

Evaluación Globalizadora B

Sistemas Operativos de Propósito General

Técnicas Digitales III
Departamento de Ingeniería Electrónica
UTN FRA

October 24, 2025

Fecha de realización: Lunes 20 de Octubre (512)
 Miércoles 22 de Octubre (511)
Fecha de Entrega: Lunes 17 de Noviembre (512)
 Miércoles 19 de Noviembre (511)

Estudiantes

APELLIDO Y NOMBRE:

APELLIDO Y NOMBRE:

Docentes:

Prof. Adj. Ing. MARÍN Ariel
Auxiliar: CARLASSARA Fabrizio
Auxiliar: VAN ROEY Julián

Abstract

Se desarrollará una Evaluación Globalizadora B con el fin de que el estudiante pueda aplicar los conocimientos teórico-prácticos asociados a los Sistemas Operativos de Propósito General (GPOS), desarrollando un driver que permita comunicar un sistema embebido corriendo este sistema operativo con otro corriendo un Sistema Operativo en Tiempo Real (RTOS), desarrollado en el primer cuatrimestre como Evaluación Globalizadora A. Se debe plantear una forma en la que ambos sistemas interactúen y comuniquen datos.



1 Objetivos

Los estudiantes deberán demostrar conocimientos asociados a:

- El desarrollo de un driver para un periférico dentro de un GPOS.
- El uso de device trees en un GPOS.
- La comunicación de dos sistemas operativos con principios de funcionamiento diferentes.
- Coordinar programas que corran a nivel de kernel y usuario en un GPOS.

2 Requisitos

Para el desarrollo de esta Evaluación Globalizadora B será necesario contar con:

- Un entorno de desarrollo para Linux con el kernel en su versión 6.6.28 según las instrucciones de la cátedra.
- Un cliente de SSH.
- Una Raspberry Pi con el kernel en su versión 6.12.25 o superior.
- Evaluación Globalizadora A con informe, aprobada.

3 Condiciones de Aprobación de la EGB

3.1 Aprobación

Para aprobar la Evaluación Globalizadora B es necesario desarrollar en su totalidad el Parte Teórica: Modo Protegido y la Consigna práctica.

3.2 Promoción

La promoción quedará sujeta a cómo se haya desarrollado la aplicación que corre en el GPOS, la cantidad de variables que se monitoreen o controlen y la implementación que se haya hecho. Se valorarán aplicaciones que tengan interfaces gráficas o web frente a interfaces de consola.

Además, quedará condicionada la promoción a la defensa de la EGB que haga cada integrante del grupo de forma individual.

Será condición de Promoción Directa presentar la EGB en el Congreso de Ingeniería Electrónica a desarrollarse el día 13 de Noviembre del corriente año; para la que se desarrollará una presentación en la que se deberá incluir: un resumen de la relación entre la EGA y EGB, la metodología de evaluación, diagramas en bloque detallados y de software, como así también el diagrama de tareas actualizado.

3.3 Entrega

Se deberá entregar en el repositorio personal forkeado de 2025_td3.

Para hacerlo, dentro del directorio de *2_egb/1_informes*, crear un archivo donde se encuentre la resolución del cuestionario en formato PDF.

La resolución práctica debe hacerse dentro de los directorios asignados. Hay uno para dejar el workspace de la aplicación con RTOS y otro para la aplicación en GPOS.

La evaluación requerirá una entrega de un informe que irá en el directorio correspondiente según los requisitos de la siguiente sección.

Cuando se quiera hacer la entrega, crear una nueva rama local con el nombre *egb/v1* y hacer un pull request a la rama apropiada del repositorio *2025_td3* de la cátedra.

3.4 Informe

Se deberá confeccionar un informe en formato IEEE con los siguientes ítems:

- Índice
- Abstract con palabras clave
- Fundamentación
- Objetivos
- Alcance
- Diagrama de software
- Diagrama de tareas de EGA
- Diagrama de conexiones completo
- Listado de comandos implementados
- Conclusiones
- Referencias

En el caso del diagrama de software, debe plasmarse gráficamente cómo interactúan las distintas piezas de software: aplicación de usuario, kernel, RTOS.

Respecto del diagrama de tareas de la EGA, debe ser uno nuevo que agregue los ajustes que se debieron hacer respecto del planteo original de la EGA y lo que se debió hacer para integrar la EGB. Este diagrama es solo del RTOS.

4 Parte Teórica: Modo Protegido

Realizar un informe desarrollando cada uno de los temas indicados a continuación:

1) Mecanismos de Protección de acceso al Segmento en Modo Protegido:

- a. Realice un diagrama completo que permita comprender el mecanismo de protección de memoria del 80x386 que permite un no acceso directo al segmento. Explique detalladamente como interpreta un valor en un registro de segmento si TI=1 o TI=0.
- b. ¿Cuáles son las condiciones que deben cumplirse para que el selector correspondiente pueda acceder al segmento? ¿Cuál es la relación entre RPL y DPL?
- c. Explique que son y para qué sirven los siguientes registros y bits: GDTR, LDTR, bit G, bit D, bit P, bit S, bit E, bit A.
- d. Práctica: Ejercicio de MP (Interpretación y funcionamiento) Consigna: Realice un diagrama interpretando en Modo Protegido el siguiente selector: CS=A3FF.

