SECON'2017

IX МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

#SECONRU

РАССЛЕДОВАНИЕ КИБЕР-ПРЕСТУПЛЕНИЙ: ИЗВЛЕКАЕМ СКРИНШОТЫ ИЗ ДАМПОВ ОПЕРАТИВНОЙ ПАМЯТИ

Анатолий Тыкушин, Михаил Болдырев

студенты программы “Secure System and Network Engineering”
АНО ВО “Университет Иннополис”

Our Agenda

“Лет ми спик фром май харт ин инглиш”

О чем это мы?

Forensics

1. **Сбор данных**
   - Как доказать, что это именно данные из источника X?
   - Как доказать, что ни один бит информации не утерян?

2. **Извлечение цифровых улик**
   - Обработка полученных данных
   (ловушки, false-positives, шифрование, стеганография)

3. **Отчет эксперта-криминалиста**
   - Обоснованность с технической точки зрения
   - Действия в рамках закона

Stage 1: Memory acquisition

Target: Physical memory

О чем это мы?

DEFT

Digital Evidence & Forensics Toolkit

CLI:

- dfcldd
- AFFLIB
- Libewf

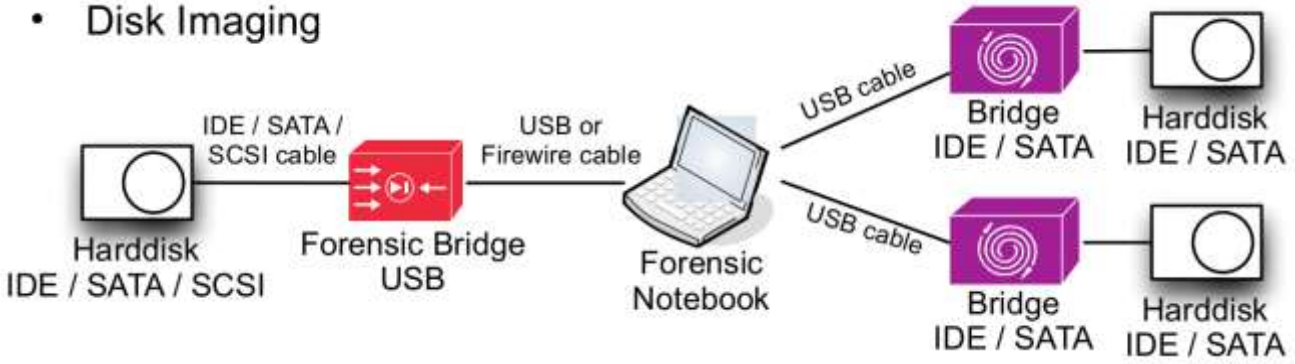
GUI-based:

- Guymager
- AIR

Commercials:

Encase, X-Ways Forensics, etc.

- Disk Imaging



О чем это мы?

Stage 1: Memory acquisition

Target: RAM

О чем это мы?

Where to fail?

Rootkits

Malicious activity

Memory losses

User-level

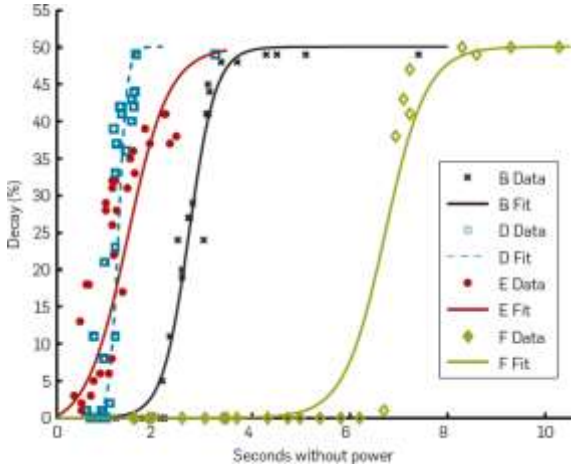
- незначительный урон;
- модификация системных команд (ls, ps, du, df);
- изменение информации о диске, запущенных процессах и пр.

Kernel-level

- серьезный урон;
- модификация ОС для произвольных реакций системы;
- позволяет прятать файлы, блоки физической и виртуальной памяти;

Kernel-level (ctd.)

- подменяем драйверы
- обходим виртуализацию памяти
- подменяем shared libraries (не сработает в случае статической линковки)



О чем это мы?

Stage 2: Extract digital evidence

Hands-on

- долго, рискованно
- кропотливо
- познавательно
- разнообразно

Automate

- быстро
- шаблонно
- не на все случаи жизни
- формализовано

О чем
это мы?

Stage 2: Extract digital evidence

Rekall

second SL LOOK

Belkasoft
forensics made easier

О чем
это мы?

Stage 2: Extract digital evidence

=== PREPARE VOLATILITY ===

BUILD OWN PROFILE

✓ Module.dwarf

✓ System.map

✓ Load built profile

✓ Ready for stage 2

О чем
это мы?

Startpack


```
volatility
↳ plugins
   ↳ linux
      ↳ common
         ↳ AbstractLinuxCommand
            ↳ <YOURCLASS>
```

```
calculate()  ──→  render_table()
                  render_text()
                  render_custom_format()
```

ask Google for help

Как это работает?

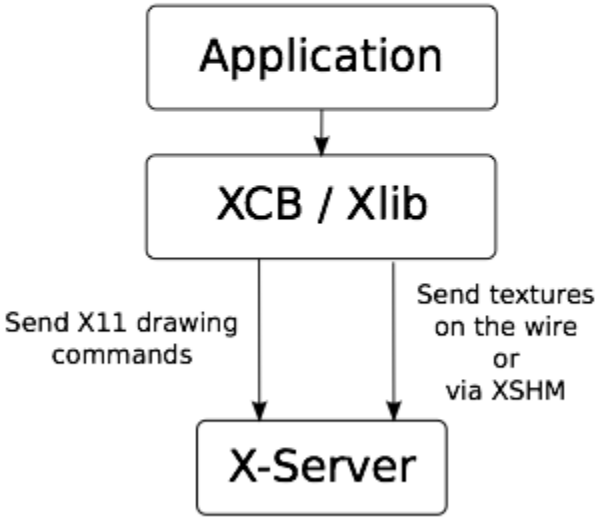
1. Знакомимся с графическим стеком Linux
   User-space, kernel, hardware
2. Анализируем информационные потоки
   Как “рождается” картинка
3. Делаем выводы
   Как эти знания применимы для нашего исследования?

Как это работает?

