

# Программный эмулятор QEMU – платформа для разработки и анализа операционных систем

Довгалюк П. М.

Новгородский государственный университет  
Институт системного программирования РАН

# Отладка ОС с помощью кода внутри системы

---

- ▶ Система должна функционировать
- ▶ Нельзя отладить любой выполняемый код
  - ▶ Загрузочный код
  - ▶ BIOS
- ▶ Нужно настраивать заранее
- ▶ Влияет на работу отлаживаемого кода

# QEMU

---

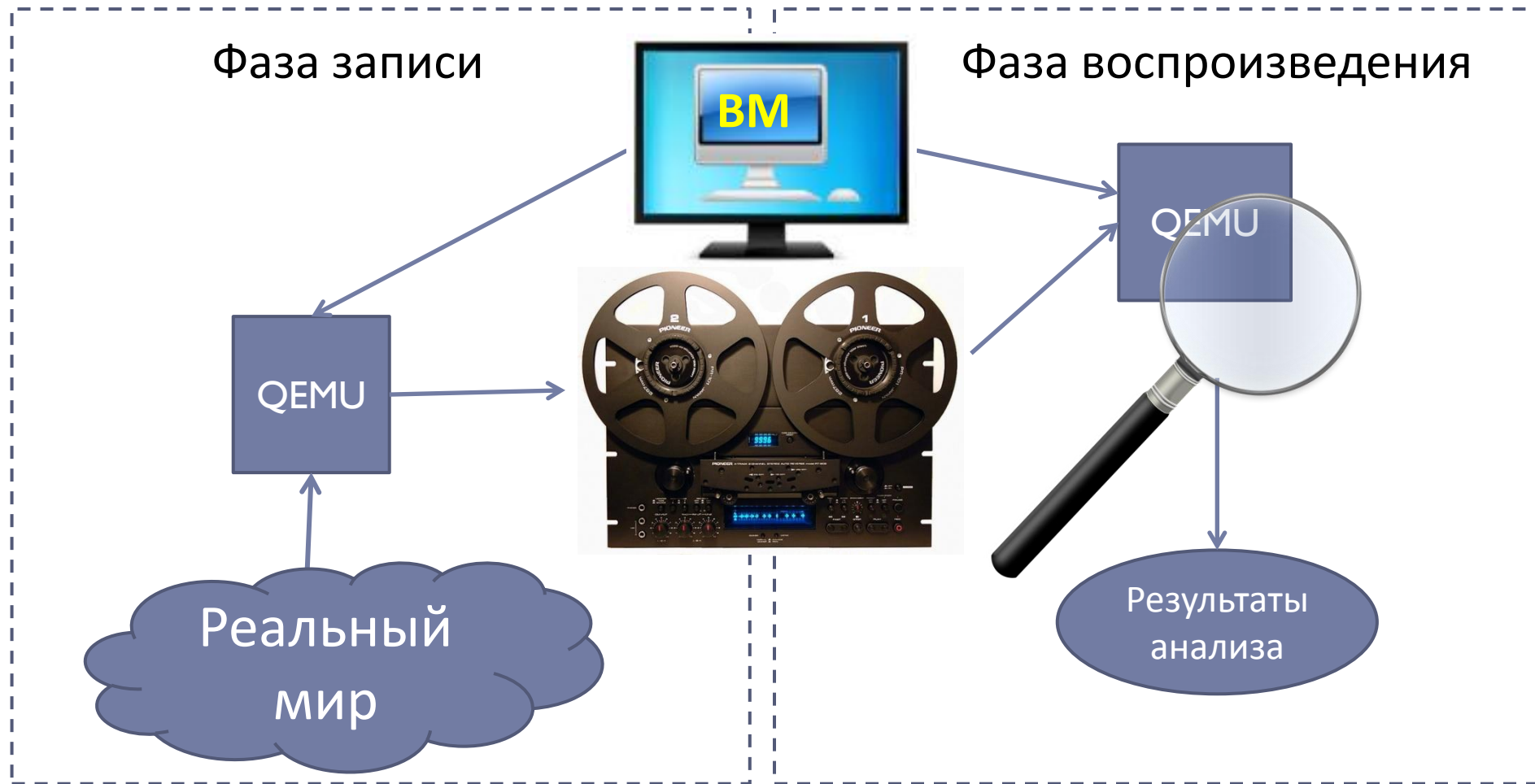
- ▶ **Симулятор с открытым исходным кодом**
  - ▶ Можно моделировать и отлаживать собственные периферийные устройства или аппаратные платформы
- ▶ **Виртуальные машины на платформах i386, ARM, MIPS, PowerPC и т.д.**
  - ▶ 20 семейств процессоров
- ▶ **Отладочные возможности**
  - ▶ Подключение удаленного отладчика gdb
  - ▶ Трассировка транслируемых и выполняемых блоков кода

