



Отладчик виртуальных машин PathFinder

Исходная цель создания отладчика



- Отладчик PathFinder создан в связи с необходимостью контроля за исполнением гипервизором инструкций виртуальной машины;
- Отладчик работает в связке с операционной системой QP ОС и гипервизором QP VMM;
- PathFinder является дополнительным инструментом и не входит в стандартную сборку гипервизора QP VMM.

Внешний вид PathFinder



The screenshot displays the PathFinder application in a Windows XP environment. The desktop features a landscape background and several icons. Open windows include:

- File Explorer (MSDOS):** Shows a directory listing with columns for Name, C:\N, and Name. Files listed include MSDOS7, DN, !TEST_2, TEST_DOS, ZUBKOV, osldr.exe, WIN98, UC, SETKA10, RINGNES, QUIEW, QBASIC, KULAKOV, GAMES, FPU, DOS, and DE.
- PathFinder:** The main application window showing assembly code in the Disassembly (16) pane. The registers pane at the bottom shows values for EAX, EBX, ECX, EDI, etc.
- System Monitor:** Located in the top right, showing system performance metrics: Загрузка системы (25%), Всего (25%), Ядро (25%), and Памяти занято (24%).
- Virtual Machine Console:** A window showing the output of a virtual machine, including the text "MSDOS" and "ГЛАВ".

The PathFinder window displays the following assembly code:

```
Disassembly (16)
0000BFBF pop     ax
0000BF00 popf
0000BF01 retm
0000BF02 push   ds
0000BF03 push   si
0000BF04 push   dx
0000BF05 lds    si, dword ptr cs:[9572h]
0000BF06 mov   word ptr [si+2], 0
0000BF07 push  a1, 1
0000BF08 call  0FB59h
0000BF09 jne   0C019h
0000BF0A cmp   byte ptr ss:[0AA0h], 0
0000BF0B jne   0BFEfh
0000BF0C mov   ah, 5
0000BF0D call  0F145h
0000BF0E mov   ax, 168Fh
0000BF0F mov   dh, 1
0000BF10 int  2Fh
0000BF11 test  ax, ax
0000BF12 je    0C020h
0000BF13 mov   ah, 84h
0000BF14 int  2Ah
0000BF15 cmp   byte ptr ss:[0DBDh], 0FFh
0000BF16 ine
```

The registers pane shows the following values:

Регистры	Точный останов	Стек вызовов
EAX = 00008200 (33280)		CS_BS = 00006530 (25904)
EBX = 00000058 (88)	EBP = 00003131 (12593)	SS = 000000C9 (201)
ECX = 00000000 (0)	ESP = 0000079A (1946)	SS_BS = 00000C90 (3216)
EDX = 00000102 (258)	EIP = 00005A8F (23183)	
ESI = 000024B8 (9400)	EFL = 00000246 (582)	ST0 = 000000000000000000000000
EDI = 000003127 (12583)	CS = 000000653 (1619)	ST1 = 000000000000000000000000
		ST2 = 000000000000000000000000
		ST3 = 0000FFFFFF0000000000000000
		ST4 = 0002FFFFFF3001403C08
		ST5 = 0000FFFFFF00A020D786
		ST6 = 44C000000000000000000000
		ST7 = 0286FFFFFF00A020E056

