Las librerías dinámicas son una mierda

Las ~~bibliotecas~~ librerías
dinámicas
son una mierda

$ whoami

Pablo Marcos Oltra

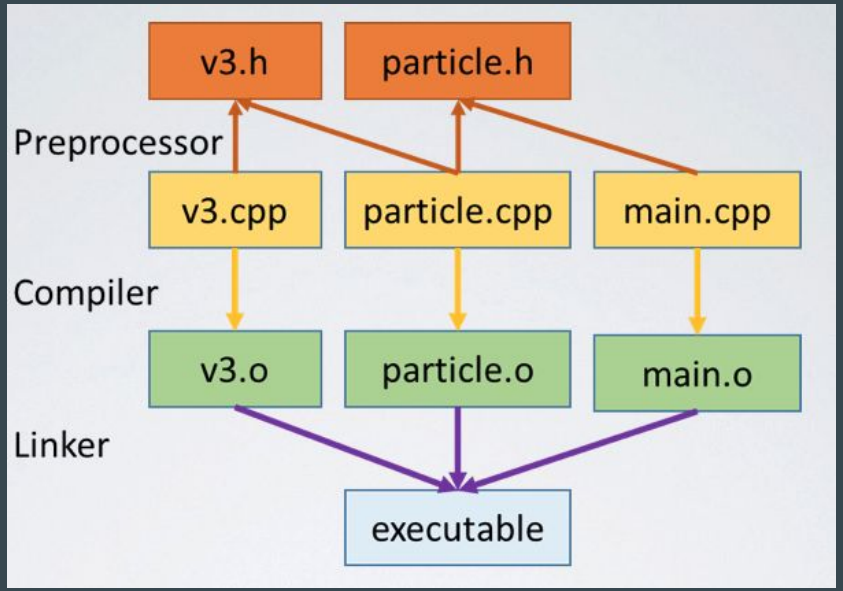
@pablomarc0s - Twitter

@pamarcos - Slack

https://github.com/pamarcos/peumconf_2019

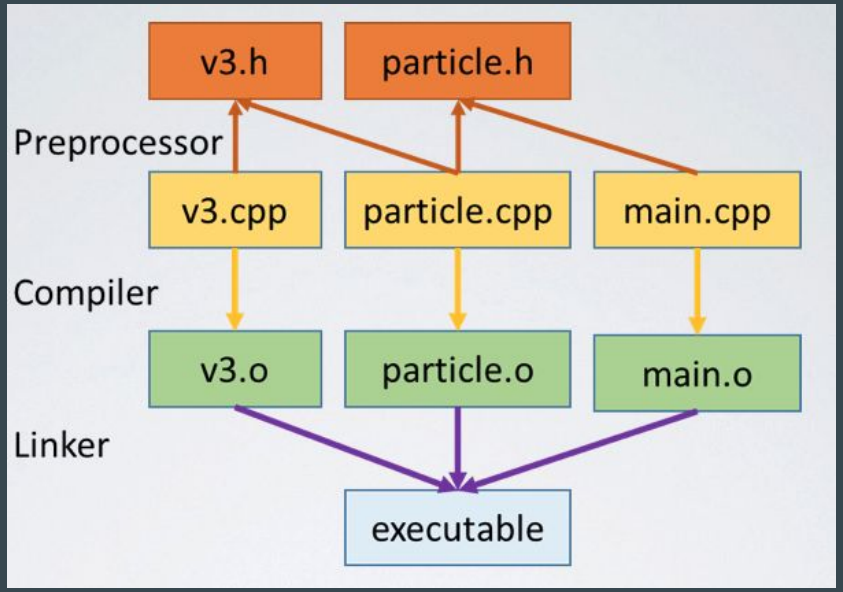

Simple Poll APP 7:06 PM
Propuestas para charlas que podría dar para la PEUM Conf 2019
1 Memoria y allocators 2
@Tx, @nico h
2 Librerías dinámicas y cómo hacer perrerías con ellas -> rollo inyección de librerías en procesos, hijackear y demás 3
@Tx, @Dynam1co, @oliver
3 Cómo funciona la compresión de vídeo 2
@mmanzano, @Ibon K
4 Estamos en la mierda -> disertación sobre software moderno en plan abuelo cebolleta, rollo Jonathan Blow pero en bien 12
@Alberto, @favio, @infogon, @Rober, @mmanzano, @Dani PR, @Vic Ptmk, @Javieraeros, @Lau, @PabloJS, @oliver, @Ibon K

