ИСП РАН

Применение программных эмуляторов для полносистемного анализа бинарного кода мобильных платформ

Батузов К.А., Ефимов В.Ю., Падарян В.А., Тихонов А.Ю.
{batuzovk, real, vartan, fireboo}@ispras.ru

x86

ARM

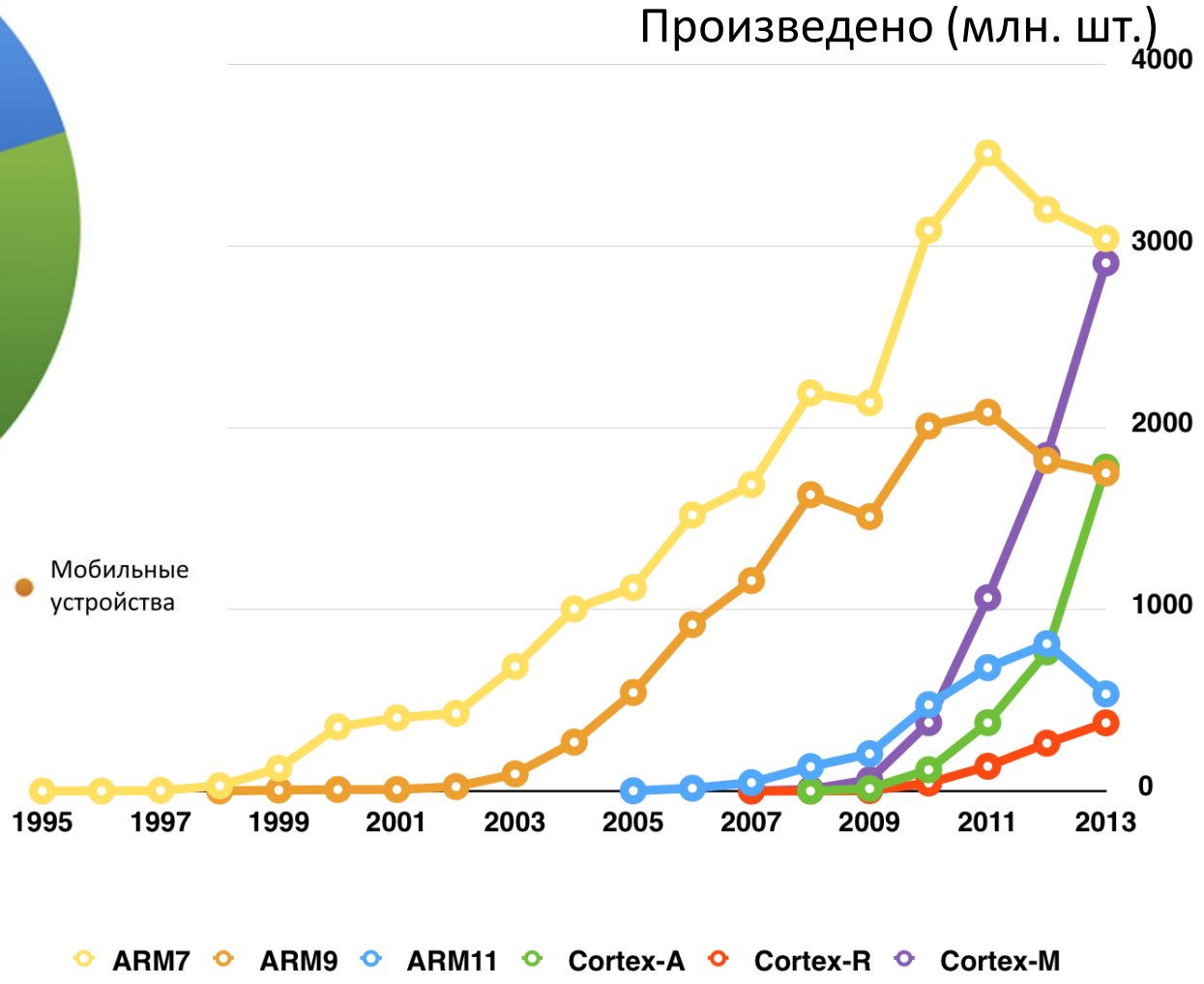
ARC

MIPS

Power

…

ARM: распределение по отраслям



Отладка встраиваемого ПО обеспечивается аппаратными средствами

- JTAG port
- Trace port

Предметы, цели и задачи анализа бинарного кода

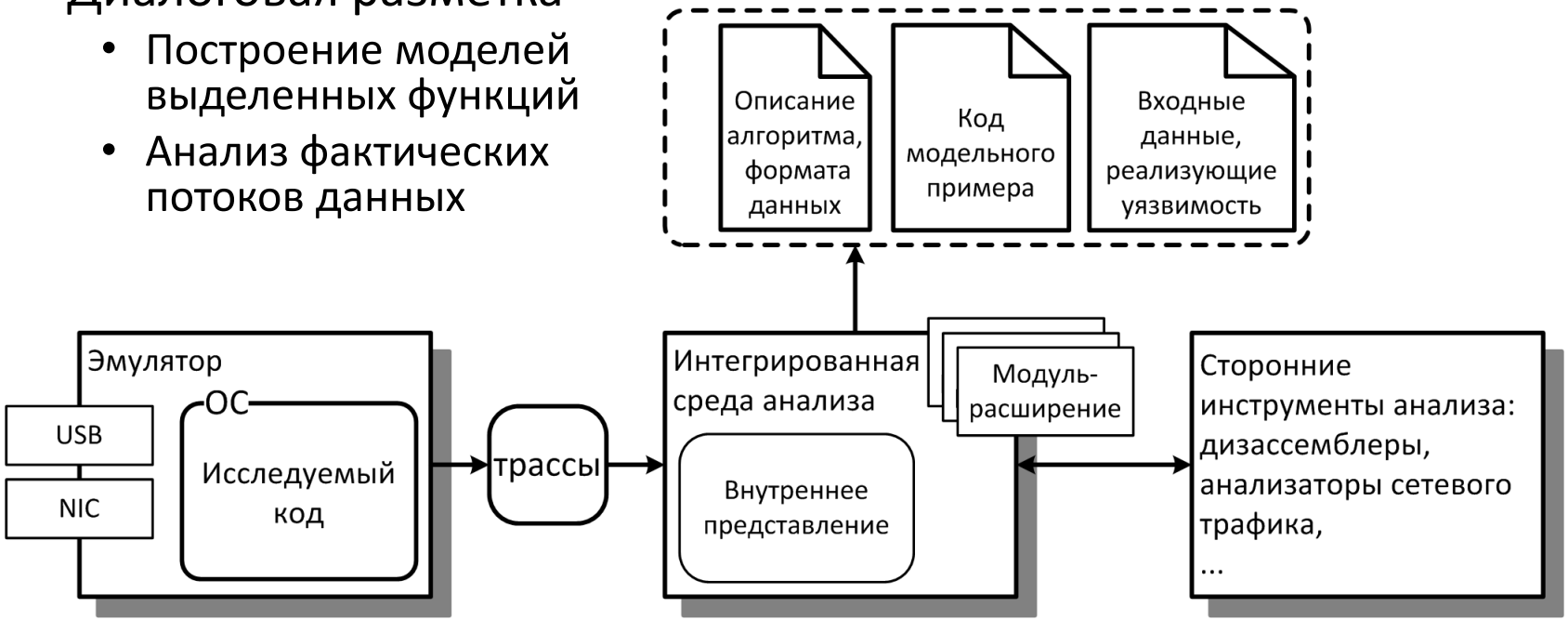
ЦПУ	Периферия
Универсальные компьютеры	
x86	Стандартная периферия, UBS-токены
Коммуникационное оборудование	
MIPS ARM PPC x86	Нестандартные сетевые контроллеры
Мобильные платформы	
ARM	«Нестандартные» модули связи (Cell, Wi-Fi, ...)