The status bar at the bottom of the PathFinder window indicates: Состояние : Останов по трассировке, Процессор: 0, Количество процессоров: 1.

```
Virtual GS FS Physical
00000000 8A 04 22 16 65 04 70 00
00000008 16 00 4A 06 65 04 70 00  J De p
00000010 65 04 70 00 57 00 00 E0 e p W p
00000018 6A 00 00 E0 1B 00 00 E0 j p p
00000020 1F 00 4A 06 28 00 4A 06  J ( J
00000028 1B 00 00 E0 1B 00 00 E0  p p
00000030 1B 00 00 E0 82 00 4A 06  pB J
00000038 1B 00 00 E0 1C 00 00 E0  p p
00000040 6D 00 00 C0 16 01 00 E0 m p p
00000048 29 01 00 E0 EE 03 70 00 ) p r p
00000050 51 01 00 E0 F6 03 70 00 Q p Y p
00000058 2D 04 70 00 8C 01 00 E0 - p M p
00000060 9F 01 00 E0 54 04 70 00 Я p T p
00000068 AD 01 00 E0 8A 04 22 16 н p K p
00000070 DB 01 00 E0 1B 00 00 E0  p p
00000078 22 05 00 00 1B 00 00 E0 " p p
00000080 A8 0F C9 00 8B 04 22 16 и p Л p
00000088 14 03 A8 13 8A 04 22 16  p и K p
00000090 2B 8F BB 18 BC 0F C9 00 + p p p p
00000098 C6 0F C9 00 DB 04 22 16  p p p p
000000A0 6C 10 C9 00 66 04 70 00 I p f p
000000A8 6C 10 C9 00 6C 10 C9 00 I p I p
000000B0 6C 10 C9 00 6C 10 C9 00 I p I p
000000B8 62 01 A8 13 CC 01 A9 13 б p и p p p
000000C0 EA E4 0F C9 00 00 00 E0 ъ p p p
000000C8 6C 10 C9 00 6C 10 C9 00 I p I p
000000D0 6C 10 C9 00 6C 10 C9 00 I p I p
000000D8 6C 10 C9 00 6C 10 C9 00 I p I p
000000E0 6C 10 C9 00 6C 10 C9 00 I p I p
000000E8 6C 10 C9 00 6C 10 C9 00 I p I p
000000F0 6C 10 C9 00 6C 10 C9 00 I p I p
000000F8 6C 10 C9 00 6C 10 C9 00 I p I p
00000100 1B 00 00 E0 3D 00 0B 02  p p= p
00000108 1B 00 00 E0 00 16 4C C4  p p L p
00000110 1B 00 00 E0 1B 00 00 E0  p p p
00000118 4D 00 0B 02 1B 00 00 E0 M p p p
00000120 1B 00 00 E0 1B 00 00 E0  p p p
00000128 1B 00 00 E0 1B 00 00 E0  p p p
00000130 1B 00 00 E0 1B 00 00 E0  p p p
00000138 1B 00 00 E0 FC 04 70 00  p pNE p
00000140 1B 00 00 E0 1B 00 00 E0  p p p
00000148 1B 00 00 E0 1B 00 00 E0  p p p
00000150 1B 00 00 E0 1B 00 00 E0  p p p
00000158 1B 00 00 E0 1B 00 00 E0  p p p
00000160 1B 00 00 E0 1B 00 00 E0  p p p
00000168 1B 00 00 E0 1B 00 00 E0  p p p
00000170 1B 00 00 E0 1B 00 00 E0  p p p
00000178 1B 00 00 E0 1B 00 00 E0  p p p
00000180 1B 00 00 E0 1B 00 00 E0  p p p
00000188 1B 00 00 E0 1B 00 00 E0  p p p
00000190 1B 00 00 E0 1B 00 00 E0  p p p
00000198 1B 00 00 E0 1B 00 00 E0  p p p
000001A0 1B 00 00 E0 1B 00 00 E0  p p p
000001A8 1B 00 00 E0 1B 00 00 E0  p p p
```



- Просмотр виртуальной и физической памяти



Disassembly (16) X

```
0000BFBF pop ax
0000BFC0 popf
0000BFC1 retn
0000BFC2 push ds
0000BFC3 push si
0000BFC4 push dx
0000BFC5 lds si, dword ptr cs:[9572h]
0000BFCA mov word ptr [si+2], 0
0000BFCF mov al, 1
0000BFD1 push ax
0000BFD2 call 0FB59h
0000BFD5 jne 0C019h
0000BFD7 cmp byte ptr ss:[0AA0h], 0
0000BFDD jne 0BFEFh
0000BFDF mov ah, 5
0000BFE1 call 0F145h
0000BFE4 mov ax, 168Fh
0000BFE7 mov dh, 1
0000BFE9 int 2Fh
0000BFEB test ax, ax
0000BFED je 0C020h
0000BFEF mov ah, 84h
0000BFF1 int 2Ah
0000BFF3 cmp byte ptr ss:[0DBDh], 0FFh
0000BFF9 jne 0C011h
0000BFFB push bx
0000BFFC push cx
0000BFFD push ds
0000BFFE push ss
0000BFFF pop ds
0000C000 sub ...
```

