Introducción a Mesa

Samuel Iglesias Gonsálvez

siglesias@igalia.com

igalia

Sobre mí

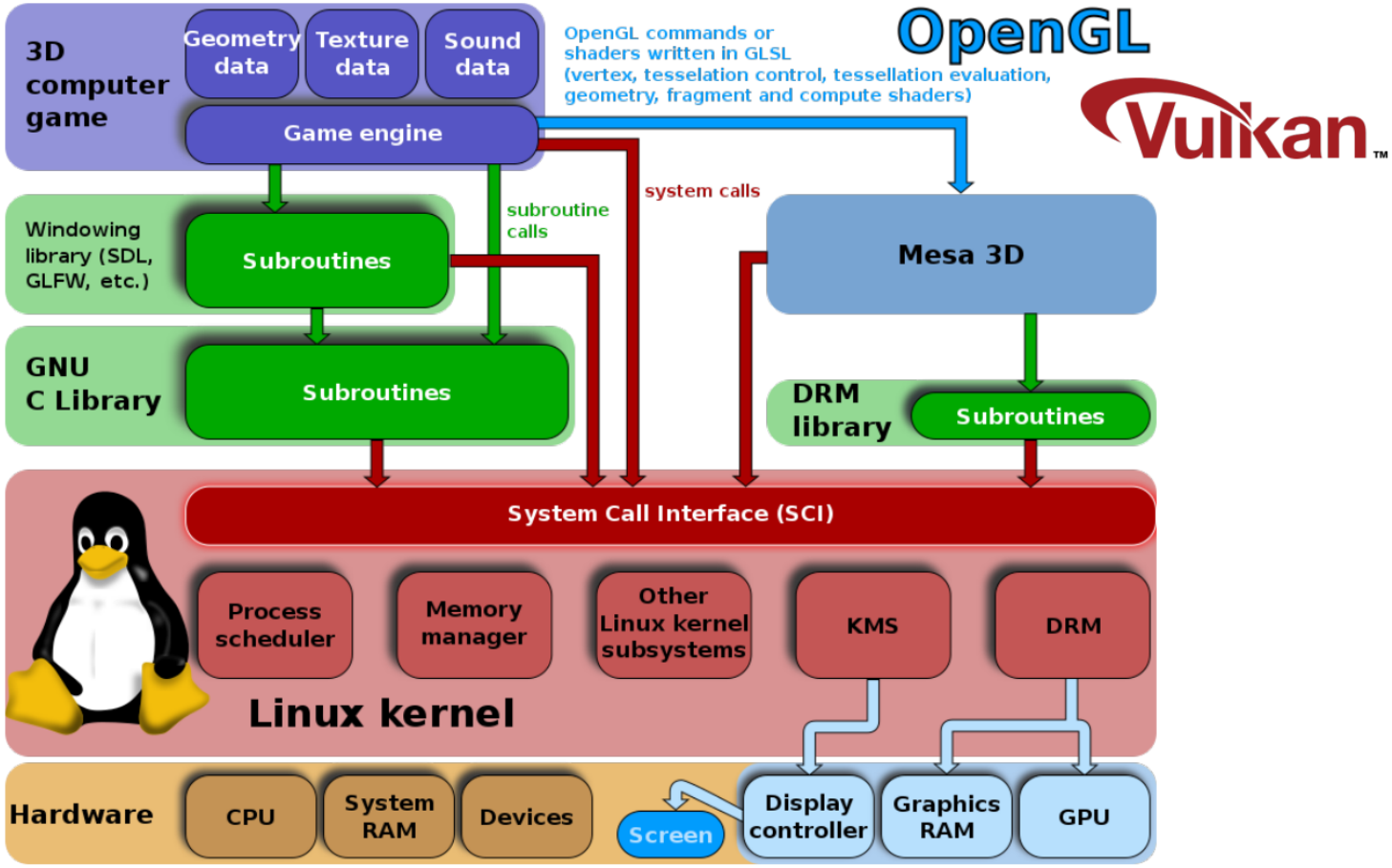
- Miembro del Graphics Team en Igalia.
- Desarrollador de drivers de Mesa desde el año 2014.
  - Principalmente en los drivers libres de GPUs de Intel y Qualcomm Adreno.
- Miembro de la Junta Directiva de **X.Org Foundation** y miembro de **Khronos Group**.