# Отладка ОС через gdbserver в симуляторе

---

- ▶ Можно подключиться в любой момент
  - ▶ Даже при критическом сбое
  - ▶ Даже до загрузки ОС
- ▶ Отладка без реального оборудования
  - ▶ Отладка прошивки/драйвера
  - ▶ Отладка моделей оборудования
- ▶ **Работает медленнее из-за виртуализации**
- ▶ **Модели оборудования не всегда идеальны**
- ▶ **Ход работы может измениться из-за остановок**

# Идея записи и воспроизведения работы системы

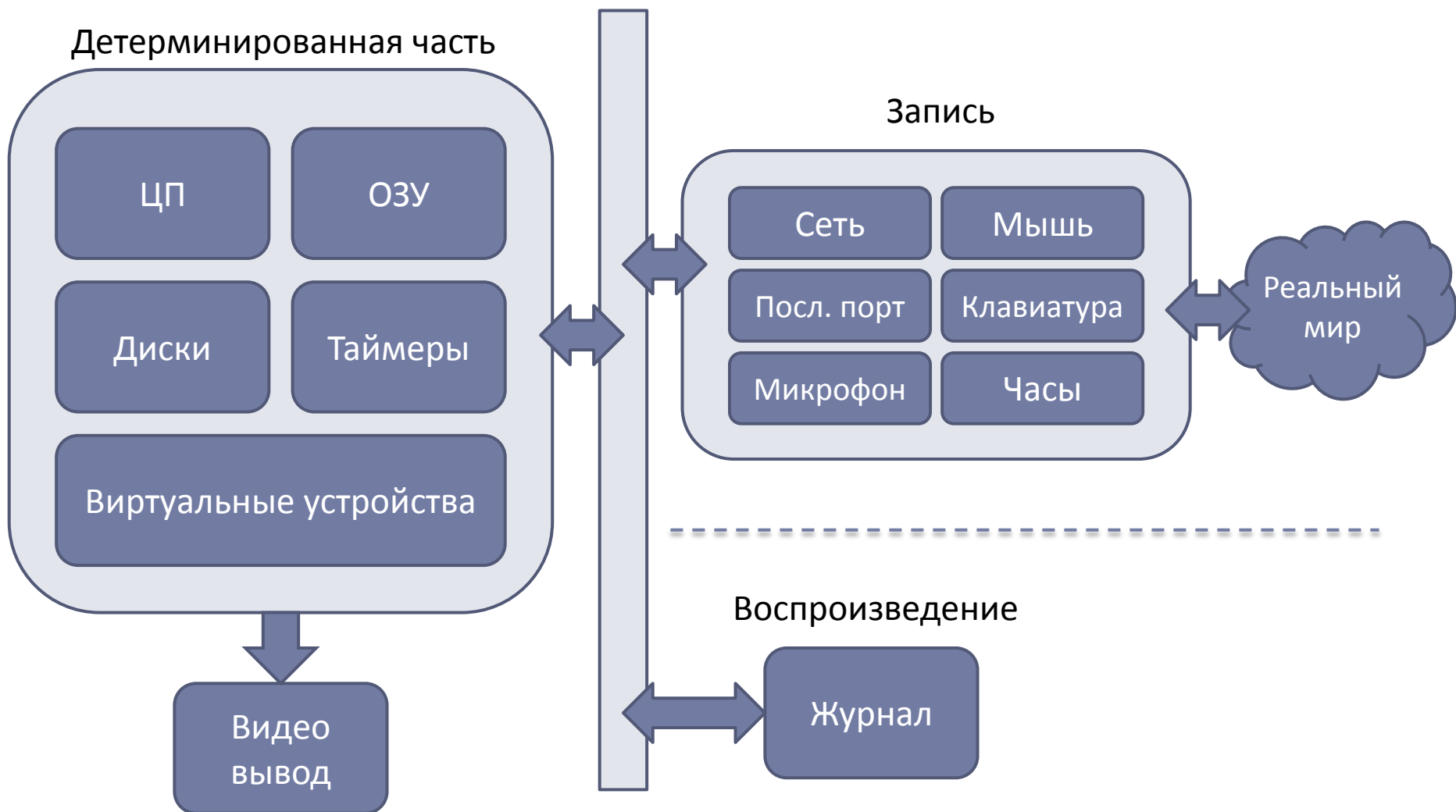


# Запись / воспроизведение работы

---

- ▶ Не нужно перенастраивать окружение
- ▶ Отладка редко проявляющихся ошибок
- ▶ Разделение записи и анализа
  - ▶ Можно анализировать сетевые приложения, критичные к временным параметрам
- ▶ Обратная отладка через `gdb`
  - ▶ Отладчик не влияет на ход работы гостевого кода
- ▶ Записанный сценарий можно переносить между машинами
  - ▶ Воспроизведение ошибочных сценариев
  - ▶ Распараллеливание анализа