Регистры | Точки останова | Стек вызовов

Адрес	Условие	Информация
<input checked="" type="checkbox"/> 0x0000BFC4	не установлено	

- Пошаговое исполнение виртуальной машины
- Задание точек останова



h

00000050 51 01 00 E0 F6 03 7

0
7
2
0
0
2
2
C
2
7
C
C
A
0
C

Установка параметров точки останова

Включить точку останова

Отменить условие Использовать данное условие

Адрес: 00000000 Тип памяти: Virtual Условие: == Значение (hex): FFFFFFFF Размер: DWord

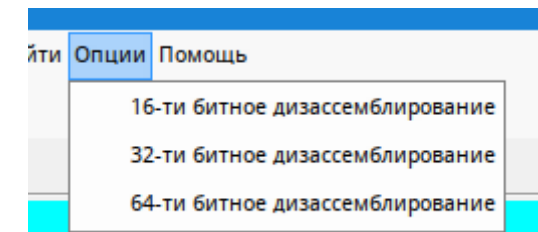
Установить Закрыть

ss: [0DBDh], 0FFh

000000D0	6C	10	C9	00	6C	10	C
000000D8	6C	10	C9	00	6C	10	C
000000E0	6C	10	C9	00	6C	10	C
000000E8	6C	10	C9	00	6C	10	C
000000F0	6C	10	C9	00	6C	10	C
000000F8	6C	10	C9	00	6C	10	C
00000100	1B	00	00	E0	3D	00	0
00000108	1B	00	00	E0	00	16	4

- Задание точек останова с условием

```
Disassembly (64) - X
000000000000BFBF pop rax
000000000000BFC0 popfq
000000000000BFC1 retn
000000000000BFC2 push ds
000000000000BFC3 push rsi
000000000000BFC4 push rdx
000000000000BFC5 lds esi, fword ptr [rsi]
000000000000BFC8 jb 000000000000BF5Fh
000000000000BFCA mov dword ptr [rdx+rax], 5001B000h
000000000000BFD2 call 000000004275FB5Bh
000000000000BFD7 cmp byte ptr [rsi], 0A0h
000000000000BFDB or al, byte ptr [rax]
000000000000BFDD jne 000000000000BFEFh
000000000000BFDf mov ah, 5
000000000000BFE1 call 0FFFFFFF8FB8F147h
000000000000BFE6 push ss
000000000000BFE7 mov dh, 1
000000000000BFE9 int 2Fh
000000000000BFEB test eax, eax
000000000000BFED je 000000000000C020h
000000000000BFEF mov ah, 84h
000000000000BFF1 int 2Ah
000000000000BFF3 cmp byte ptr [rsi], 0BDh
000000000000BFF7 or eax, 531675FFh
000000000000BFFC push rcx
000000000000BFFD push ds
000000000000BFFE push ss
000000000000BFFF pop ds
000000000000C000 sub eax, eax
000000000000C002 call 0FFFFFFFEBE8C307h
000000000000C007 ...
```



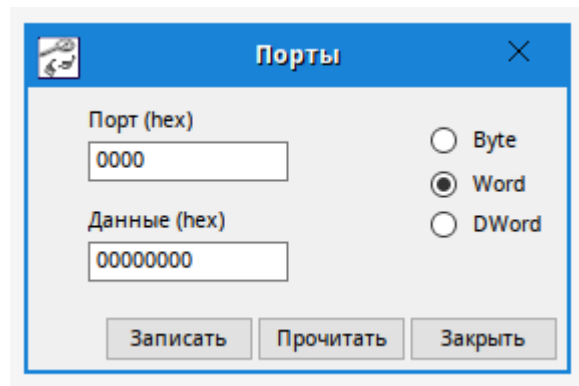
- Отладчик автоматически определяет режим дизассемблирования на основе анализа системных регистров;
- Режим дизассемблирования можно задать вручную