Compilación 101




```
g++ -o v3.o -c v3.cpp
g++ -o particle.o -c particle.cpp
g++ -o main.o -c main.cpp
g++ -o executable v3.o particle.o main.o
```

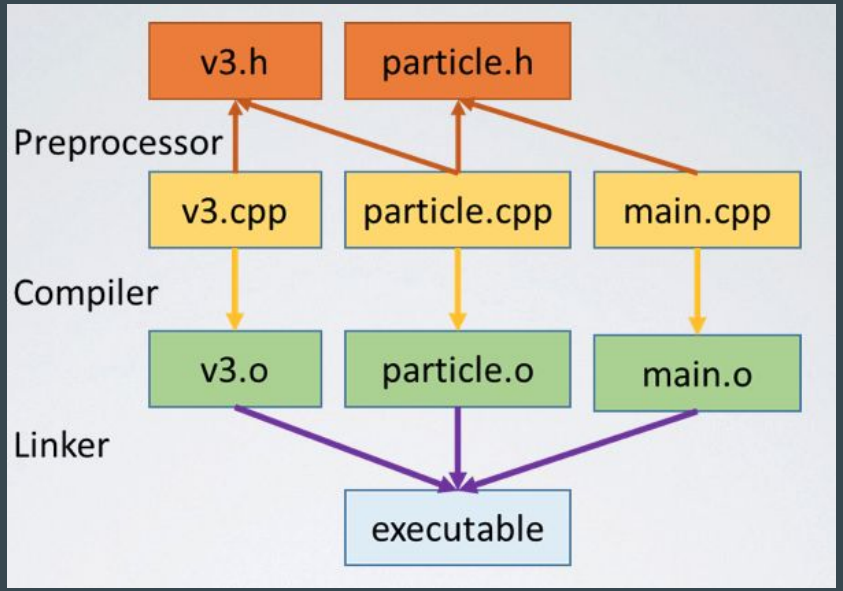
Compilación 101



Compilación

```
g++ -o v3.o -c v3.cpp
g++ -o particle.o -c particle.cpp
g++ -o main.o -c main.cpp
g++ -o executable v3.o particle.o main.o
```

Compilación 101




```
g++ -o v3.o -c v3.cpp
g++ -o particle.o -c particle.cpp
g++ -o main.o -c main.cpp
```