# Воспроизведение в симуляторе



# Запись и воспроизведение

---

- ▶ Должны работать на всех платформах QEMU
  - ▶ Протестированы для x86, x86-64, ARM, MIPS
- ▶ Отладка и анализ всей системы
  - ▶ Ядро и BIOS
  - ▶ Виртуальные устройства
- ▶ Нет воздействия на гостевую систему при анализе
  - ▶ Профилирование
  - ▶ Анализ помеченных данных
  - ▶ Отладка
  - ▶ Трассировка



# Показатели работы записи / воспроизведения

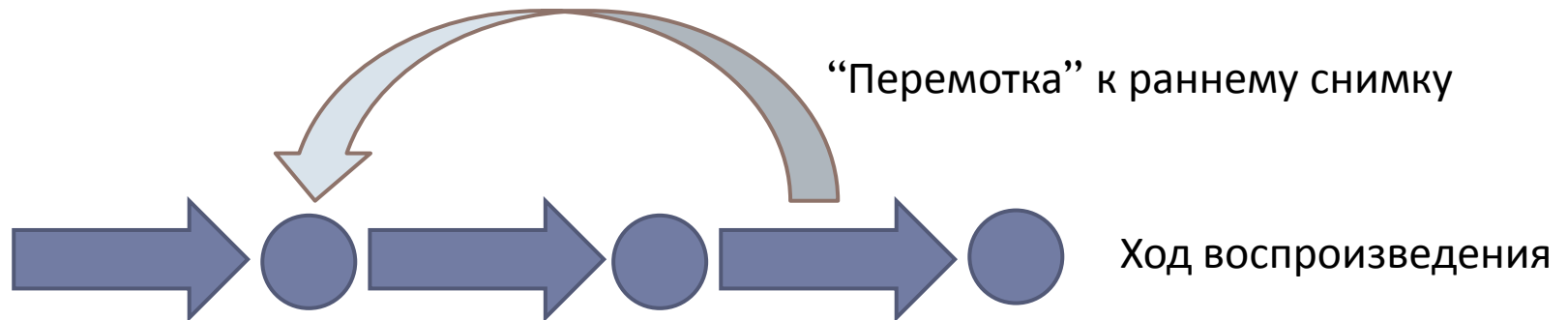
---

Загрузка ОС	Замедление при записи	Замедление при воспроизведении	Размер журнала, байт на 1000 инструкций
i386 (Win)	31%	156%	2.3
i386 (Debian)	41%	136%	21.9
ARM (Debian)	32%	191%	17.9
MIPS (Debian)	14%	139%	75.4

# Обратная отладка

---

- ▶ Нужна для изучения того, что уже было
- ▶ Делаются снимки системы для возврата назад



# Обратная отладка


---

## ▶ Команды GDB

- ▶ reverse-continue
- ▶ reverse-step
- ▶ reverse-stepi
- ▶ reverse-next
- ▶ reverse-nexti
- ▶ reverse-finish