- Восстановление алгоритмов, форматов данных и протоколов
- Поиск закладок
- Поиск уязвимостей

- Статический анализ
- Динамический анализ
  - Слайсинг
  - Анализ помеченных данных
  - Фаззинг
  - Символьная интерпретация

Комбинированный анализ бинарного кода

- Программные эмуляторы
  - отладка, сбор трасс
- Интегрированная среда анализа
  - Последовательное повышение уровня представления
  - Построение статико-динамического представления
  - Диалоговая разметка
    - Построение моделей выделенных функций
    - Анализ фактических потоков данных



\trace\trace

Трасса Анализ Правка Поиск Вид Навигация Trace

4D00 --------

Регистры

GPR SR CP15 EXT ???

pc = ffff0008	r0 = 002ee000	r1 = 4004c474
r4 = 002ed000	r5 = 4004c320	r6 = 002ee000
r9 = 4004c474	r10 = 485a07b4	r11 = 0000001a
r14 = 4001d3af	r13_svc = d493bff8	r14_svc = 40010274
r13_und = c02f68f8	r14_und = c0023bc0	r13_irq = c02f68e0
r9_fiq = 00000000	r10_fiq = 00000000	r11_fiq = 00000000
r14_fiq = 00000000	cpsr = 00000093	spsr_svc = 00000010
spsr_irq = 80000193	spsr_fiq = 00000000	cpacr = 00f00000
fcseidr = 00000000	contextidr = 0000013b	tpidrprw = 00000000
pid = 00000000000000e0	zid = 00000000000000e0	

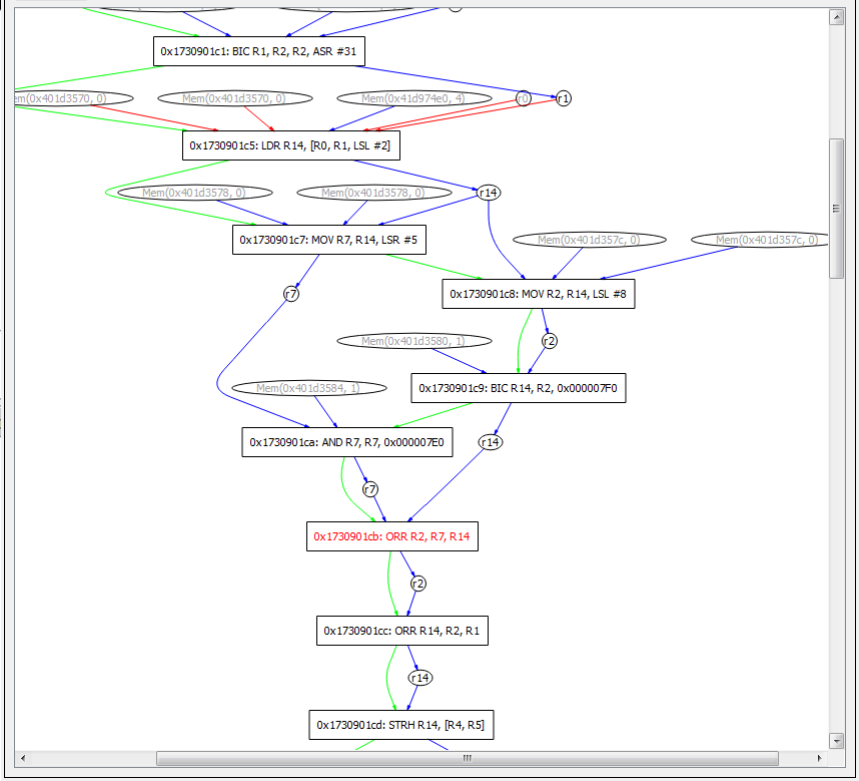
	Модул	ИСК	Опкод	Инструкция,	Ссылки,	Ком
4CF1		4001D38C	2468	LDR	R4,	
4CF2		4001D38E	2068	LDR	R0,	
4CF3		4001D390	10B9	CBNZ	0x4	
4CF4		4001D398	0D4D	LDR	R5,	
4CF5		4001D39A	7D44	ADD	R5,	
4CF6		4001D39C	2D68	LDR	R5,	
4CF7		4001D39E	2C68	LDR	R4,	
4CF8		4001D3A0	1F34	ADDS	R4,	
4CF9		4001D3A2	1F0424F0	BIC	R4,	
4CFA		4001D3A6	A619	ADDS	R6,	
4CFB		4001D3A8	3046	MOV	R0,	
4CFC		4001D3AA	5EEFF2F7	BLX	0x3	
4CFD		40010268	90002DE9	STMDB	R13	
4CFE		4001026C	2D70A0E3	MOV	R7,	
4CFF		40010270	000000EF	SVC	0x00000000	
4D00		FFFF0008	10F49FE5	LDR	PC, [PC, 0x00000410] ; [FFFF0420]	
4D01		C0023EC0	48D04DE2	SUB	R13_SVC, R13_SVC, 0x00000048	
4D02		C0023EC4	FF1F8DE8	STMIA	R13_SVC, {R0, R1, R2, R3, R4, R5, R6, R7, R8, R9,	
4D03		C0023EC8	3C808DE2	ADD	R8, R13_SVC, 0x0000003C	
4D04		C0023ECC	006048E9	STMDB	R8, {R13, R14}^	
4D05		C0023ED0	00804FE1	MRS	R8, SPSR_SVC	
4D06		C0023ED4	3CE08DE5	STR	R14_SVC, [R13_SVC, 0x0000003C] ; [D493BFEC]	
4D07		C0023ED8	40808DE5	STR	R8, [R13_SVC, 0x00000040] ; [D493BFF0]	
4D08		C0023EDC	44008DE5	STR	R0, [R13_SVC, 0x00000044] ; [D493BFF4]	
4D09		C0023EE0	00B0A0E3	MOV	R11, 0x00000000	
4D0A		C0023EE4	94C09FE5	LDR	R12, [PC, 0x00000094] ; [C0023F80]	
4D0B		C0023EE8	00C09CE5	LDR	R12, [R12] ; [C02D7D08]	
4D0C		C0023EEC	10CF01EE	MCR	p15, 0, R12, c1, c0, 0	
4D0D		C0023EF0	800008F1	CPSIE	I	
4D0E		C0023EF4	AD96A0E1	MOV	R9, R13_SVC, LSR #13	
4D0F		C0023EF8	8996A0E1	MOV	R9, R9, LSL #13	

