется маска [0000 0001]  
Это значит, будем резать так [0000000|0]

**mov** **bl**, **byte** ptr **[edi**+**ecx**\*2**]** *;bl=sel[i+ecx\*2]*

**mov** **eax**,5 *;eax=5*

**mul** **bl** *;eax=sel[i+ecx\*2]\*5*

**add** **eax**,loop5

exchvar1 = 03h [ 0000 0011 ]

exchvar2 = 8Fh [ 1000 1111 ]

**mov** **bl**, **byte** ptr **[esi**+**eax]** *;bl=массив X*

**mov** **byte** ptr **[**exchvar1**]**, **bl** *;exchvar1=bl=массив X*

**mov** **bl**, **byte** ptr **[edi**+**ecx**\*2+1**]** *;bl=sel*

**mov** **eax**,5 *;eax=5*

**mul** **bl** *;al=bl(sel)\*5*

**add** **eax**,loop5

**mov** **bl**, **byte** ptr **[esi**+**eax]** *;bl=массив X*

**mov** **byte** ptr **[**exchvar2**]**, **bl** *;exchvar2=bl=массив X*

zeroTryAgain:

**mov** **bh**,**byte** ptr **[**exchvar2**]** *; на случай повтора, если случится Xi в результате скрещивания будет равен 0*

*; далее работаем только bh вместо exchvar2*

**mov** **eax**, lrand *;eax=lrand=edx=rand*

Получаем случайное число, например

4F ED A1 E9h

*; два повтора для лучшего начального смешивания.*

**mul** numA ; *EAX=a \* X(i-1) (умножаем EAX на dword, указанный по адресу в numA=48271)*

**div** numM *; a \* X(i-1) mod m (полученное произведение в EAX делим на dword, указанного по адресу в numM=2147483647)*

**mov** **eax**,**edx** *;EAX=rand[i-1]*

**mul** numA *;EAX=a \* X(i-1) (умножаем EAX на dword, указанный по адресу в numA=48271)*

**div** numM *;edx=*

**mov** lrand, **edx** *;lrand=edx=rand[i-1]*

**and** **dl**,15 *;выбор 4 последних бит из dl=rand[i], выбор позиции рассечения хромосомы при скрещивании*

**mov** **al**,0ffh *;на случай, если сучайное число ровно восемь, заранее готовим маску перед прыжком (будет полное замещение хромосомы следующего родителя)*

**cmp** **dl**,8

Берем последние 4-бита  
E9h [ 1110 1001 ] **and** **dl**,15 *;в DL = 09h [0000 1001]*  
*Если бы в DL получился 0, значит маска в AL=0 и операция AND с этой маской ничего кроме 0 в exchvar1 не оставит. Если в dl было бы 8, то по маске из AL=0FFh взяли бы все биты из exchvar1 и заместили ими exchvar2. (Все это прыжки на метку nParent (следующий родитель - next Parent))*

Но у нас 09h и проходит процедура формирования маски из последних трех бит, а это в нашем случае [001] Значит маска в AL= [0000 0001] И это уже используется далее в скрещивании.

**je** nParent *;если 8, то это ситуация когда все биты берем от этого родителя*

**and** **dl**,7 *;если выше не 8, то точно (число 0..7) тогда берем первые три бита*

**cmp** **dl**,0

**jz** nParent *; без рассечения будут браться все биты из соотв. хромосомы след. родителя*

**mov** **al**,0 *; иначе выбираем только часть хромосом от обоих*

**mov** **bl**,0

Цикл (1..7) через BL

rOne:

**shl** **al**,1 *; al=маска (1 заполнила все от точки рассечения до конца вправо на число бит из cl=три последние бита из dl=rand[i])*

**add** **al**,1

**inc** **bl**

**cmp** **bl**,**dl**

**jl** rOne

 nParent:

Само cкрещивание:

AL [ 0000 0001 ] <-- AL [0000 0001] AND 8Fh [ 1000 1111 ] (взяли только последний бит от 8Fh)

BH [ 0000 0010 ] <-- BH [0000 0011] AND BL [ 1111 1110 ] (взяли все биты от 03h кроме последнего)

Соединение кусков хромосом через OR

BH [ 0000 0011 ] <-- BH [ 0000 0010 ] OR AL [ 0000 0001 ]

**mov** **bl**,**al** *;bl=копия маски из al*

**and** **al**,**byte** ptr **[**exchvar1**]** *;в al сохраняется результат and al и X[i] для отбора только тех бит, что 1 в al*

**not** **bl** *;bl=инвертир маску для отбора недостающей части от хромосомы другого родителя*

**and** **bh**,**bl** *;наложили маску на X[i] другого родителя*

**or** **bh**,**al** *;схлопнуни разрезанные чаcти в новую хромосому Xi*

**jz** zeroTryAgain *;если хромосома оказалась равно нулю пробуем по другому еще раз через выбор другого места для разрезания*

**mov** **eax**,**ecx**

**mov** **bl**,5

**mul** **bl**

**add** **eax**,loop5

**add** **eax**,addrXBuf

**mov** **byte** ptr **[eax]**,**bh** *;Сохраняем получившегося от скрещивания потомка во внешний массив XBuf похожий на массив X***.** *Потом,**во внешней программе, этим массивом заменим через копирование (rep movsb) часть массива X.*

**inc** loop5

**cmp** loop5,5

**jl** lp5

Цикл на loop5 < 5

**inc** **cl**

**cmp** **cl**,K

Цикл на CL < K

**jl** a1

**mov** **edx**,**dword** ptr**[**rand**]**

**mov** **eax**,lrand *;возвращаем текущее состояние случайного датчика*

**mov** **dword** ptr**[edx]**,**eax**

**popa**

**ret** 16

Skreshiv endp