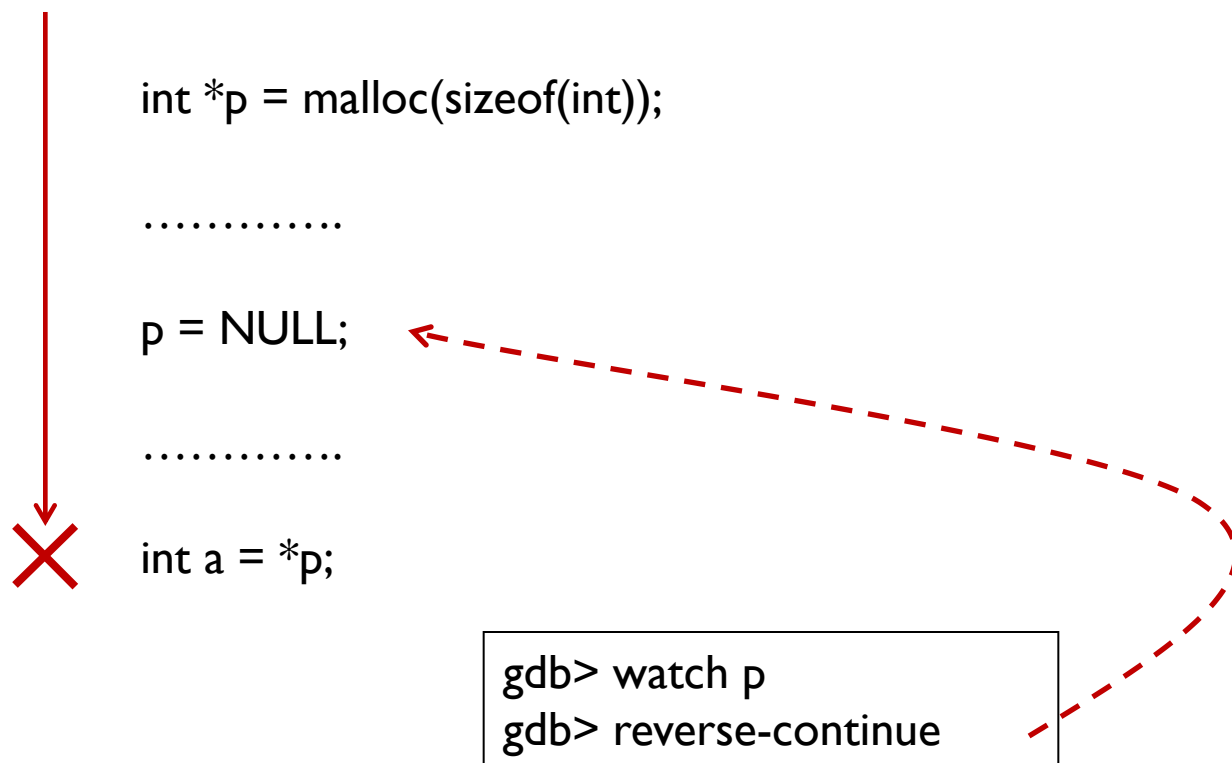
# Обратная отладка

---

```
int *p = malloc(sizeof(int));  
.....  
p = NULL;  
.....  
 int a = *p;
```

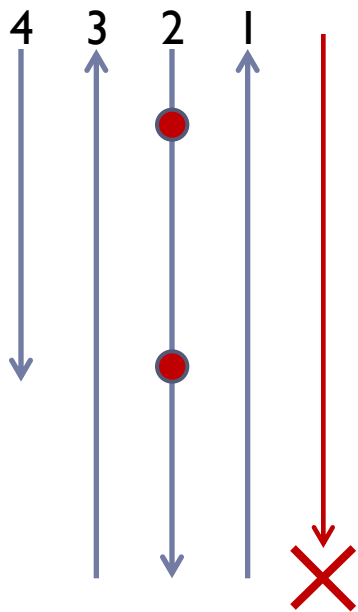
# Обратная отладка

---



# Обратная отладка

---



```
int *p = malloc(sizeof(int));
```

```
.....
```

```
p = NULL;
```

```
.....
```


```
int a = *p;
```

```
gdb> watch p  
gdb> reverse-continue
```