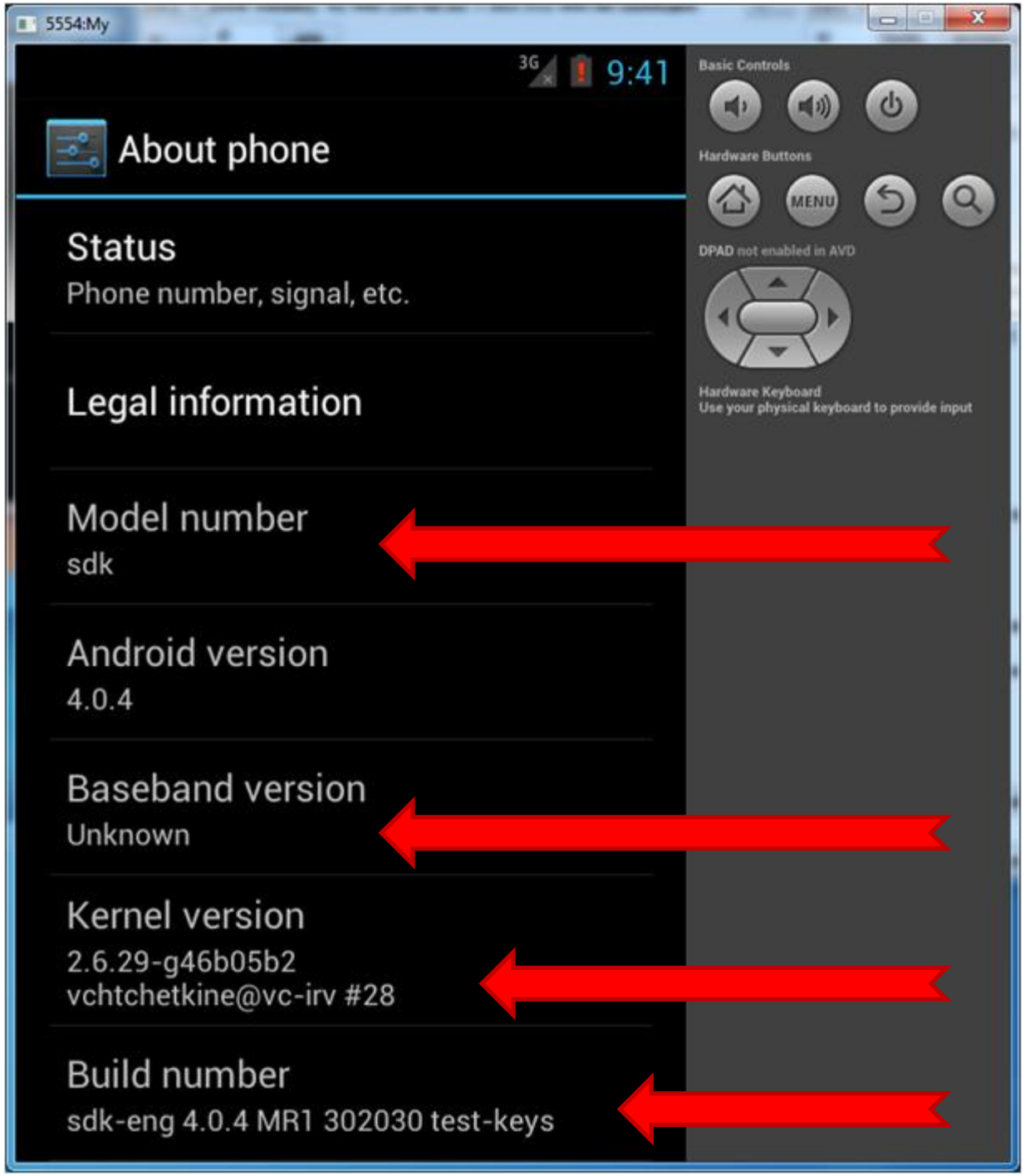
Трасса

Граф зависимостей

Параметры

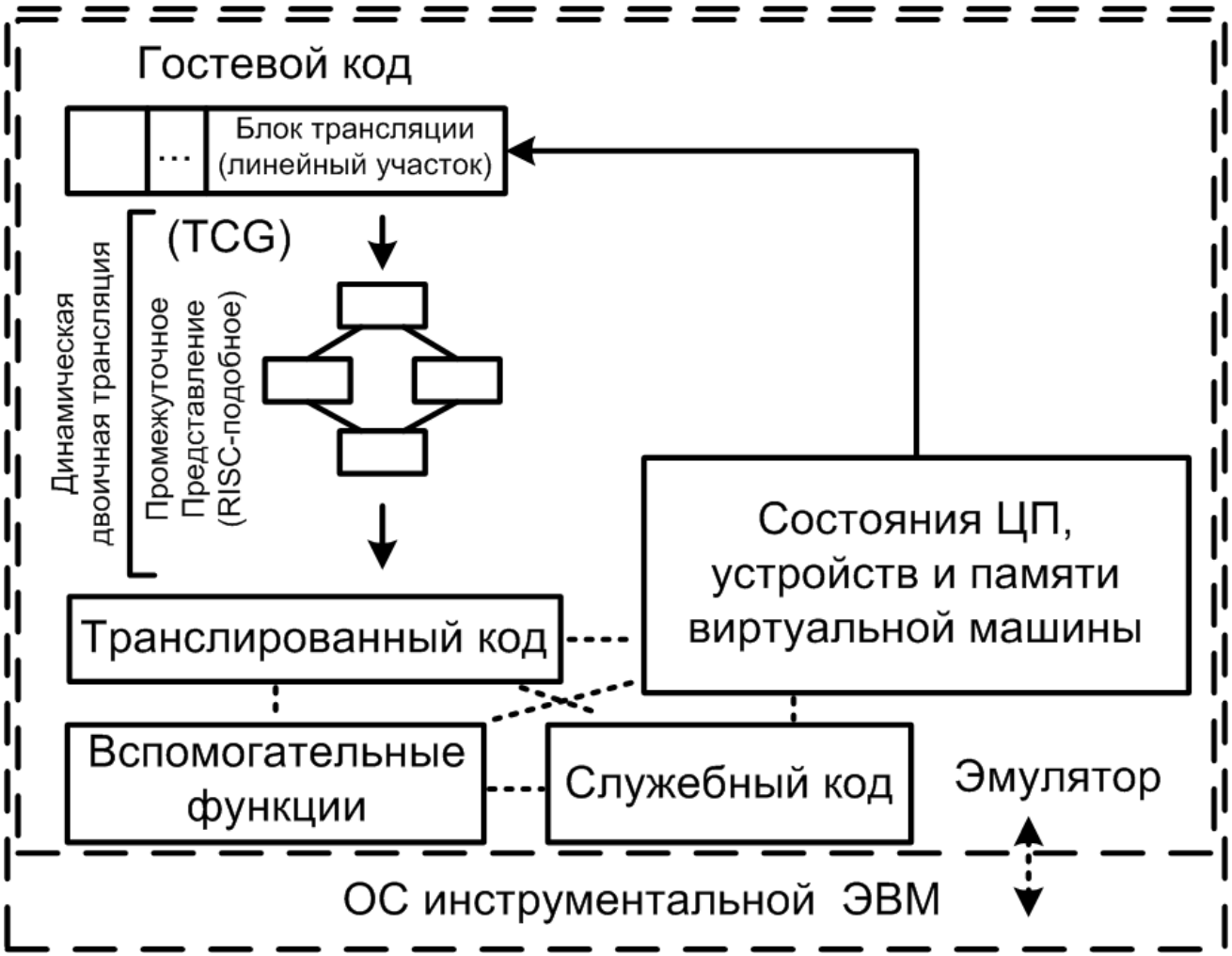
Игнорировать изменение шагов трассы



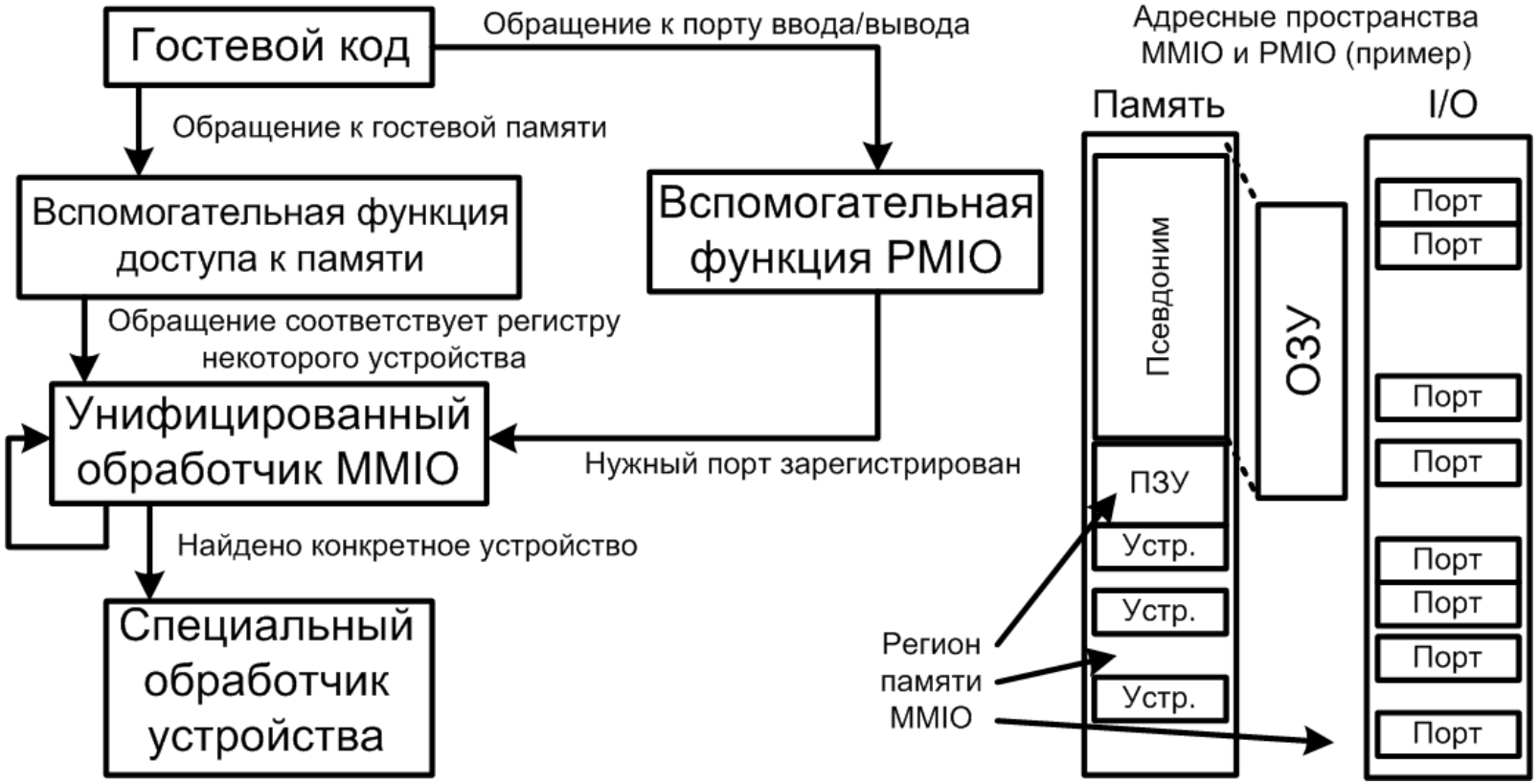


- Qemu-Android реализует виртуальную платформу goldfish
- Распространяется специальная сборка Android для выполнения в эмуляторе
- Ведется разработка более свежей платформы ranchu (64-разрядный ARM)
- Нет возможности выполнять извлеченные прошивки реальных устройств