igalia

Mesa

- Mesa es una librería **open-source** que implementa soporte de OpenGL y Vulkan (y otras APIs) para una gran variedad de hardware.
- Fue creada por Brian Paul en los años 90.
  - La versión 1.0 fue publicada en **Febrero de 1995**.
  - La versión actual es **21.1.3**.
    - Actualmente hay una release por trimestre: AÑO.YY

igalia


3D computer game
Geometry data
Texture data
Sound data
Game engine
OpenGL commands or shaders written in GLSL (vertex, tesselation control, tessellation evaluation, geometry, fragment and compute shaders)
OpenGL
Vulkan
system calls
subroutine calls
Windowing library (SDL, GLFW, etc.)
Subroutines
Mesa 3D
GNU C Library
Subroutines
DRM library
Subroutines
System Call Interface (SCI)
Process scheduler
Memory manager
Other Linux kernel subsystems
KMS
DRM
Linux kernel
Hardware
CPU
System RAM
Devices
Screen
Display controller
Graphics RAM
GPU

Funcionalidades

- Traducir llamadas a las APIs a instrucciones que serán ejecutadas por el HW.
- Configuración del HW para la ejecución de los shaders: configurar depth, stencil tests, blend, etc.
- Buscar soluciones para limitaciones del HW: soporte a tipos de 64 bits, falta de instrucciones en el ISA, etc.
- Indicar a la aplicación los límites del HW: número de recursos disponibles, formatos de imagen soportados...

igalia

Funcionalidades

- Mesa se encarga de compilar los shaders (**GLSL** o **SPIR-V**).
  - Hay un **frontend compiler** que produce como salida un lenguaje intermedio llamado **NIR**.
  - Cada uno de los drivers de Mesa, implementa un **backend compiler** para obtener el código **ensamblador** a partir de NIR.

igalia

GLSL

```
uniform vec4 args1, args2;

void main()
{
        gl_FragColor = log2(args1) + args2;
}

```

NIR

```
NIR (final form) for fragment shader:
shader: MESA_SHADER_FRAGMENT
name: GLSL3
inputs: 0
outputs: 0
uniforms: 32
shared: 0
decl_var uniform INTERP_MODE_NONE vec4 args1 (0, 0, 0)
decl_var uniform INTERP_MODE_NONE vec4 args2 (1, 16, 0)
decl_var shader_out INTERP_MODE_NONE vec4 gl_FragColor (FRAG_RESULT_COLOR, 4, 0)
decl_function main returning void

impl main {
        block block_0:
        /* preds: */
        vec1 32 ssa_0 = load_const (0x00000000 /* 0.000000 */)
        vec4 32 ssa_1 = intrinsic load_uniform (ssa_0) () (0, 16) /* base=0 */ /* range=16 */   /* args1 */
        vec1 32 ssa_2 = flog2 ssa_1.x
        vec1 32 ssa_3 = flog2 ssa_1.y
        vec1 32 ssa_4 = flog2 ssa_1.z
        vec1 32 ssa_5 = flog2 ssa_1.w
        vec4 32 ssa_6 = intrinsic load_uniform (ssa_0) () (16, 16) /* base=16 */ /* range=16 */ /* args2 */
        vec1 32 ssa_7 = fadd ssa_2, ssa_6.x
        vec1 32 ssa_8 = fadd ssa_3, ssa_6.y
        vec1 32 ssa_9 = fadd ssa_4, ssa_6.z
        vec1 32 ssa_10 = fadd ssa_5, ssa_6.w
        vec4 32 ssa_11 = vec4 ssa_7, ssa_8, ssa_9, ssa_10
        intrinsic store_output (ssa_11, ssa_0) () (4, 15, 0) /* base=4 */ /* wrmask=xyzw */ /* component=0 */   /* gl_FragColor */
        /* succs: block_0 */
        block block_0:
}