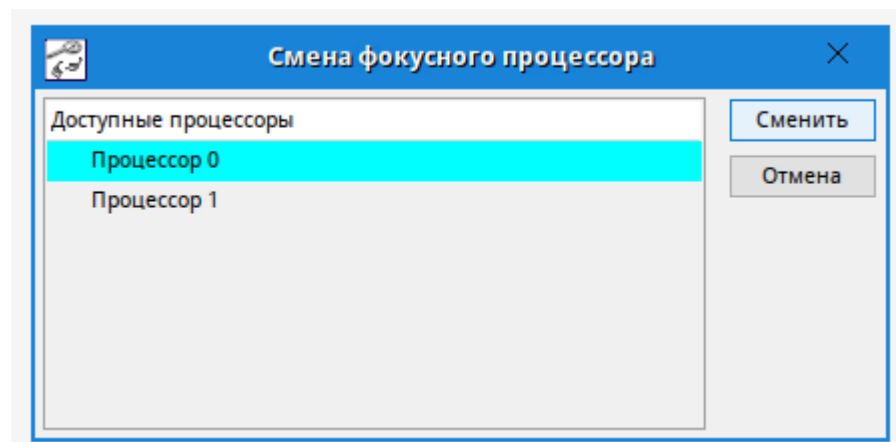
Регистры	Точки останова	Стек вызовов	
EAX = 00008200 (33280)	ES = 000028AD (10413)	DR7 = 00000400 (1024)	MM2 = 0000000000000000
EBX = 00000058 (88)	ES_BS = 00028AD0 (166608)		MM3 = FFFFFFF000000000
ECX = 00000000 (0)	FS = 00000000 (0)	ST0 = 00000000000000000000	MM4 = FFFFFFF03001403C08
EDX = 00000102 (258)	FS_BS = 00000000 (0)	ST1 = 00000000000000000000	MM5 = FFFFFFF000A020D786
ESI = 000024B8 (9400)	GS = 00000000 (0)	ST2 = 00000000000000000000	MM6 = 0000000000000000
EDI = 00003127 (12583)	GS_BS = 00000000 (0)	ST3 = 0000FFFFFF0000000000	MM7 = FFFFFFF000A020E056
		ST4 = 0002FFFFFF03001403C08	
EBP = 00003131 (12593)	CR0 = 00000000 (0)	ST5 = 0000FFFFFF000A020D786	XMM0 = 0000000000000000 0000000000000000
ESP = 0000079A (1946)	CR2 = 00000000 (0)	ST6 = 44C00000000000000000	XMM1 = 0000000000000000 0000000000000000
EIP = 00005A8F (23183)	CR3 = 00030000 (196608)	ST7 = 0286FFFFFF000A020E056	XMM2 = 0000000000000000 0000000000000000
EFL = 00000246 (582)	CR4 = 00000000 (0)	FPS = FFFF (65535)	XMM3 = 0000000000000000 0000000000000000
CS = 00000653 (1619)	TR = 00000028 (40)	FPC = 0000 (0)	XMM4 = 0000000000000000 0000000000000000
CS_BS = 00006530 (25904)		FPT = 0000 (0)	XMM5 = 0000000000000000 0000000000000000
SS = 000000C9 (201)	DR0 = 00000000 (0)	FIP = 00000000 (0)	XMM6 = 0000000000000000 0000000000000000
SS_BS = 00000C90 (3216)	DR1 = 00000000 (0)	FDP = 00000000 (0)	XMM7 = 0000000000000000 0000000000000000
	DR2 = 00000000 (0)		MXCSR = 00000000 (0)
DS = 000000C9 (201)	DR3 = 00000000 (0)	MM0 = 0000000000000000	
DS_BS = 00000C90 (3216)	DR6 = FFFF0FF0 (4294905840)	MM1 = 0000000000000000	

Регистры	Точки останова	Стек вызовов
EAX = 00008200 (33280)		
EBX = 00000058 (88)		
ECX = 00000000 (0)		
EDX = 00000102 (258)		
ESI = 000024B8 (9400)		
EDI = 00003127 (12583)		