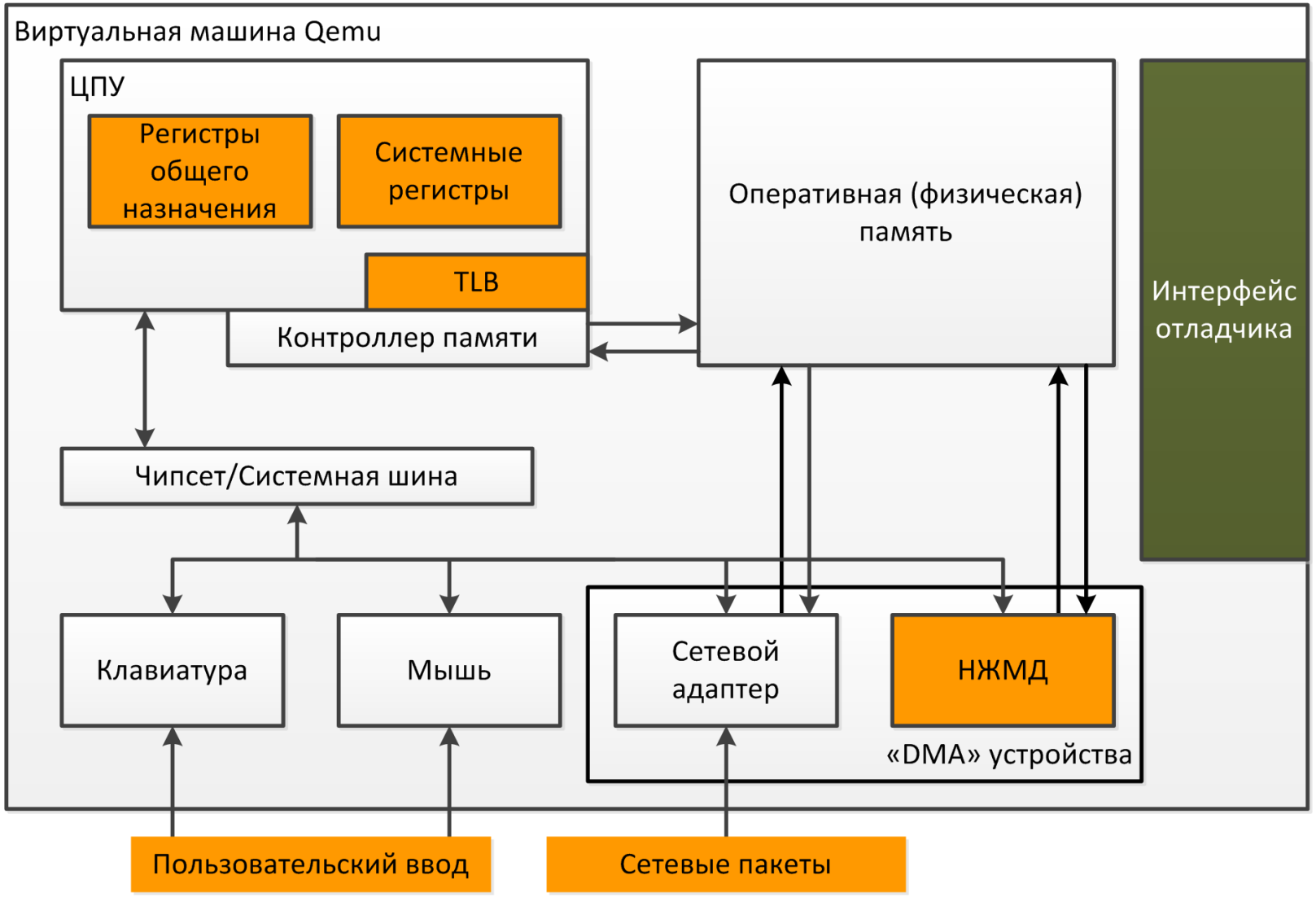
Устройство эмулятора Qemu



Обращение к периферийным устройствам



Эмулятор QEMU: сбор данных для динамического анализа



Проект iEMU: динамический анализ платформы iOS

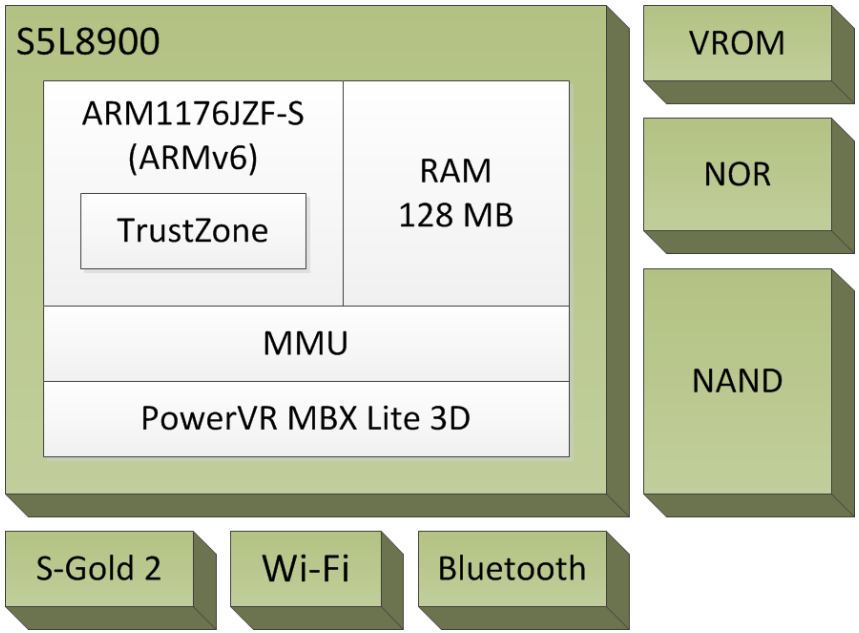
- Полносистемный программный эмулятор для анализа исполняемого кода iOS (и приложений)
  - Код эмулятора основан на Qemu 0.14
  - Разработчик – Chris Wade (cmw, Niacin)
- Заявленные возможности
  - Сборка на Windows и Linux
  - Выполнение встроенного ПО и ядра iOS
  - Возможность подключения отладчика
    - gdb, IDA Pro

	iPhone		iPad
	2G	4	1G
iBoot			
LLB			
ядро iOS			
OpenIBoot			
Bootrom			

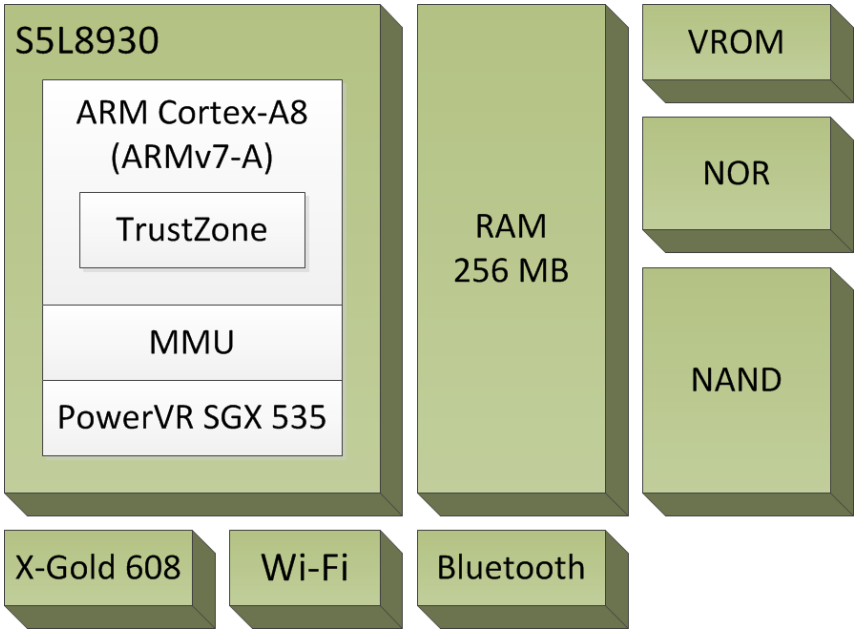
- Последние опубликованные результаты – декабрь 2013. Проект остановлен из-за судебного преследования.

