



## **Arquitectura de Computadoras - 72.08**

Trabajo Práctico Especial

Segundo Cuatrimestre 2025

Grupo 10:

Felipe Ignacio Boyaciyan - 64026

Maximo Agustin Ramos - 64082

Victoria Helena Park - 64498

## **Introducción**

En este Trabajo Práctico Especial de la materia Arquitectura de Computadoras (72.08) se ha implementado un kernel booteable mediante Pure64, diseñado para la arquitectura Intel x86-64 en modo largo (Long Mode).

El sistema cumple las funciones básicas de un sistema operativo moderno: administrar los recursos de hardware y proveer una API que permita a las aplicaciones de usuario (userland) interactuar con dichos recursos de forma controlada y eficiente.

El kernel gestiona interrupciones, llamadas al sistema, temporizadores, video, teclado y ofrece servicios esenciales como escritura en pantalla, lectura de entrada y control básico del flujo de ejecución.

De esta manera, actúa como intermediario entre el hardware y los programas de usuario, brindando una capa de abstracción segura para el desarrollo y prueba de software en bajo nivel.

En el espacio de userland, se desarrolló un intérprete de comandos, que ofrece distintas funcionalidades, entre ellas, el juego tron, con distintos modos de juego y niveles.

## Estructura del proyecto

El proyecto está compuesto por distintos componentes, siendo los principales Kernel y Userland.

- **Kernel**

Implementaciones:

- Manejos de interrupciones (de hardware y software)
- Manejo de excepciones.
- Drivers de video, teclado y audio.

- **Userland**

Implementaciones:

- Intérprete de comandos
- Tron

- **Bootloader**

- **Toolchain**

- **Image**

## Entorno y arranque

El sistema se carga mediante Pure64, un bootloader que inicializa el procesador en modo largo (64 bits) y transfiere el control al kernel.

Desde ese punto, el kernel:

1. Configura la IDT (Interrupt Descriptor Table).
2. Habilita las interrupciones y los controladores (teclado, timer, video).
3. Inicializa los manejadores de excepciones y llamadas al sistema (INT 0x80).
4. Lanza el proceso del intérprete de comandos (shell) en modo usuario.

Luego, el usuario se comunica con el kernel a través de llamadas al sistema (*syscalls*) los cuales interactúan directamente con los drivers de teclado, video y audio.

## Intérprete de comandos - Manual de usuario

El intérprete de comandos permite al usuario interactuar con el sistema mediante instrucciones escritas en la consola.

Cada comando ejecuta una rutina en modo usuario que realiza una o varias llamadas al sistema (syscalls) para comunicarse con el kernel y acceder al hardware.

Al iniciar el sistema, la shell le da la bienvenida al usuario, indicando que haga uso del comando 'help' para ver los comandos.

Los comandos disponibles son:

- **help**: lista todos los comandos disponibles junto a su descripción.
- **clear**: limpia la pantalla
- **registers**: permite al usuario ver el estado de los registros al presionar la tecla indicada. Imprime un mensaje de error en caso de no haberla presionado anteriormente.
- **time**: imprime la hora y fecha en pantalla
- **zoomIn**: permite agrandar el tamaño de la letra
- **zoomOut**: permite achicar el tamaño de letra, con un límite inferior.
- **tron**: permite al usuario jugar
- **beep**: emite un sonido
- **benchmark**:
- **divZero**: fuerza una excepción de *divide by zero*
- **invOpCode**: fuerza una excepción de *invalid code operation*

# Benchmark

```
QEMU
Welcome! Run the command 'help' to see all the commands
$> benchmark

=====
                BENCHMARK SUITE
=====
System information:
CPU Speed: 1225 MHz
Active cores: 1
Detected cores: 1

Running benchmarks...
(Each test runs 100000000 iterations)

Arithmetic Operations:
Time: 1.1s
Ticks: 20
Ops/sec: ~360000000

Memory Access:
Time: 0.6s
Ticks: 11
Ops/sec: ~163636363

Comparison Operations:
Time: 1.2s
Ticks: 23
Ops/sec: ~313043478

Function Calls:
Time: 1.2s
Ticks: 22
Ops/sec: ~81818181

[Floating point test skipped - not implemented]
=====
                Benchmarks completed!
=====
```

## **Aspectos técnicos**

El desarrollo alcanza un sistema operativo elemental que cumple su función, aunque todavía exhibe carencias y áreas susceptibles de optimización:

Por el momento, carece de capacidad para ejecutar múltiples tareas simultáneamente, limitándose a correr una aplicación por vez (ya sea la terminal de comandos o algún programa de entretenimiento). Una evolución deseable consistiría en diseñar un scheduler que distribuya los recursos del procesador entre diferentes aplicaciones, incorporando además las estructuras necesarias para administrarlos (bloques de control de procesos) y optimizar la gestión de memoria mediante técnicas como paginación o división por segmentos. El intercambio de contexto entre procesos resultaría fundamental para concretar esta funcionalidad.

Por otro lado, la compatibilidad con dispositivos externos abarca únicamente entrada por teclado, salida de video y el altavoz integrado. Una extensión factible implicaría agregar drivers para dispositivos adicionales como ratones con interfaz PS/2 o bus serie universal, incluyendo el tratamiento adecuado de sus interrupciones y la visualización gráfica del puntero en pantalla.

En cuanto al renderizado de video, existe una limitación visible al momento de actualizar la pantalla completa: la imagen no se refresca de forma instantánea, sino que se aprecia un efecto de barrido progresivo durante el redibujado. Esta característica podría mejorarse implementando técnicas de doble buffer o sincronización vertical para lograr transiciones más fluidas y eliminar el parpadeo perceptible.