Linux Graphics Stack Behind the scenes

User-space Layer

2D application's rendering

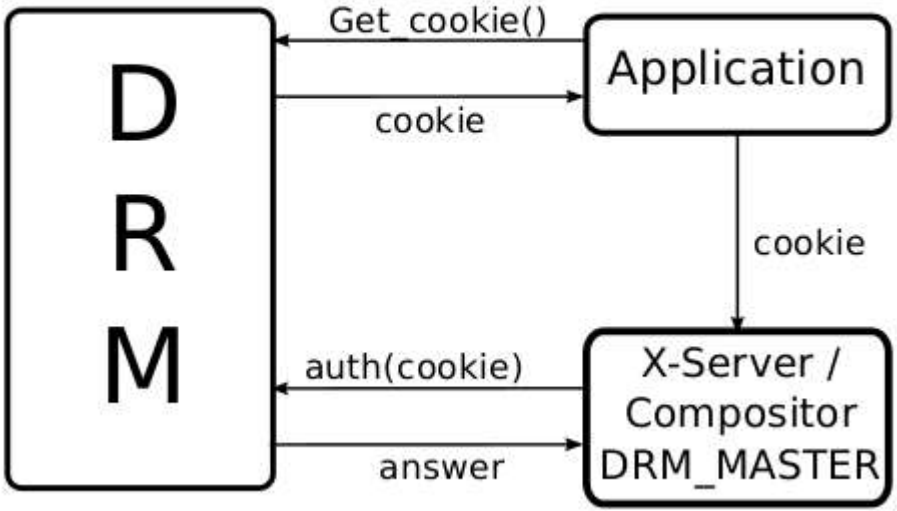


http://phd.mupuf.org/files/fosdem2013_drinext_drm2.pdf

Как это работает?

Linux Graphics Stack Behind the scenes

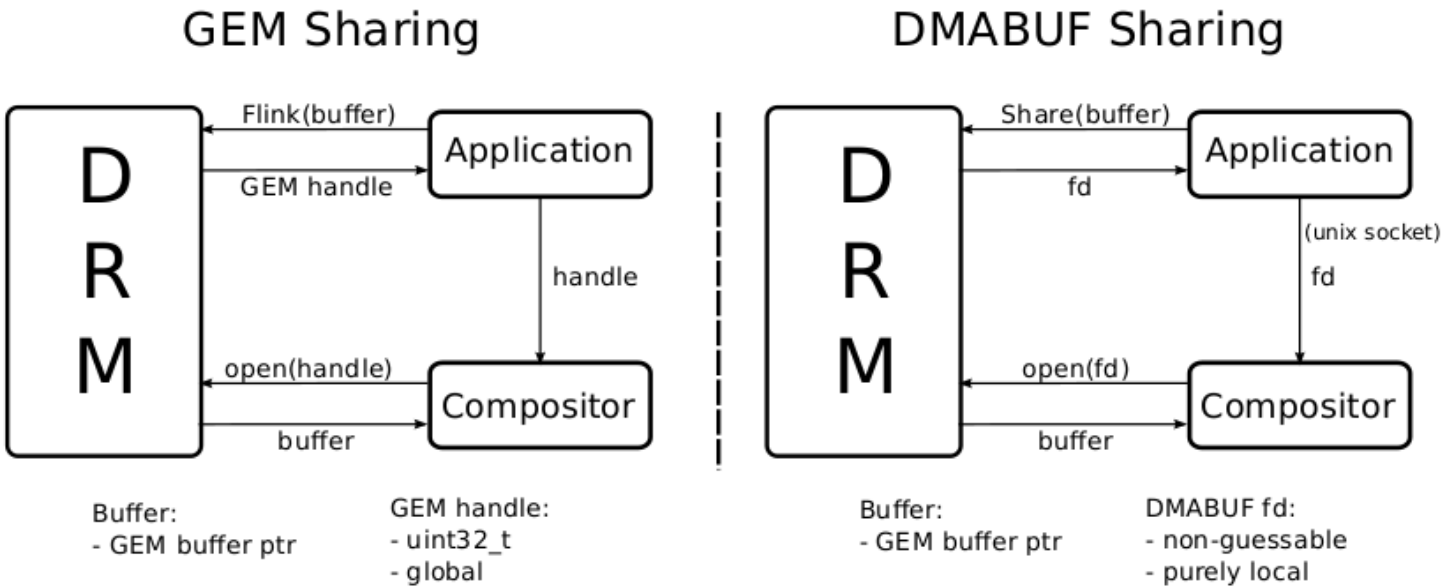
DRM Auth



http://phd.mupuf.org/files/fosdem2013_drinext_drm2.pdf

Как это работает?

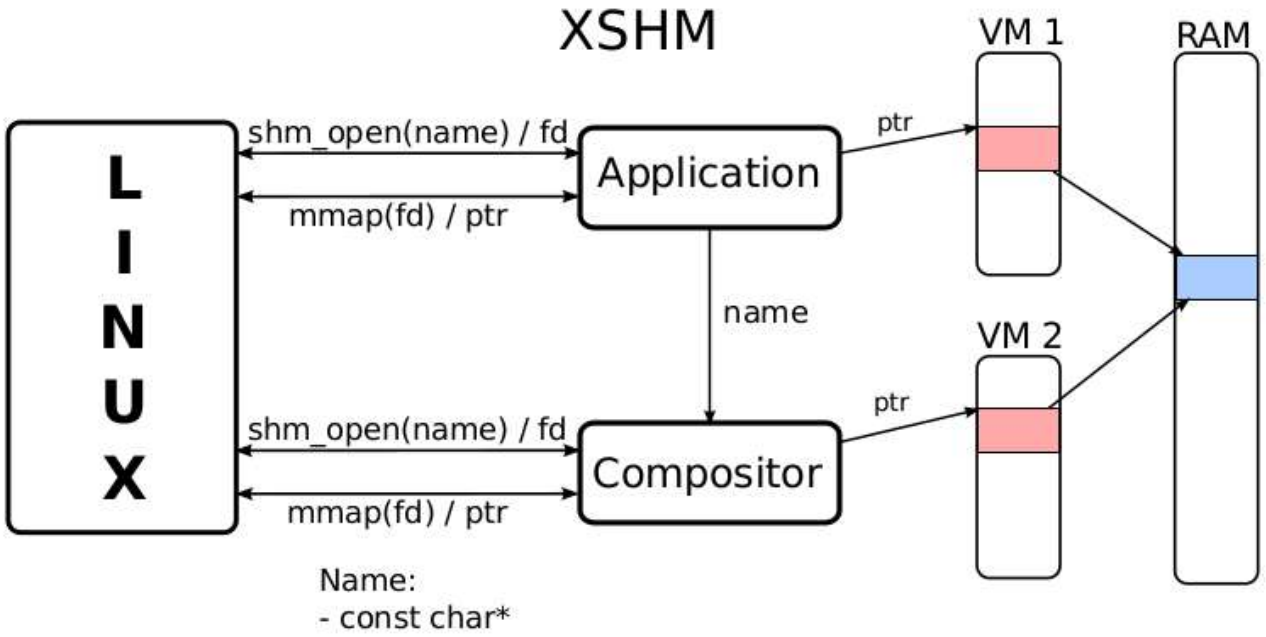
Linux Graphics Stack
Behind the scenes



http://phd.mupuf.org/files/fosdem2013_drinext_drm2.pdf

Как это работает?