Аппаратура

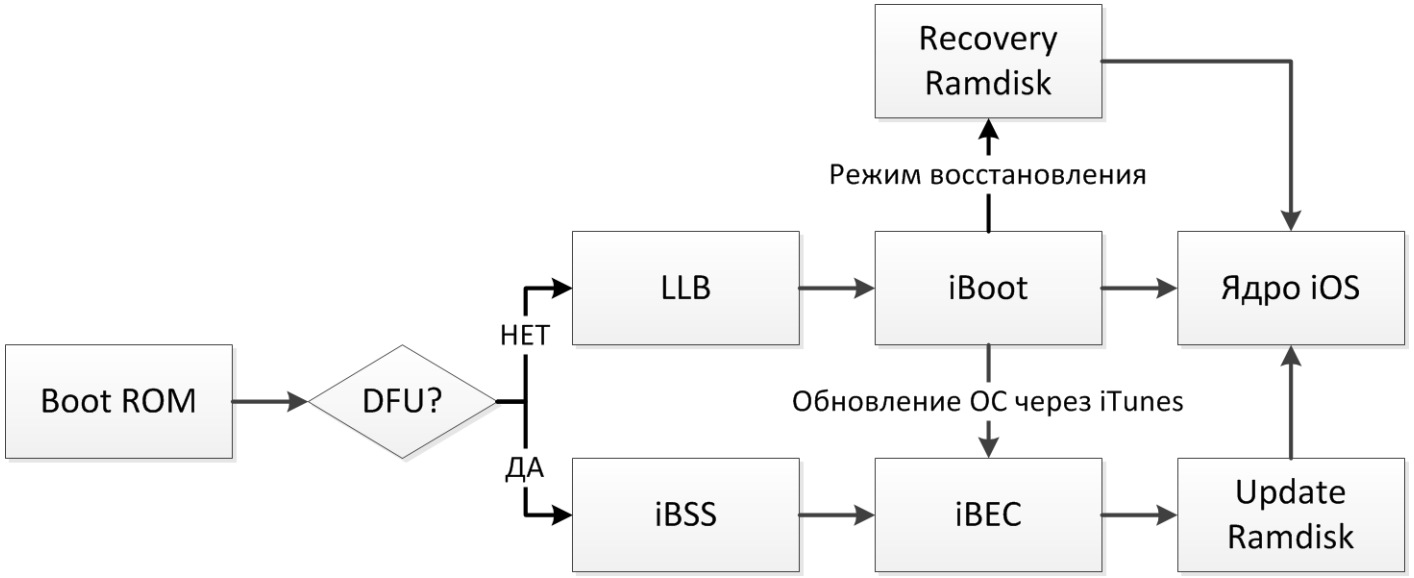
iPhone



iPad

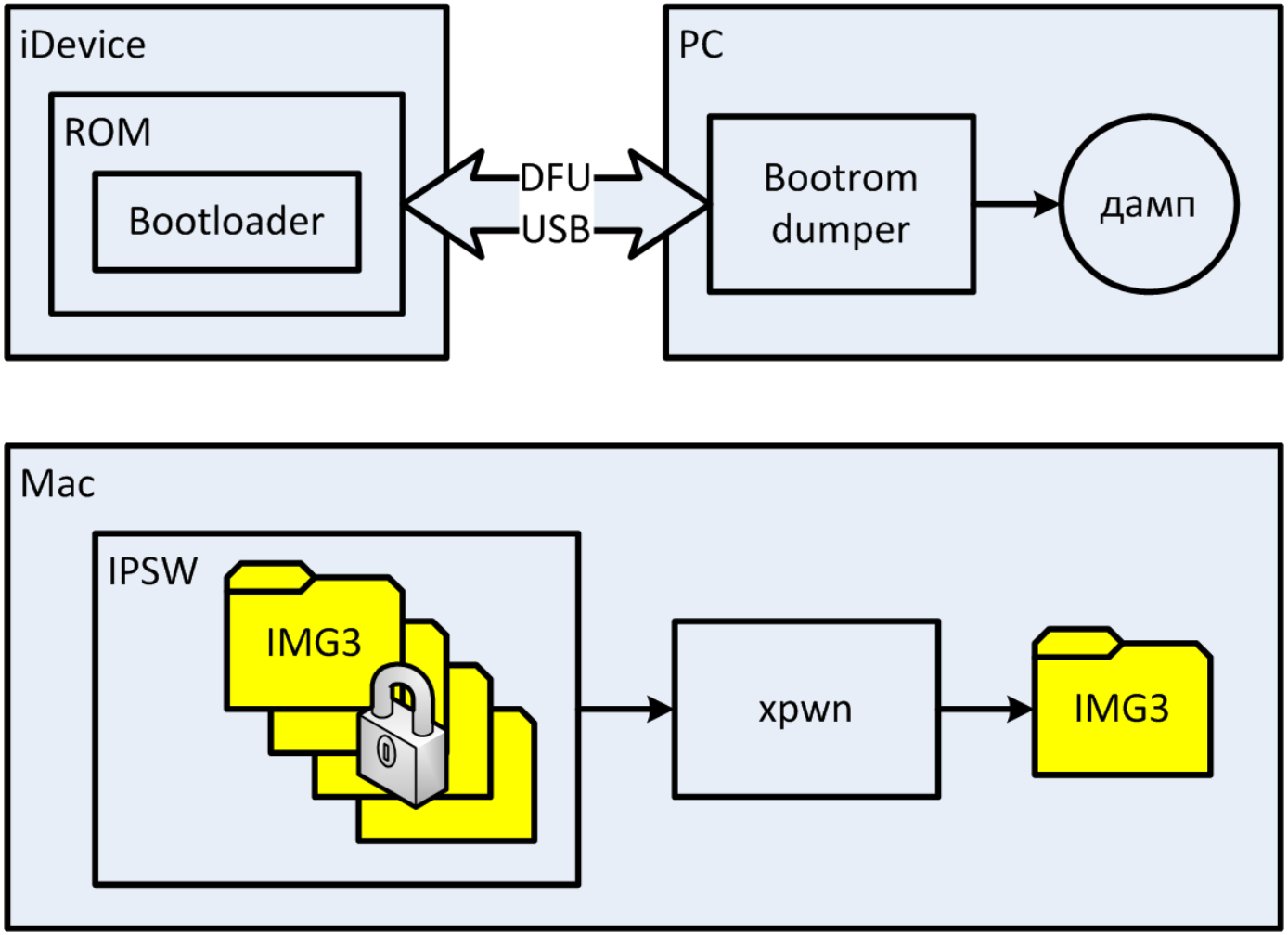


Процесс загрузки



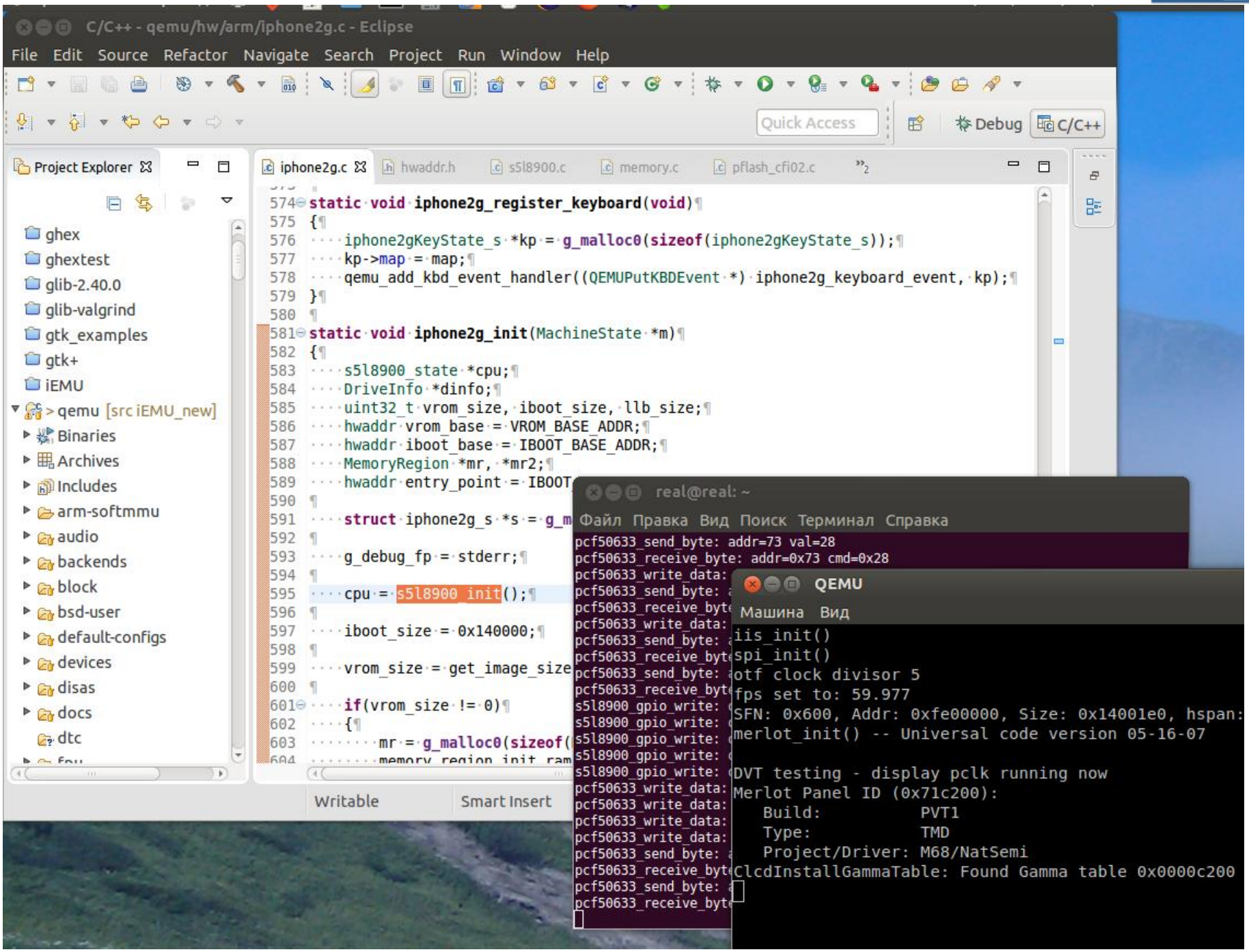
- Каждый этап загрузки Bootrom – iBoot
  - Устанавливает обработчики прерываний
  - …
  - Проверяет целостность кода следующего этапа загрузки
  - Копирует в определенные адреса памяти следующий загрузчик и передает ему управление

