
Título: Desarrollo de un kernel académico para arquitecturas x86-64 en C++
Autor: Ernesto Martínez García <me@ecomaikgolf.com>
Tutor: Antonio Miguel Corbi Bellot <acorbi@dlsi.ua.es>
Titulación: Grado en Ingeniería Informática
Curso Académico: 2021/2022 (C3)

<https://github.com/ecomaikgolf/alma>

El proyecto trata de desarrollar y documentar “*alma*”, un kernel académico escrito en C++ para arquitecturas x86-64.

alma es una pieza de software capaz de ejecutarse de manera independiente —sin sistema operativo— en un ordenador y proveer funcionalidades básicas al usuario como las que podemos encontrar en Linux, Windows, Unix, etc. *alma* no es un núcleo útil ni profesional, pero implementa las funcionalidades clave que sustentan los sistemas operativos modernos —el alma de los sistemas operativos, de ahí el nombre—.

En sus últimas versiones, *alma* es un núcleo que puedes construir en un archivo *iso*, cargarlo en un USB al igual que se hace con distribuciones Linux o Windows, y arrancarlo desde la BIOS/UEFI en un ordenador que cumpla con los requisitos. Cada píxel que aparece en pantalla, cada pistón del teclado, cada reserva de memoria, todo está gestionado exclusivamente por *alma* y plasmado en este trabajo.

El objetivo del proyecto no es desarrollar un kernel aplicable en casos reales ni útil para determinadas tareas. *alma* ha sido desarrollado con fines meramente académicos, al igual que otros proyectos similares desarrollados por otras universidades tales como *xv6* (*Massachusetts Institute of Technology*), *OS/161* (*Harvard*) y *SWEB* (*Graz University of Technology*).

Junto a *alma* también se ha desarrollado un *bootloader* escrito en C para arranques UEFI, empleado en las primeras versiones de *alma*. En las últimas versiones, *alma* sigue una especificación estandarizada para el arranque, por lo que puede ser iniciado con multitud de *bootloaders* —se ha elegido *limine* como *bootloader* por defecto—.

La construcción de la *toolchain* necesaria para trabajar con el proyecto se ha automatizado mediante un *makefile*, ya que es una tarea complicada, larga y tediosa. Construir *alma* requiere de una serie de modificaciones durante la construcción del compilador (`gcc`): *target* genérico, desactivar la *red-zone* y compilar *libgcc* con `mmodel`.

Para construir el proyecto se ha desarrollado un sistema de construcción con `cmake` multinivel, capaz de construir *alma*, el *bootloader* y generar el *iso* de forma automática en tan solo 0,41 segundos con `ninja` o `make`. También se ha compuesto un entorno virtualizado de desarrollo —la “*alma build vm*”— con todas las dependencias del proyecto instaladas, capaz de construir y ejecutar el proyecto de forma gráfica, portable y controlada.

Todo el desarrollo del proyecto se ha llevado a cabo en Github, manteniendo *mirrors* en Gitlab y en un servidor *git* propio para garantizar la seguridad del proyecto. A fecha de entrega el repositorio tiene 446 *commits*, albergando 12,831 líneas de código escritas.