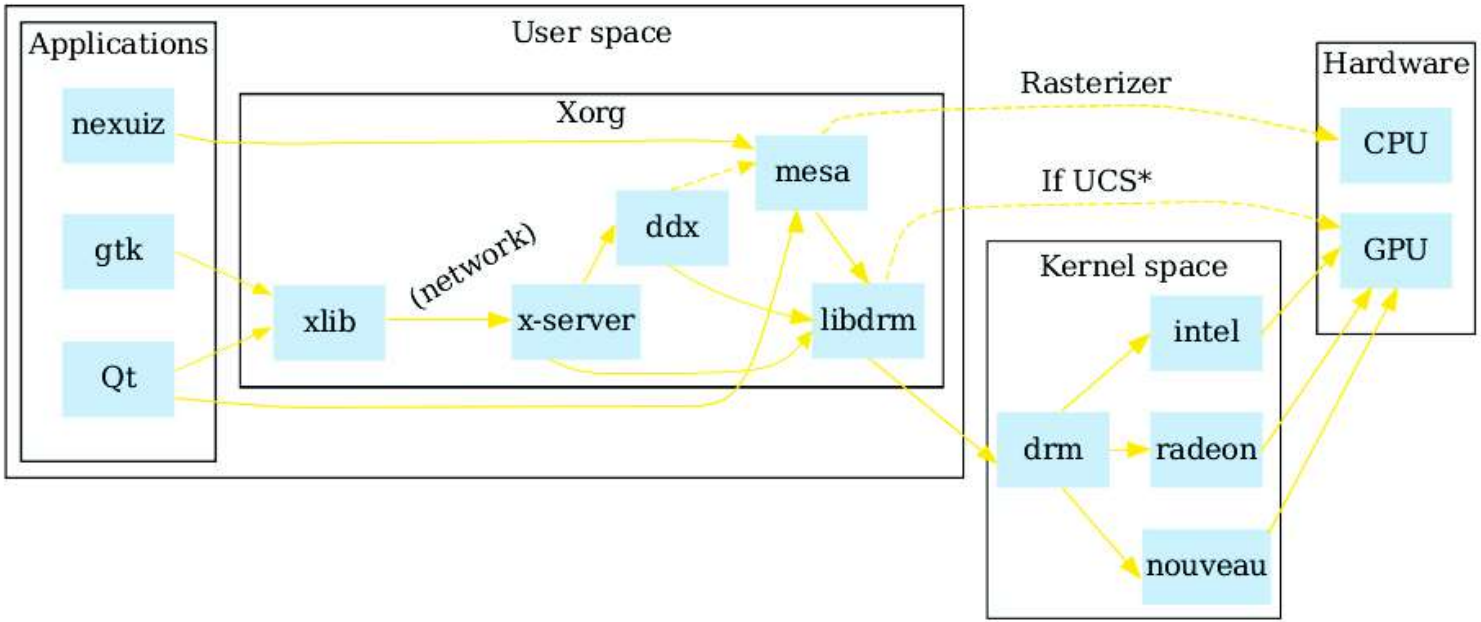
Linux Graphics Stack
Behind the scenes



http://phd.mupuf.org/files/fosdem2013_drinext_drm2.pdf

Как это работает?

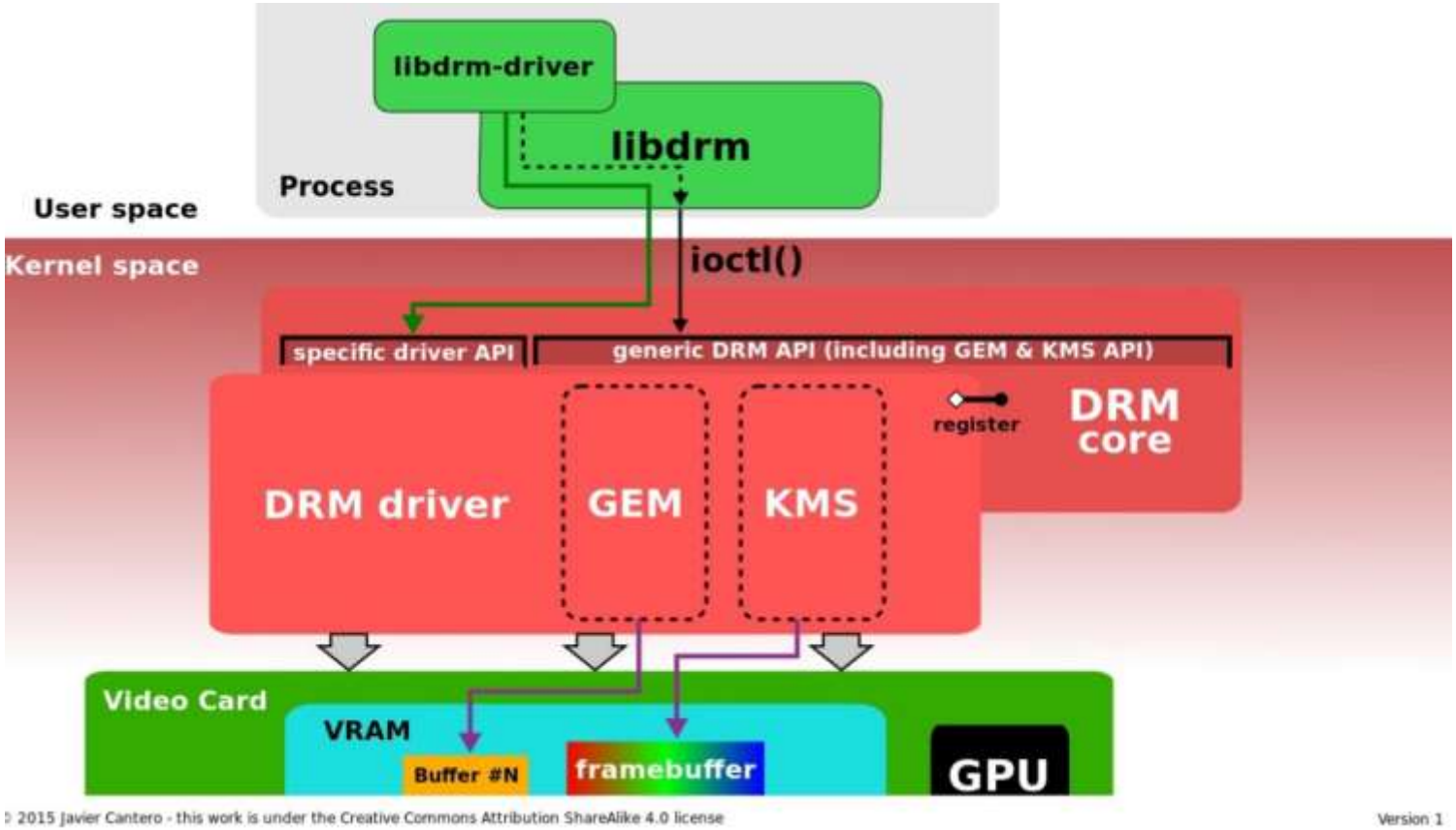
Linux Graphics Stack



http://phd.mupuf.org/files/fosdem2013_drinext_drm2.pdf

Как это работает?