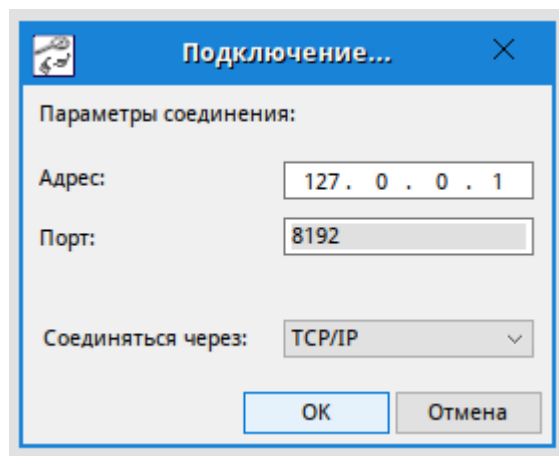
- Отладчик отображает и позволяет управлять большинством регистров процессора;
- Отображение регистров можно настраивать.



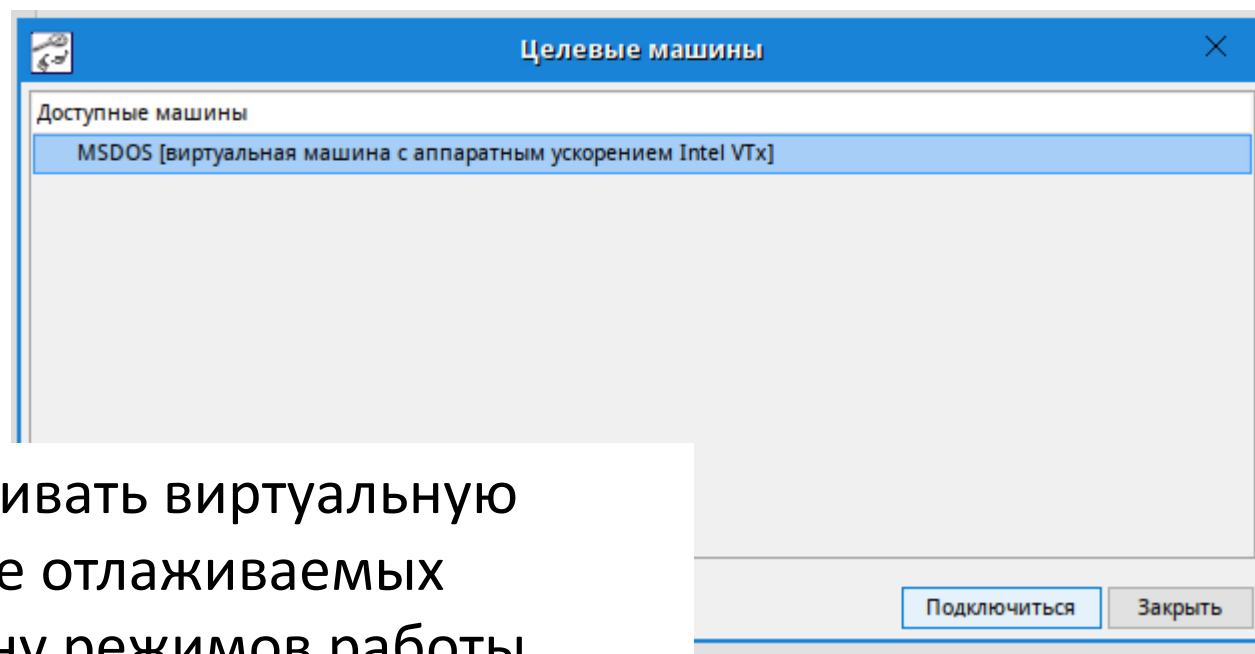
- Можно выполнять операции с портами ввода вывода.



- Можно переключаться между процессорами виртуальной машины.



- PathFinder связывается с гипервизором по протоколу TCP



- PathFinder позволяет отлаживать виртуальную машину на участках кода, не отлаживаемых другими отладчиками: смену режимов работы процессора, настройку виртуальной памяти, переключение TSS и.т.д.



- PathFinder находится в состоянии бета-версии. Он может быть доработан до коммерческого продукта при наличии заинтересованных организаций и предприятий.

***Благодарю за
внимание!***

Вопросы?