Извлечение кода загрузчиков



Добавленная в Qemu виртуальная аппаратура

- Виртуальные платформы iPhone2G и iPad1G (SoC s5l8900 и s5l8930)
  - Контроллер прерываний PL192
  - Шина Serial Peripheral Interface (SPI)
  - Шина Inter-Integrated Circuit (I2C)
  - NOR память
  - NAND память
  - UART
  - Аппаратная криптография: AES, SHA1
  - Часы и таймеры
- Контроллер LCD дисплея
- Контроллер аппаратных кнопок
- USB – TCP адаптер
- Контроллер электропитания PCF50633
- Изменения в скриптах сборки, скиннинг
- Используется уже существующий модуль архитектуры ARM


C/C++ - qemu/hw/arm/iphone2g.c - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Debug
C/C++
Project Explorer
ghex
ghextest
glib-2.40.0
glib-valgrind
gtk_examples
gtk+
iEMU
qemu [src iEMU_new]
Binaries
Archives
Includes
arm-softmmu
audio
backends
block
bsd-user
default-configs
devices
disas
docs
dtc
iphone2g.c
hwaddr.h
s5l8900.c
memory.c
pflash_cfi02.c
static void iphone2g_register_keyboard(void)
{
iphone2gKeyState_s *kp = g_malloc0(sizeof(iphone2gKeyState_s));
kp->map = map;
qemu_add_kbd_event_handler((QEMUPutKBDEvent *) iphone2g_keyboard_event, kp);
}
static void iphone2g_init(MachineState *m)
{
s5l8900_state *cpu;
DriveInfo *dinfo;
uint32_t vrom_size, iboot_size, llb_size;
hwaddr vrom_base = VROM_BASE_ADDR;
hwaddr iboot_base = IBOOT_BASE_ADDR;
MemoryRegion *mr, *mr2;
hwaddr entry_point = IBOOT
struct iphone2g_s *s = g_m
g_debug_fp = stderr;
cpu = s5l8900_init();
iboot_size = 0x140000;
vrom_size = get_image_size
if(vrom_size != 0)
{
mr = g_malloc0(sizeof(
Writable
Smart Insert
real@real: ~
Файл Правка Вид Поиск Терминал Справка
pcf50633_send_byte: addr=73 val=28
pcf50633_receive_byte: addr=0x73 cmd=0x28
pcf50633_write_data:
pcf50633_send_byte:
pcf50633_receive_byte
pcf50633_write_data:
pcf50633_send_byte:
pcf50633_receive_byte
pcf50633_send_byte:
pcf50633_receive_byte
s5l8900_gpio_write:
s5l8900_gpio_write:
s5l8900_gpio_write:
s5l8900_gpio_write:
s5l8900_gpio_write:
pcf50633_write_data:
pcf50633_write_data:
pcf50633_write_data:
pcf50633_write_data:
pcf50633_send_byte:
pcf50633_receive_byte
pcf50633_send_byte:
pcf50633_receive_byte
QEMU
Машина Вид
iis_init()
spi_init()
otf clock divisor 5
fps set to: 59.977
SFN: 0x600, Addr: 0xfe00000, Size: 0x14001e0, hspan:
merlot_init() -- Universal code version 05-16-07
DVT testing - display pclk running now
Merlot Panel ID (0x71c200):
Build: PVT1
Type: TMD
Project/Driver: M68/NatSemi
ClcdInstallGammaTable: Found Gamma table 0x0000c200

IDA - C:\Documents and Settings\Пользователь\Рабочий стол\iphone1\iphone1-2.idb (<GDB remote

File Edit Jump Search View Debugger Options Windows Help

Library function Data Regular function Unexplored Instruction

IDA View-PC | Pseudocode-D | Pseudocode-E

```
1  void __fastcall sub_18004244()
2  {
3    char *v0; // r4@1
4    int v1; // r0@3
5
6    v0 = 0;
7    // initialize virtual memory
8    p_first_descriptor = descriptors;
9    do
10   {
11     something_with_TLB(v0, v0, 0, 0);
12     v0 += 0x100000;
13   }
14   while ( v0 );
15   setup_virtual_memory();
16   sub_180001F4(3);
17   sub_180001FC(p_first_descriptor);
18   invalidate_unified_TLB();
19   sub_18003E8A(v1);
20   invalidate_unified_TLB();
21 }
```

Заключение

- Программные эмуляторы обеспечивают значительно более широкие возможности для контроля и анализа исполняемого кода
  - Honeypot + возможность воспроизведения
    - поиск 0-day-уязвимостей
  - Фаззинг белого ящика
  - Отладка условно работоспособного кода
  - Автоматизация анализа собранных трасс
- PoC – iEMU
- Риски подхода
  - Отсутствие документации для виртуальной аппаратуры

Спасибо за внимание!

Вопросы?