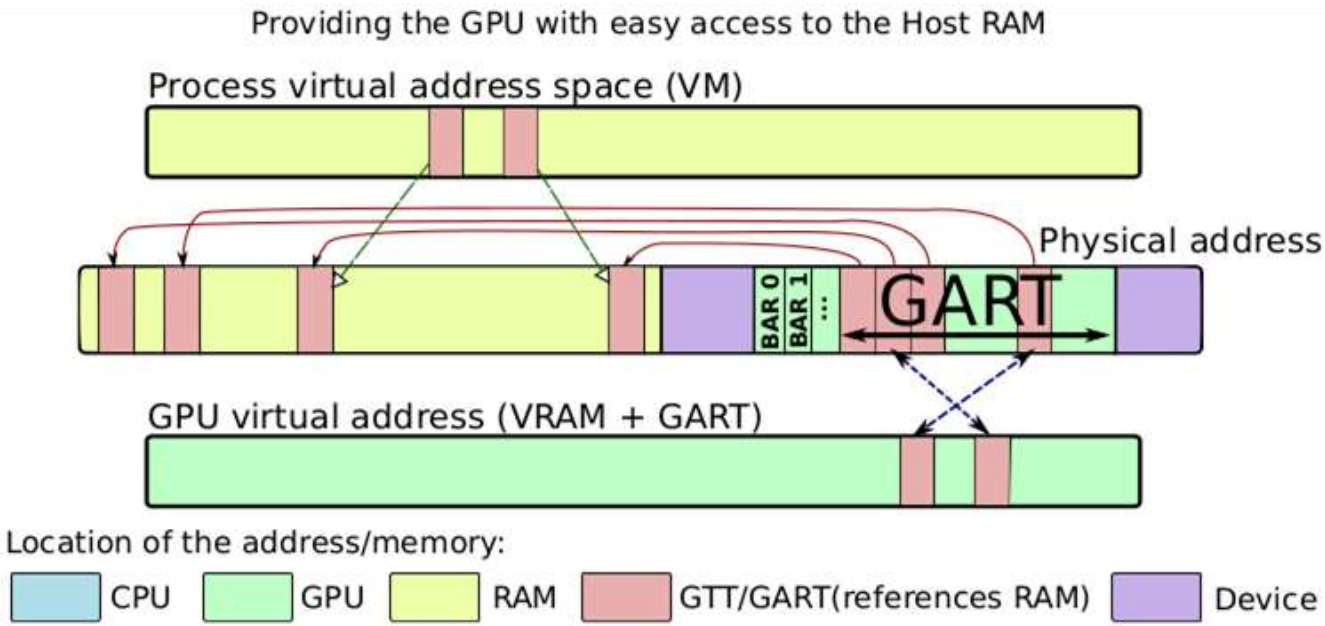
Linux Graphics Stack



https://en.wikipedia.org/wiki/Direct_Rendering_Manager#/media/File:DRM_architecture.svg

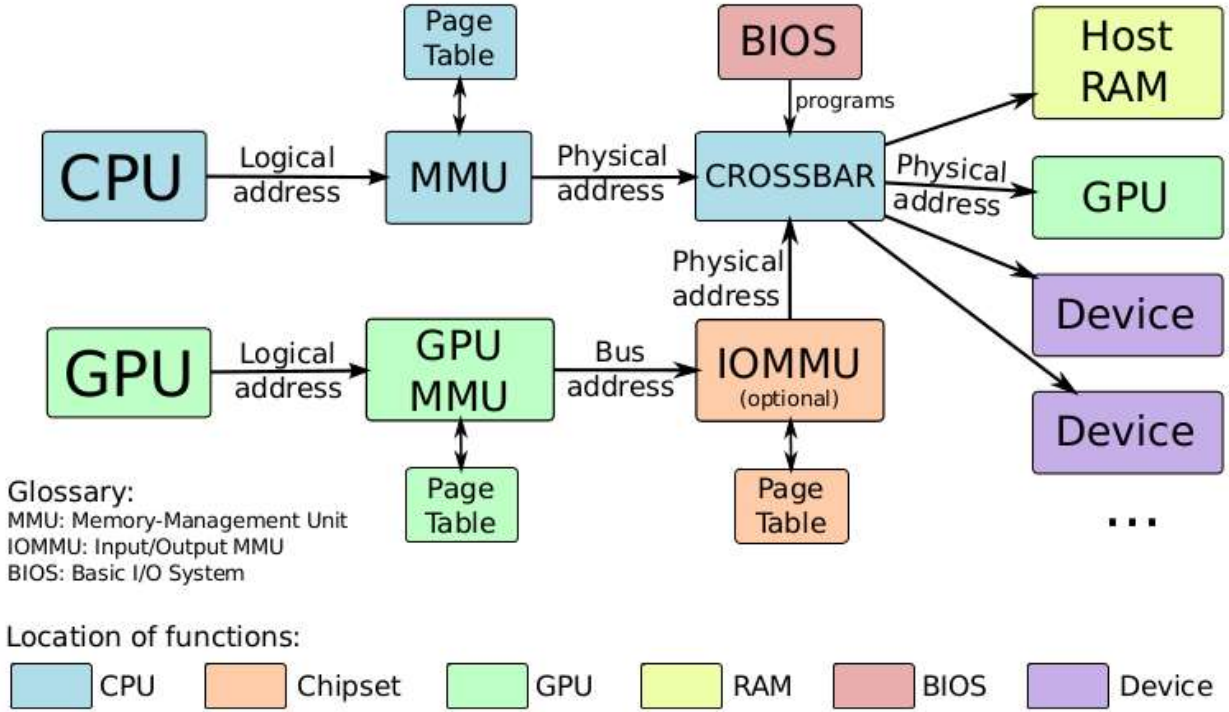
Как это работает?