Enlazado

```
g++ -o executable v3.o particle.o main.o
```

Símbolos

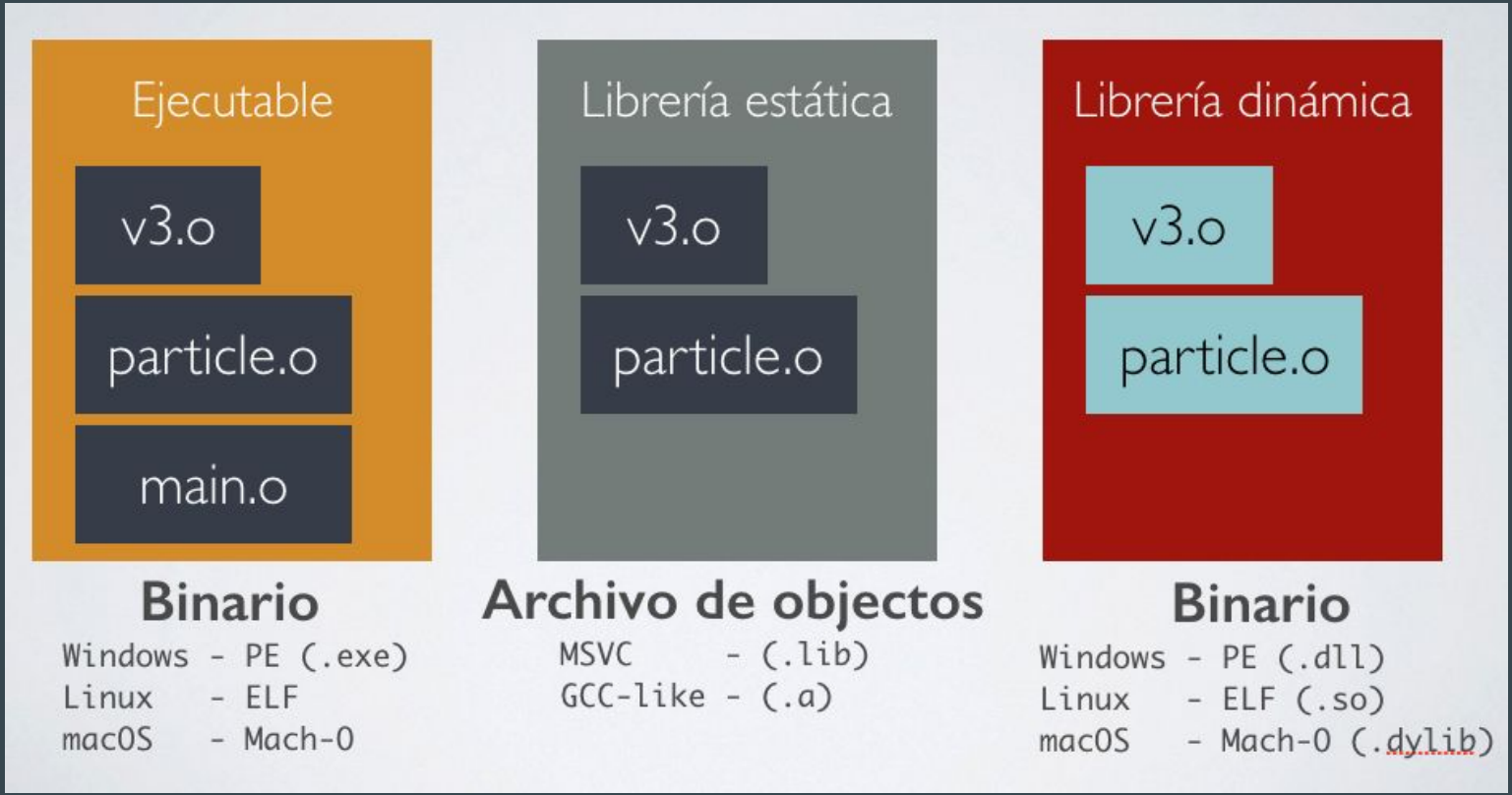
File: **foo.c**

```c
#include <stdio.h>

void foo()
{
    printf("Original foo\n");
}
```

```
✘ fluendo@MacBook-Pro-de-fluendo ~/Repos/peumconf_2019/demo_rpath  master nm libfoo.dylib
0000000000000f60 T _foo
                 U _printf
                 U dyld_stub_binder
```

Empaquetando objectos



Librerías estáticas vs dinámicas

El dynamic linker

Usando librerías dinámicas

En tiempo de compilación

```
g++ -o executable main.o -lsharedLib.so
```

En tiempo de ejecución

- Windows: `LoadLibrary y GetProcAddress`
- Unix: `dlopen y dlsym`

Demo 1

Demo 1:
~~Azogue~~
Asogue

https://github.com/pamarcos/Urho3D/tree/RCCpp

¿Dónde se buscan las librerías?

Unix - Linux y macOS

1. Directorios del RPATH
2. LD_LIBRARY_PATH (Linux) y DYLD_LIBRARY_PATH (macOS)
3. Directorios del system search path: /etc/ld.so.conf y /lib (Linux), /usr/lib, /usr/local/lib, etc

Windows

1. Directorio del ejecutable
2. Directorio del sistema: C:\Windows\System32
3. Directorio de Windows: C:\Windows
4. El directorio actual: CWD
5. Directorios listados en la variable de entorno PATH

Inyectando una librería

Linux

`LD_PRELOAD`

macOS

`DYLD_INSERT_LIBRARIES`

Demo 2

Demo 2:
~~Alcacil~~
Arcasil

https://github.com/pamarcos/peumconf_2019/demo_injection

Demo 3

Demo 3: Ajíbiri

https://github.com/pamarcos/peumconf_2019/demo_rpath

Utilidades

Linux

```
nm, ldd, strace, ltrace, LD_DEBUG
```

macOS

```
nm, otool, DYLD_PRINT_LIBRARIES, DYLD_*
```

Windows

```
Dependency walker, Dependencies (versión moderna),
Process Explorer
```

Preguntas?