```

Ensamblador

```
Native code for unnamed fragment shader GLSL3
SIMD16 shader: 9 instructions. 0 loops. 54 cycles. 0:0 spills:fills. Promoted 0 constants. Compacted 144 to 80 bytes (44%)
   START B0 (54 cycles)
math log(16)    g3<1>F          g2<0,1,0>F      null<8,8,1>F    { align1 1H compacted };
math log(16)    g5<1>F          g2.1<0,1,0>F    null<8,8,1>F    { align1 1H compacted };
math log(16)    g7<1>F          g2.2<0,1,0>F    null<8,8,1>F    { align1 1H compacted };
math log(16)    g9<1>F          g2.3<0,1,0>F    null<8,8,1>F    { align1 1H compacted };
add(16)         g120<1>F        g3<8,8,1>F      g2.4<0,1,0>F    { align1 1H compacted };
add(16)         g122<1>F        g5<8,8,1>F      g2.5<0,1,0>F    { align1 1H compacted };
add(16)         g124<1>F        g7<8,8,1>F      g2.6<0,1,0>F    { align1 1H compacted };
add(16)         g126<1>F        g9<8,8,1>F      g2.7<0,1,0>F    { align1 1H compacted };
sendc(16)       null<1>UW       g120<8,8,1>F
                                render RT write SIMD16 LastRT Surface = 0 mlen 8 rlen 0 { align1 1H EOT };
   END B0
```

Kernel

- Mesa utiliza el API de **DRM** (Direct Rendering Management) para hablar con el HW a través del kernel.
  - Obtener información del dispositivo (GPU id, funcionalidades soportadas...)
  - Reserva de memoria de video.
  - Mapeo de memoria para leer/escribir desde user-space.

igalia

Drivers en Mesa

GPU	Driver OpenGL	Driver Vulkan
Intel	*i915*, **i965**, **Iris**, Crocus	**Anvil**
AMD	r200, r300, r600, **radeonsi**	RADV
Nvidia	nouveau, **tegra**	-
Broadcom	vc4, v3d	vd3v
Qualcomm Adreno	freedreno	Turnip
Arm Mali	lima, panfrost	panvk

igalia

Drivers en Mesa

GPU	Driver OpenGL	Driver Vulkan
Imagination Technologies	imx	-
Vivante	etnaviv	-
Apple M1	asahi	-

igalia

Layered Drivers

Driver	Convierte de	Convierte a
Zink	OpenGL	Vulkan
D3D12	OpenGL	Direct3D 12

Otros	Driver OpenGL	Driver Vulkan
Virtual Machines	**svga3d (vmware)**, virgl	venus

Otros drivers

- Emulación por software:
  - OpenGL: swrast (llvmpipe), OpenSWR, softpipe.
  - Vulkan: swrast (lavapipe).
- Nine: driver de gallium para Direct3D 9.
- Clover: driver de gallium para OpenCL.

igalia

¿Qué soporta cada driver? Mesa matrix

OpenGL

Leaderboard

There is a total of **160** extensions to implement. The ranking is based on the number of extensions done by driver.

#	Driver	Extensions	OpenGL
1	zink	(100.0%) 160	4.6
	mesa	(100.0%) 160	4.6
	llvmpipe	(100.0%) 160	4.6
	i965	(100.0%) 160	4.6
	radeonsi	(100.0%) 160	4.6
6	nvc0	(98.8%) 158	4.5
7	r600	(96.2%) 154	4.5
	virgl	(96.2%) 154	4.3
9	softpipe	(88.8%) 142	3.3
10	freedreno	(81.2%) 130	3.3
11	swr	(80.0%) 128	3.3
12	nv50	(74.4%) 119	3.3
13	panfrost	(66.9%) 107	3.1
14	d3d12	(61.9%) 99	3.3
15	v3d	(58.8%) 94	N/A
16	vc4	(36.9%) 59	N/A
17	lima	(35.0%) 56	N/A
18	etnaviv	(26.2%) 42	N/A

Vulkan

Leaderboard

There is a total of **48** extensions to implement. The ranking is based on the number of extensions done by driver.