Linux Graphics Stack
Graphics translation table



Как это работает?

Linux Graphics Stack
CPU & GPU Memory requests routing



Продумываем нашу идею

Одним выстрелом двух зайцев

✓Познавательно для нас, вклад в коммьюнити

~~Костыльно-ориентированный подход~~

⚐ Чтобы не стыдно было показать

Архитектурное решение

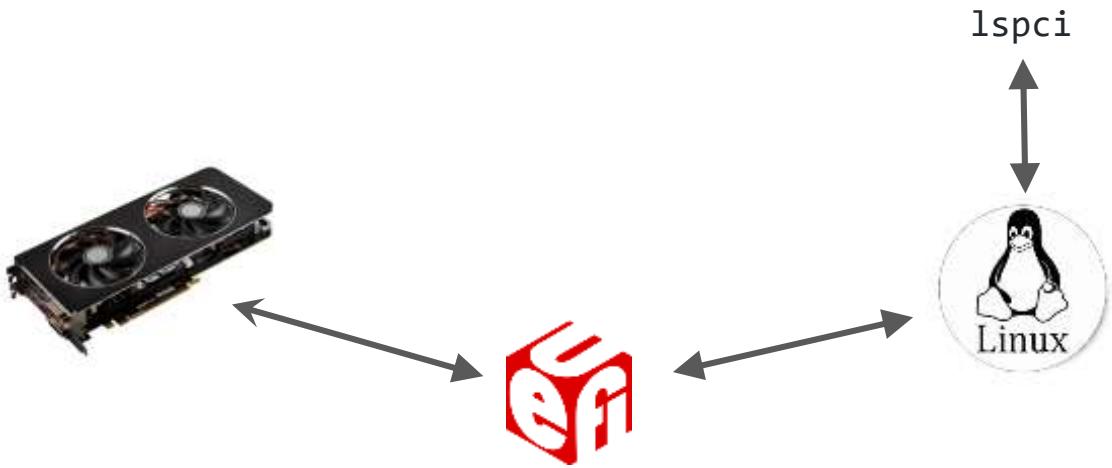
Стек технологий, задачи, этапы, ~~эджайл~~


YOUR PLAN.
REALITY.
HUSTLE + GRIND

Работаем

Graphics memory mapped into RAM

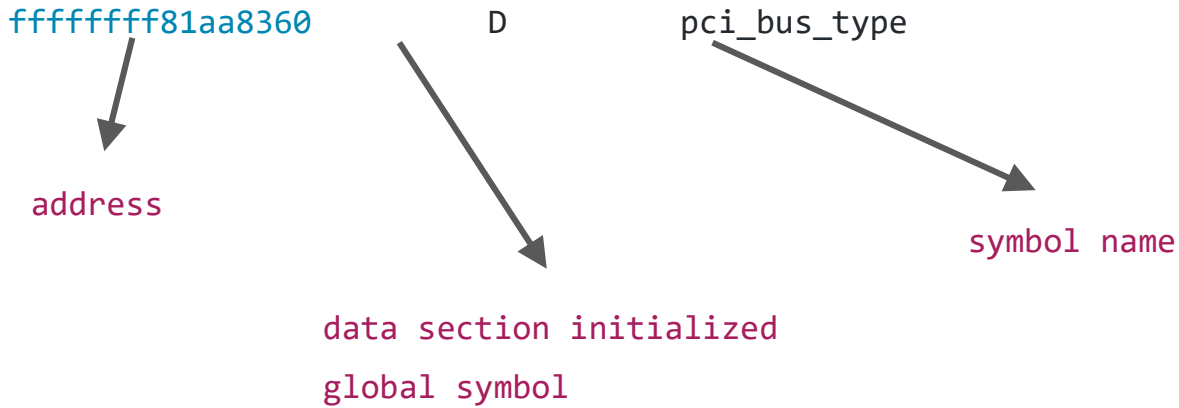
```
01:00.0 VGA compatible controller: Advanced Micro Devices, Inc. [AMD/ATI] Caicos XTX
[Radeon HD 8490 / R5 235X OEM] (prog-if 00 [VGA controller])
            Subsystem: Hewlett-Packard Company Caicos XTX [Radeon HD 8490 / R5 235X OEM]
            Flags: bus master, fast devsel, latency 0, IRQ 27
            Memory at e0000000 (64-bit, prefetchable) [size=256M]
```



Работаем

PCI devices presentation in Linux Kernel

system.map:



system.map sample:

```
000000000000a020 D cpu_llc_id
000000000000a080 D cpu_core_map
000000000000a0c0 D cpu_sibling_map
000000000000a100 D cpu_info
000000000000a1e0 D cpu_number
000000000000a1e8 D this_cpu_off
000000000000a1f0 D x86_cpu_to_apicid
000000000000a240 d pmc_prev_left
000000000000a440 D cpu_hw_events
000000000000b700 d bts_ctx
```

