



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

Introducción a los casos de uso	4
---	---

Registros de cambios

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

Introducción a los casos de uso

El presente documento describe los casos de uso del emulador de microprocesador Leon3 para desarrollo de software satelital y simuladores. Se presentan tres casos de uso representativos al uso nominal del software propuesto:

1. Cargar software de vuelo.
2. Inspeccionar registros y memorias.
3. Validar programa introducido al emulador.

Cuadro 1. Caso de Uso: Cargar software de vuelo [NEMU-UC-01]

Título	Descripción
1. Nombre 1.1 código 1.2 Breve descripción 1.3 Actor principal 1.4 Disparadores	Cargar software de vuelo [NEMU-UC-01] Capacidad de carga de software de vuelo arbitrario. Desarrolladores de FSW y operadores. Llamada a funcion por API.
2. Flujo de eventos 2.1 Flujo básico 2.2 Flujo alternativo	<ol style="list-style-type: none">1. Se inicia una instancia de emulación.2. Se llama a la funcion de API para la carga de binarios.3. El software dado es cargado en la memoria emulada y se actualizan los registros correspondientes (Como contadores de programa).4. La función retorna al usuario. <ul style="list-style-type: none">■ Ante un binario inválido, como podría ser por compilador inválido o arquitectura incorrecta, la función retornará un código de error.■ Ante un archivo no existente, la función retornará un código de error.
3. Requerimientos especiales	N/A.
4. Pre-Condiciones	Se posee un binario para ejecutar dentro del emulador.
5. Post-Condiciones	Contador de programa actualizado y programa cargado en memoria.

Cuadro 2. Caso de Uso: Inspeccionar registros y memorias [NEMU-UC-02]

Título	Descripción
1. Nombre	Inspeccionar registros y memorias
1.1 código	[NEMU-UC-02]
1.2 Breve descripción	Acceso a los registros y memorias emuladas, tanto para verificación de funcionamiento como para desarrollo de drivers.
1.3 Actor principal	Desarrolladores de FSW y Desarrolladores de modelos.
1.4 Disparadores	Llamada a funcion por API.
2. Flujo de eventos	
2.1 Flujo básico	<ol style="list-style-type: none">1. Se inicia una instancia de emulación.2. Se llama a la funcion de API para la carga de binarios.3. Se llama a la funcion de API para volcar los registros a archivo CSV.4. Alternativamente se puede llamar a la funcion de volcar memoria, especificando la posición initial, el tamaño a volcar y el archivo de salida para su analisis.
2.2 Flujo alternativo	<ul style="list-style-type: none">■ Ante una posición de memoria o memoria invalida la función retornará un código de error.
3. Requerimientos especiales	N/A.
4. Pre-Condiciones	Se posee un binario para ejecutar dentro del emulador.
5. Post-Condiciones	Se tendrá una instancia del emulador con un binario cargado y un archivo en formato CSV con los datos solicitados.

Cuadro 3. Caso de Uso: Validar programa introducido al emulador [NEMU-UC-03]

Título	Descripción
1. Nombre	Validar programa introducido al emulador
1.1 código	[NEMU-UC-03]
1.2 Breve descripción	El emulador permitirá ejecutar y validar binarios producidos para el microprocesador.
1.3 Actor principal	Desarrolladores de FSW y Desarrolladores de modelos.
1.4 Disparadores	Llamada a funcion por API.
2. Flujo de eventos	
2.1 Flujo básico	<ol style="list-style-type: none"> 1. Se inicia una instancia de emulación. 2. Se llama a la funcion de API para la carga de binarios. 3. Se analiza si el binario dado realiza las operaciones esperadas.
2.2 Flujo alternativo	N/A.
3. Requerimientos especiales	N/A.
4. Pre-Condiciones	Se posee un binario para ejecutar dentro del emulador con un comportamiento conocido.
5. Post-Condiciones	Se tendrá una instancia del emulador con un binario cargado.