# Зачем это нужно?

---

- ▶ Обратная отладка через gdb
  - ▶ пользовательские приложения
  - ▶ драйверы
  - ▶ ядро ОС
  - ▶ BIOS



```
initrd-tools: 0.1.73
mount: fs type devfs not supported by kernel
umount: devfs: not mounted
mount: fs type devfs not supported by kernel
umount: devfs: not mounted
pivot_root: No such file or directory
/sbin/init: 426: cannot open dev/console: No such file
Kernel panic: Attempted to kill init!
```

# Зачем это нужно?

- ▶ Сбор данных
  - ▶ трассировка инструкций
  - ▶ обращения к памяти
  - ▶ сетевой трафик
- ▶ Динамический анализ
  - ▶ помеченные данные
  - ▶ символическое выполнение

The screenshot shows a Wireshark interface capturing traffic on a Realtek RTL8139/810x Family Fast Ethernet NIC. The main display area shows a list of captured packets with the following columns: No., Time, Source, Destination, Protocol, and Info.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	DHCP Discover
2	0.000704	IntelCor_0d:56:78	Broadcast	ARP	who has 10.0
3	0.070622	0.0.0.0	255.255.255.255	DHCP	DHCP Request
4	0.472383	0.0.0.0	255.255.255.255	DHCP	DHCP Request
5	0.644042	Realteku_12:34:56	Broadcast	ARP	who has 10.0
6	0.644108	Realteku_12:34:56	Broadcast	ARP	who has 10.0
7	1.001716	Realteku_12:34:56	Broadcast	ARP	who has 10.0
8	1.001767	Realteku_0a:e3:c7	Realteku_12:34:56	ARP	10.0.0.158 i
9	1.385747	10.0.0.162	10.0.0.158	TCP	37340 > iscs
10	1.385965	10.0.0.158	10.0.0.162	TCP	iscsi-target
11	1.386246	10.0.0.162	10.0.0.158	TCP	37340 > iscs
12	1.390919	10.0.0.162	10.0.0.158	TCP	[TCP segment
13	1.610420	10.0.0.158	10.0.0.162	TCP	iscsi-target
14	1.610748	10.0.0.162	10.0.0.158	TCP	[TCP segment
15	1.813495	10.0.0.158	10.0.0.162	TCP	iscsi-target

The packet details pane for the selected packet (No. 1) shows the following structure:

- Frame 1: 429 bytes on wire (3432 bits), 429 bytes captured (3432 bits)
- Ethernet II, Src: Realteku\_12:34:56 (52:54:00:12:34:56), Dst: Broadcast (ff
- Internet Protocol, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.25
- User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
- Bootstrap Protocol

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 ff ff ff ff ff ff 52 54 00 12 34 56 08 00 45 00 .....RT..4V..E.
0010 01 9f 00 01 00 00 40 11 79 4e 00 00 00 00 ff ff .....@.yN.....
0020 ff ff 00 44 00 43 01 8b c7 bb 01 01 06 00 00 12 ...D.C.....
0030 34 56 00 04 00 00 00 00 00 00 00 00 00 00 00 4v.....
0040 00 00 00 00 00 00 52 54 00 12 34 56 00 00 00 .....RT..4V.....
```





# WinDbg

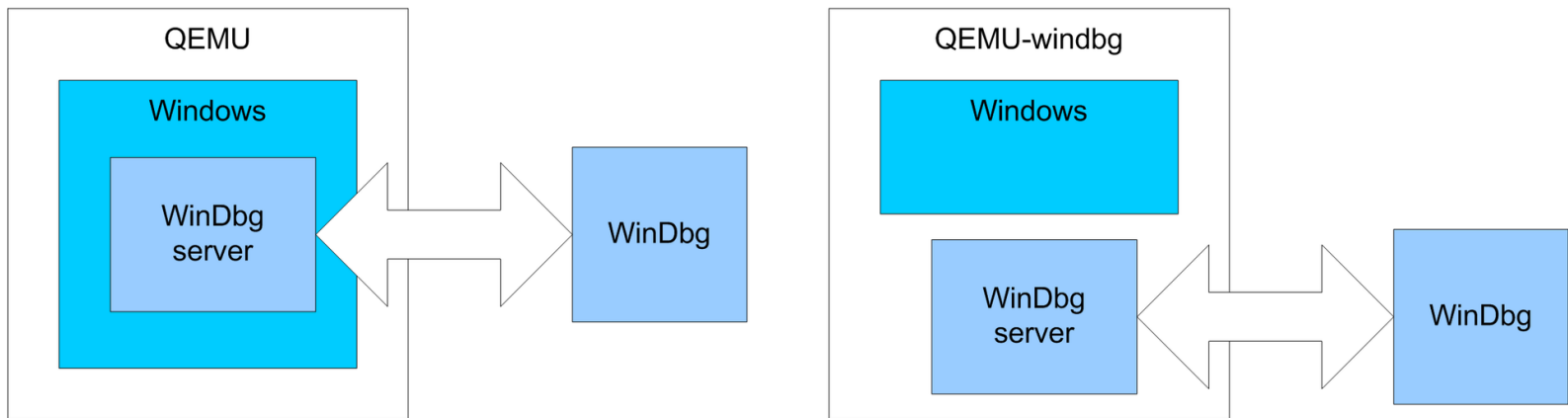
Disassembly - Kernel 'com:pipe,baud=115200,port=\\.\pipe\windbg, resets=...

Offset: @scopeip