Работаем

PCI devices presentation in Linux Kernel

Структура `sysbus_private` - данные драйвера

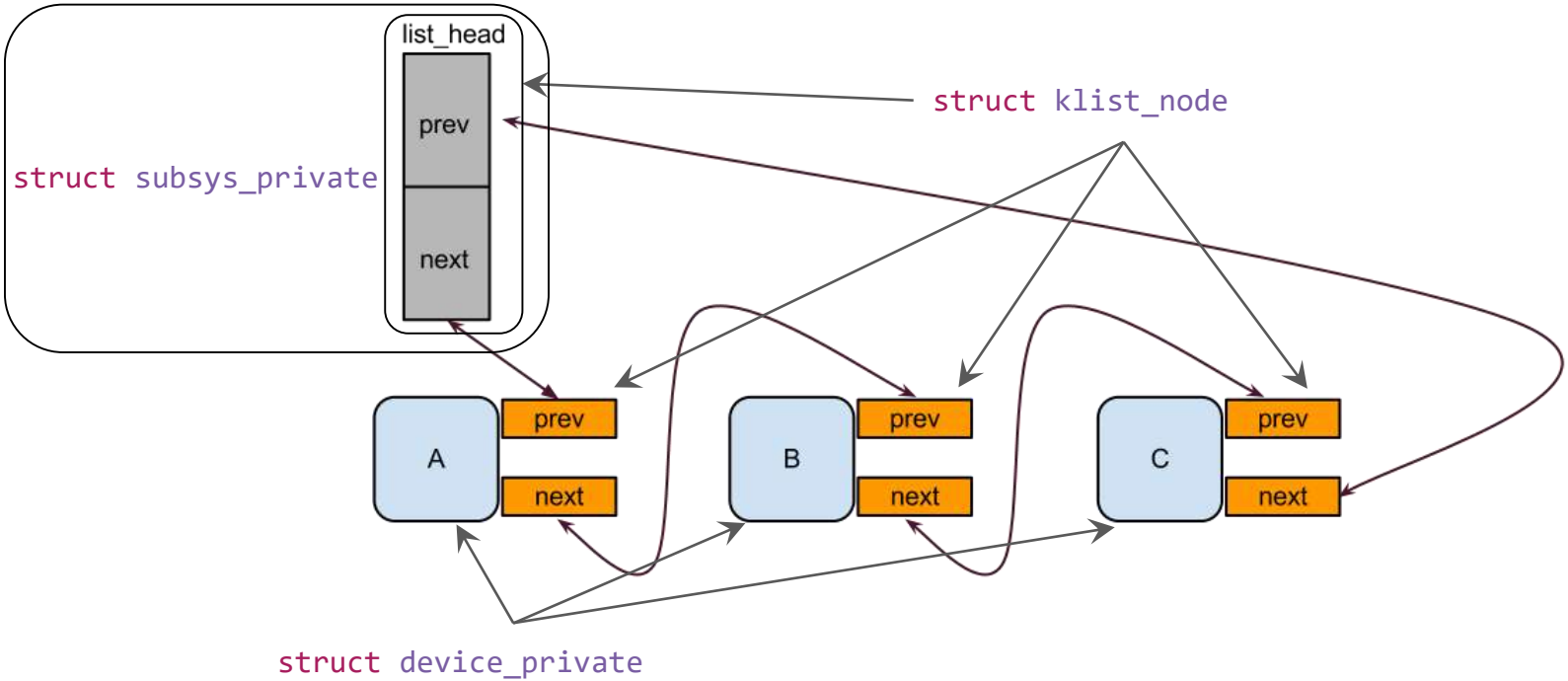
```
struct bus_type {
    …
    struct subsys_private *p;
    …
}
```

Находим по известному сдвигу (DWARF)

Работаем

PCI devices presentation in Linux Kernel

Структура `klist klist_devices` - связный список PCI устройств



Работаем

PCI devices presentation in Linux Kernel

`pci_dev.class` - PCI класс и подкласс устройства (0:1 и 3:0 для видео)

`struct resource` - MMIO & PMIO участки памяти

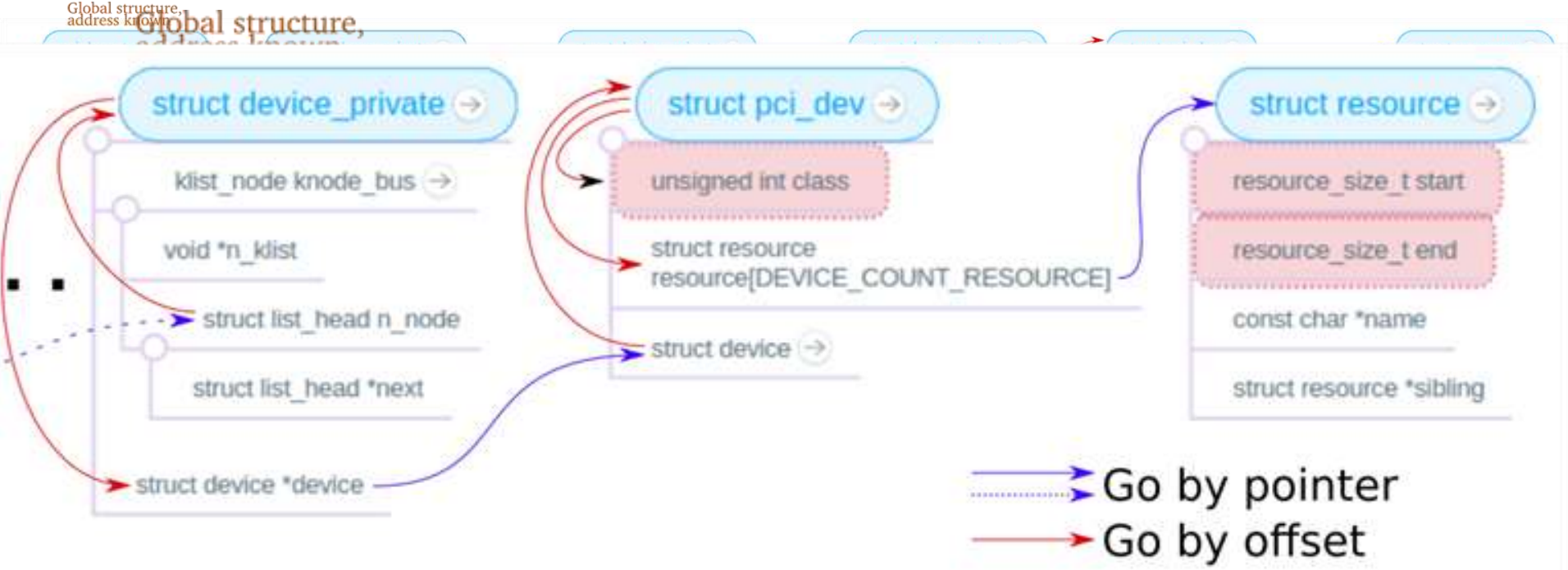
```
struct device_private {
        ...
        struct klist_node knode_bus;
        struct device *device;
        ...
};
```

```
struct pci_dev {
        ...
        unsigned int class;
        struct device dev;
        struct resource resource[];
        ...
}
```

```
struct resource {
        resource_size_t start;
        resource_size_t end;
        ...
};
```

Работаем

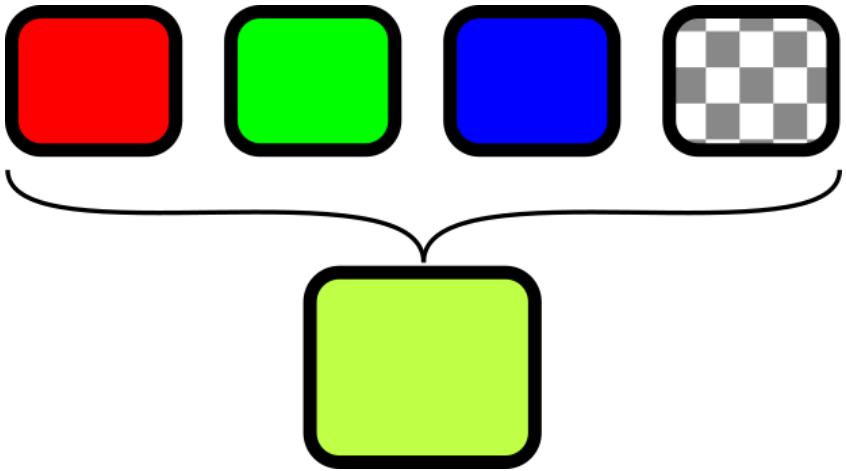
PCI devices presentation in Linux Kernel