#	Driver	Extensions	Vulkan
1	radv	(100.0%) 48	1.2
	vn	(100.0%) 48	1.2
	anv	(100.0%) 48	1.2
	mesa	(100.0%) 48	1.2
5	tu	(83.3%) 40	1.1
6	lvp	(79.2%) 38	1.1
7	v3dv	(47.9%) 23	1.0

Vulkan 1.2

		Intel	Software	AMD	Qualcomm	Broadcom	Emulation
Extension	mesa	anv	lvp	radv	tu	v3dv	vn
	100.0%	100.0%	58.3%	100.0%	66.7%	12.5%	100.0%
VK_KHR_8bit_storage	2020-11-02	2020-11-02	2021-03-22	2020-11-02			2021-04-23
VK_KHR_buffer_device_address	2020-11-02	2020-11-02	2021-03-16	2020-11-02			2021-04-23
VK_KHR_create_renderpass2	2020-11-02	2020-11-02	2021-03-14	2020-11-02	2021-03-11		2021-04-23
VK_KHR_depth_stencil_resolve	2020-11-02	2020-11-02		2020-11-02	2021-03-11		2021-04-23
VK_KHR_draw_indirect_count	2020-11-02	2020-11-02	2021-03-11	2020-11-02	2021-03-11		2021-04-23
VK_KHR_driver_properties	2020-11-02	2020-11-02	2021-03-11	2020-11-02			2021-04-23
VK_KHR_image_format_list	2020-11-02	2020-11-02	2021-04-04	2020-11-02	2021-03-11	2021-06-10	2021-04-23
VK_KHR_imageless_framebuffer	2020-11-02	2020-11-02	2021-04-05	2020-11-02			2021-04-23
VK_KHR_sampler_mirror_clamp_to_edge	2020-11-02	2020-11-02	2021-03-11	2020-11-02	2021-03-11	2021-06-10	2021-04-23
VK_KHR_separate_depth_stencil_layouts	2020-11-02	2020-11-02		2020-11-02			2021-04-23
VK_KHR_shader_atomic_int64	2020-11-02	2020-11-02	2021-03-29	2020-11-02			2021-04-23
VK_KHR_shader_float16_int8	2020-11-02	2020-11-02		2020-11-02	2021-04-05		2021-04-23
VK_KHR_shader_float_controls	2020-11-02	2020-11-02		2020-11-02	2021-04-05		2021-04-23
VK_KHR_shader_subgroup_extended_types	2020-11-02	2020-11-02		2020-11-02			2021-04-23
VK_KHR_spirv_1_4	2020-11-02	2020-11-02		2020-11-02	2021-04-14		2021-04-23

Comunidad

- La comunidad está formada por desarrolladores voluntarios y múltiples empresas.

igalia

¿Cómo nos organizamos?

- Lista de correo para desarrolladores.
  - https://lists.freedesktop.org/mailman/listinfo/mesa-dev
- Canales de IRC (OFTC)
  - #dri-devel, #intel-3d, #radeon, #videocore, #freedreno...

igalia

¿Cómo nos organizamos?

- Gitlab (issue tracker, revisión de código, integración continua).
  - **https://gitlab.freedesktop.org/mesa/**

igalia

Conferencias

- **X.Org Developers Conference**.
  - **Edición de 2021**: 15-17 Septiembre, Virtual.
- **FOSDEM**.

Cómo contribuir

- Testing!
  - Instala la última versión de Mesa.
    - Compila el **código fuente** (**instrucciones**).
    - Si usas Ubuntu, puedes añadir el **PPA de Oibaf**.
    - Usa una distro que proporcione nightly builds de Mesa o, al menos, la última versión estable.

igalia

Cómo contribuir

- Prueba con:
  - Testing suites como: **VK-GL-CTS**, **piglit**, **crucible**.
  - Videojuegos u otras aplicaciones que usen OpenGL o Vulkan.
- Si encuentras un fallo...
  - ¡**Repórtalo**! (**instrucciones**).
  - ¡Envía un **parche** arreglándolo! (**instrucciones**).

igalia

Más información

- Website
  - **https://mesa3d.org**
- Twitter
  - **X.Org Foundation**
  - **X.Org Developers Conference**
- Canal Youtube

https://www.youtube.com/channel/UCXlH5v1PkEhjzLFTUTm_U7g

Q&A

¡Estamos contratando!