```
fc4fd3c0 53          push     ebx
fc4fd3c1 56          push     esi
fc4fd3c2 8b7508     mov     esi,dword ptr [ebp+8]
fc4fd3c5 8b5e28     mov     ebx,dword ptr [esi+28h]
fc4fd3c8 57          push     edi
fc4fd3c9 8b7d0c     mov     edi,dword ptr [ebp+0Ch]
fc4fd3cc 68b8110000 push     11B8h
fc4fd3d1 68f4d34ffc push     offset CLASSPNP!ClassFindModePage+
fc4fd3d6 57          push     edi
fc4fd3d7 56          push     esi
fc4fd3d8 e8b3efffff call    CLASSPNP!ClassAcquireRemoveLockEx
fc4fd3dd 85c0     test     eax,eax
fc4fd3df 57          push     edi
fc4fd3e0 56          push     esi
fc4fd3e1 0f85e12a0000 jne    CLASSPNP!ClassDeviceControlDispatc
fc4fd3e7 8b4360     mov     eax,dword ptr [ebx+60h]
fc4fd3ea ff5018     call    dword ptr [eax+18h]
fc4fd3ed 5f          pop     edi
fc4fd3ee 5e          pop     esi
fc4fd3ef 5b          pop     ebx
fc4fd3f0 5d          pop     ebp
fc4fd3f1 c20800     ret     8
fc4fd3f4 643a5c7870 cmp     bl,byte ptr fs:[eax+edi*2+70h]
fc4fd3f9 7370     jae    CLASSPNP!FreeDictionaryEntry+0x46
fc4fd3fb 5c          pop     esp
fc4fd3fc 647269     jb     CLASSPNP!FreeDictionaryEntry+0x43
fc4fd3ff 7665     jbe    CLASSPNP!FreeDictionaryEntry+0x41
fc4fd401 7273     jb     CLASSPNP!FreeDictionaryEntry+0x51
fc4fd403 5c          pop     esp
```

# WinDbg

---



# WinDbg

---

- ▶ Отладка не обнаруживается программами
- ▶ Работают все обычные команды
- ▶ Доступны высокоуровневые возможности WinDbg
  
- ▶ Не работает перехват событий
  - ▶ int 3
  - ▶ создание процесса
  - ▶ исключительная ситуация
  - ▶ ...

# Интроспекция

---

- ▶ Отладка программ внутри симулятора
  - ▶ Извлечение данных о процессах и потоках
- ▶ Трассировка
  - ▶ Инструкции
  - ▶ Сетевые операции
  - ▶ Дисковые операции
- ▶ Мониторинг конкретного процесса
  - ▶ Системные вызовы
  - ▶ Вызовы API-функций

# Интроспекция

---

- ▶ Трассировка машинных инструкций
- ▶ Трассировка обращений к диску
- ▶ Трассировка отображений виртуальной памяти в физическую
- ▶ Отслеживание системных вызовов
- ▶ Отслеживание вызовов API-функций
- ▶ Сбор информации о процессах
- ▶ Извлечение записываемых файлов

# QEMU

---

- ▶ Отладка в gdb и WinDbg
- ▶ Запись/воспроизведение работы системы
- ▶ Динамическое инструментирование
- ▶ Интроспекция