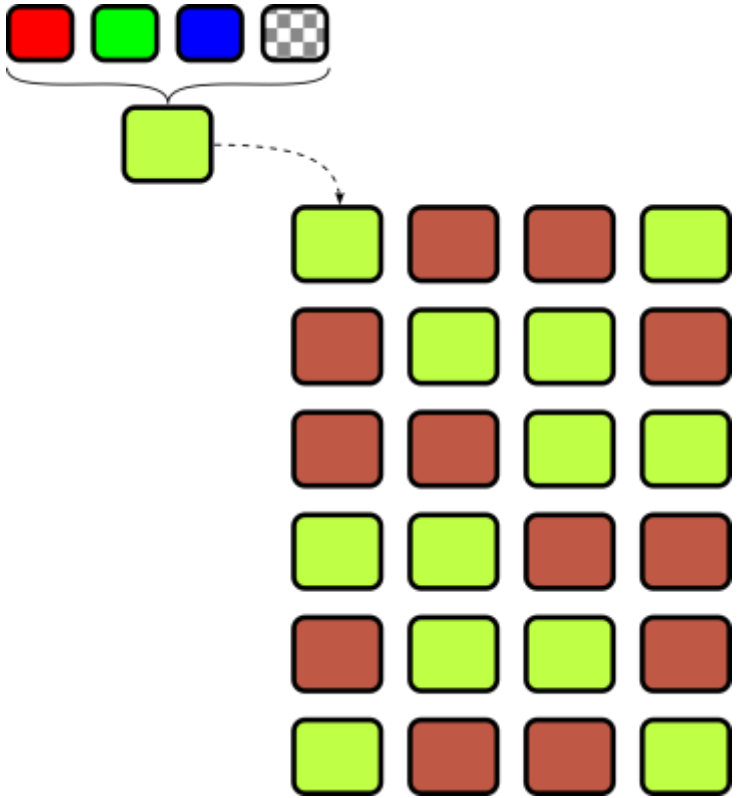
Let's help Tux find all video PCI devices!

Pixel representation

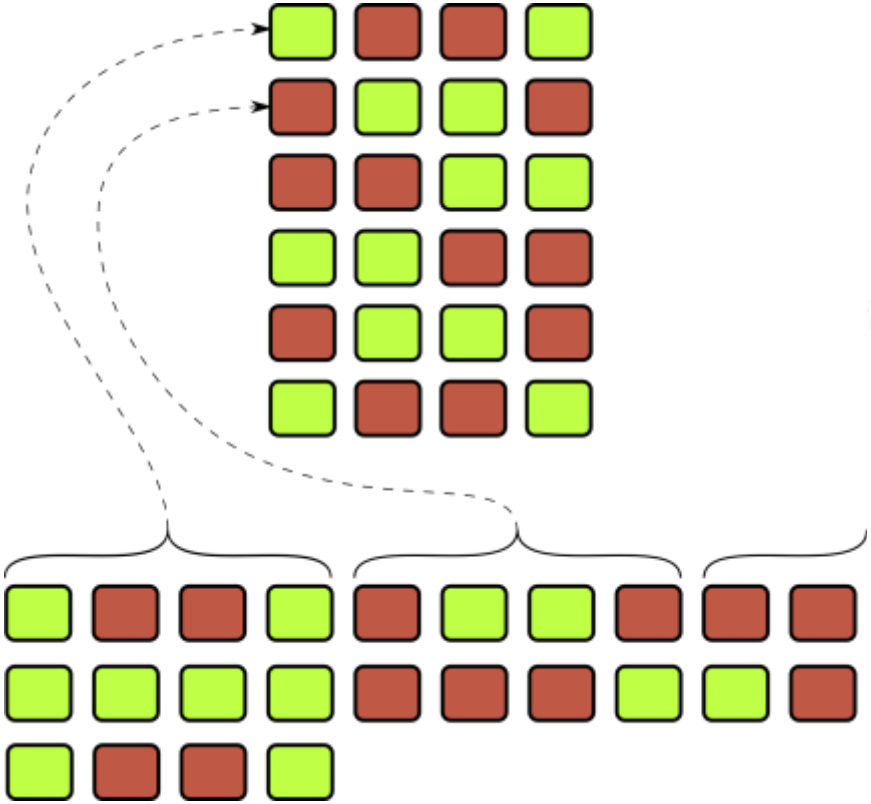
```
__u32 bits_per_pixel;

__u32 line_length;

struct fb_bitfield red;
       red.offset;
```

