



Emulador de microprocesador Leon3 para desarrollo de software satelital y simuladores

Especificación de casos de uso

Autor:

Ing. Iriarte Fernandez, Nicolás Ezequiel (NicolasIriarte95@gmail.com)

Director:

Esp. Lic. Horro, Nicolás Eduardo (INVAP.S.E.)

Docentes:

Alejandro Permingeat; Esteban Volentini; Mariano Finochietto y
Santiago Salamandri.

*Este documento fue realizado en el curso Ingeniería de Software
el 16 de noviembre de 2023, tercer bimestre.*

Índice

Arquitectura del sistema	4
1.1 FileSystem	4
1.2 MemoryModel	4
1.3 Registers	5
1.4 TimeKeeper	5
1.5 Scheduler	5
1.6 CPU	5
1.7 C-API	6

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0	Creación del documento	16 de noviembre de 2023

Arquitectura del sistema

El presente documento describe la arquitectura del sistema. El sistema, visto desde un alto nivel, puede dividirse en siete componentes, tal como se muestra en la figura 1. Cada uno de ellos tiene un comportamiento bien definido.

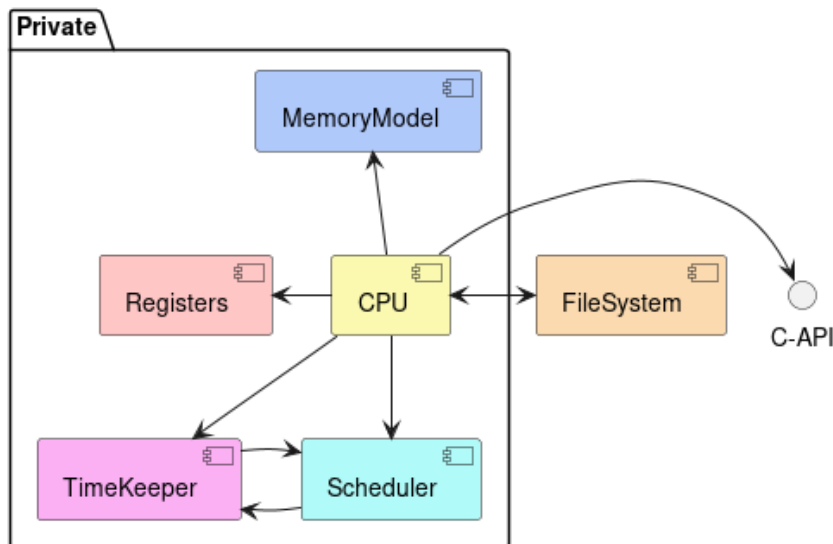


Figura 1. Diagrama de componentes del sistema.

A continuación se detalla cada uno de los componentes del sistema y su responsabilidad.

1.1. FileSystem

El componente `FileSystem` es el encargado de interactuar con el sistema de archivos de la computadora. Será quien permita leer y escribir archivos en disco. Entre sus responsabilidades se encuentran:

- Leer el binario a ejecutar.
- Leer y escribir archivos de configuración.
- Leer y escribir archivos de log.

1.2. MemoryModel

El componente `MemoryModel` es el encargado de modelar la memoria del sistema. En dicha memoria se almacenará el binario a ejecutar, así como también los datos que se deseen almacenar.

1.3. Registers

El componente **Registers** es el encargado de modelar los registros del microprocesador. Será modelado como una estructura que contiene los registros del microprocesador como variables del tipo correspondiente (*uint32_t*, *uint64_t*, *uint8_t*, etc), su tipo dependerá del registro que represente.

1.4. TimeKeeper

El componente **TimeKeeper** es el encargado de modelar el tiempo del sistema. Será quien permita verificar si el tiempo de procesamiento está demorando más de lo esperado, y controlar otros escenarios a futuro en caso de ser necesario, tal como podrían ser emulaciones a mayor velocidad que el tiempo real.

1.5. Scheduler

El componente **Scheduler** es el encargado de orquestar y ejecutar todas las instrucciones del microprocesador. Es el encargado de buscar la próxima instrucción a ejecutar, ejecutarla, y actualizar los registros y la memoria según corresponda. Dicho componente trabajará en conjunto con el componente **TimeKeeper** para verificar que el tiempo de ejecución no se exceda del tiempo real. El **Scheduler** será implementada como una máquina de estados finitos, tentativamente como se muestra en la figura 2.

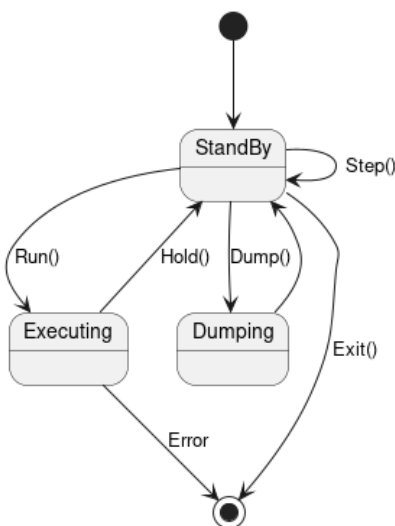


Figura 2. Diagrama de estados tentativo del Scheduler.

1.6. CPU

El componente **CPU** es el encargado de modelar el microprocesador. Será quien interactúe con el resto de componentes, siendo el que contiene la lógica de la emulación de cada una de las instrucciones del microprocesador. Entre sus responsabilidades se encuentran:

- Inicializar los componentes `MemoryModel`, `Registers`, `TimeKeeper` y `Scheduler`.
- Cargar el binario a ejecutar en la memoria.
- Controlar las pausas y reinicios de la emulación.

1.7. C-API

El componente `C-API` es el encargado de exponer una API para que el usuario pueda interactuar con el sistema. Dicha API será expuesta mediante una librería dinámica, que podrá ser utilizada desde el language de programación `C` y `C++`. Todas las interacciones con el sistema se realizarán a través de esta API.