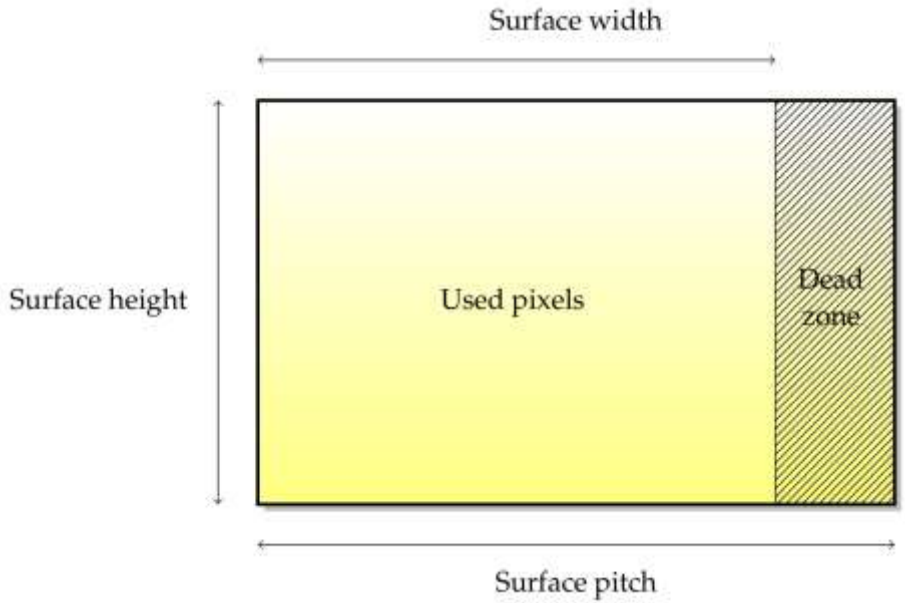


Plane representation



Serial plane representation



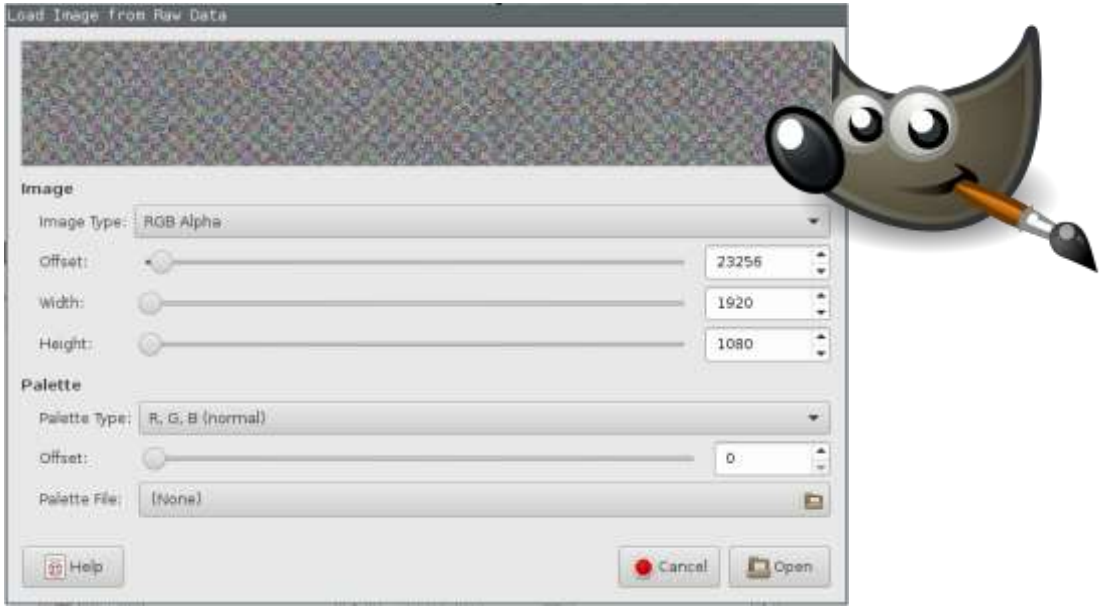


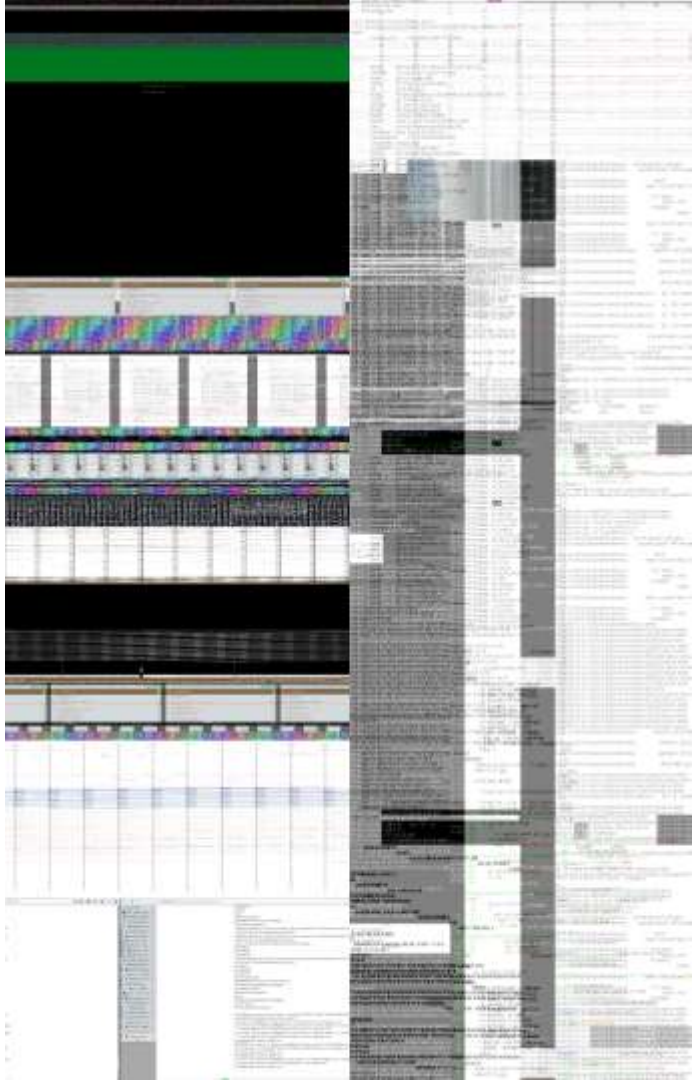
Работаем

Retrieving images

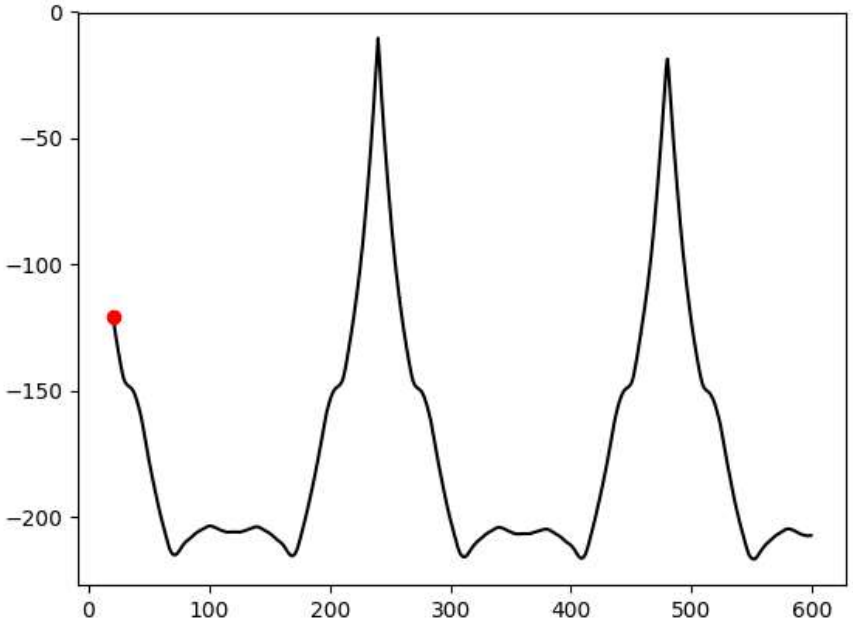
Много разрозненных кадровых буферов с различной геометрией и параметрами отрисовки

Можно исследовать вручную?





Guessing chunk line width



SECON'2017

IX МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

РАССЛЕДОВАНИЕ КИБЕР-ПРЕСТУПЛЕНИЙ: ИЗВЛЕКАЕМ СКРИНШОТЫ ИЗ ДАМПОВ ОПЕРАТИВНОЙ ПАМЯТИ

Анатолий Тыкушин, Михаил Болдырев

студенты программы “Secure System and Network Engineering”
АНО ВО “Университет Иннополис”