2) Descriptores de Segmento y Descriptores de Sistema:

- a. ¿Cuántas tablas de descriptores existen en la arquitectura 80x386? Explique cada una de ellas. NOTA: No es lo mismo “tablas de descriptores” que “descriptores”.
- b. Explique de qué forma actúa el bit S como atributo de un descriptor. Utilice esta información para decir en qué tipo de descriptor se convierte el mismo. ¿Qué “tipos” de estos descriptores existen?
- c. Explique el concepto y funcionamiento de un “Call via Gate”.

3) Segmento de Estado de la Tarea (TSS):

- a. ¿Qué se busca con el uso del TSS relacionado al uso del procesador y la conmutación de tareas?
- b. Explique el funcionamiento y realice un diagrama de interpretación del mismo. Esquema del TSS.

4) Paginado:

- a. Explique conceptualmente para qué sirve el proceso de paginado y como se relaciona con el sistema de segmentado.
- b. ¿Qué hace la unidad de paginado? ¿Cómo lo hace?

5 Consigna práctica

En función de la consigna asignada en la EGA, proponer uno o varios datos que se puedan monitorear o controlar en el RTOS desde el GPOS. Ejemplos:

- Para un control de temperatura, podrían asignarse los setpoints desde el GPOS o pedir la temperatura desde el mismo.
- En una medición de nivel, podrían pedirse los últimos valores guardados o la medición actual.
- En un control de acceso podrían solicitarse cuántas veces un usuario falló en ingresar una clave o cambiar la misma.

La propuesta debe ser presentada formalmente una semana después de que se de esta consigna. La propuesta queda en revisión por los docentes y sujeta a sugerencias y cambios. Una vez aprobada, se puede proceder a desarrollar la EGB.

5.1 RTOS

La parte de la EGB que se desarrolla en RTOS debe funcionar igual que en la EGA, con el agregado de que debe encontrarse la forma de asignar una tarea adicional a que use el UART de la Raspberry Pi Pico para poder pedir/recibir los datos por parte de la aplicación con GPOS.

Dependiendo de la propuesta que se haya hecho para la EGB, puede que se necesiten hacer algunos cambios adicionales en la dinámica previa de las tareas o utilizar algunos recursos adicionales.

El firmware de la EGA debe poder recibir comandos desde la EGB que estén estandarizados y documentados (estilo get/set). En todos los casos, la EGA debe retornar hacia la EGB un código de error/estado del comando además de lo que solicite la EGB.

5.2 GPOS

La parte de la EGB que se desarrolla en GPOS tendrá dos partes: un programa que correrá en kernel space y otra que correrá en user space.

5.2.1 Kernel Space

Esta parte de la aplicación es la que debe trabajar directamente con el hardware. Se debe desarrollar un módulo del kernel que pueda hacer uso del UART de la Raspberry Pi con el device tree proporcionado.

Este módulo del kernel tendrá que luego ser cargado en el kernel y tener la habilidad de leer/escribir un char device que manipule el UART según sea necesario en la propuesta.

5.2.2 User Space

En esta sección, se debe desarrollar una pequeña interfaz que correrá sin privilegios. Esta aplicación deberá interactuar con el módulo del kernel desarrollado para poder enviar y recibir datos de la EGA.

Se solicitará que haya dos tipos de aplicaciones de usuario. La primera, deberá consistir en una aplicación de terminal sencilla. Esta debe únicamente permitir correr comandos de escritura/lectura que se hayan establecido y documentado.

La segunda aplicación debe hacer uso de recursos más gráficos para poder solicitar y controlar los mismos datos que la aplicación anterior. En cuanto al lenguaje o framework para desarrollar la aplicación, queda a criterio de cada grupo mientras se apruebe previamente con los docentes de la cátedra. Esta puede desarrollarse para correr localmente en la Raspberry Pi y conectarse remotamente desde otro dispositivo o correr en otro dispositivo y enviar comandos por SSH.

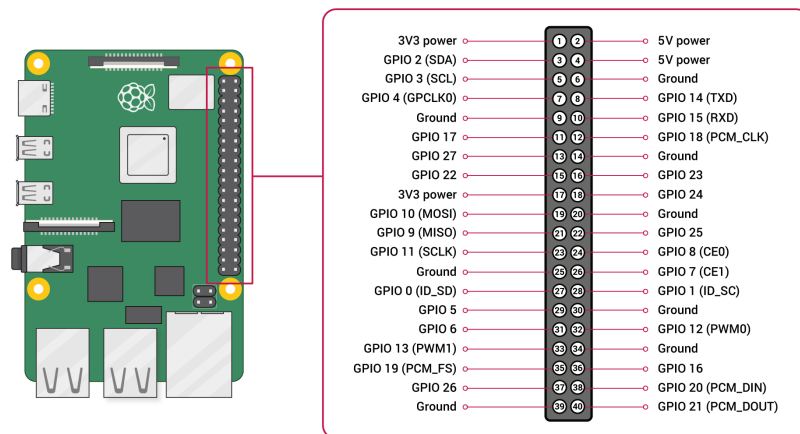


Figure 1: Pinout de Raspberry Pi 4

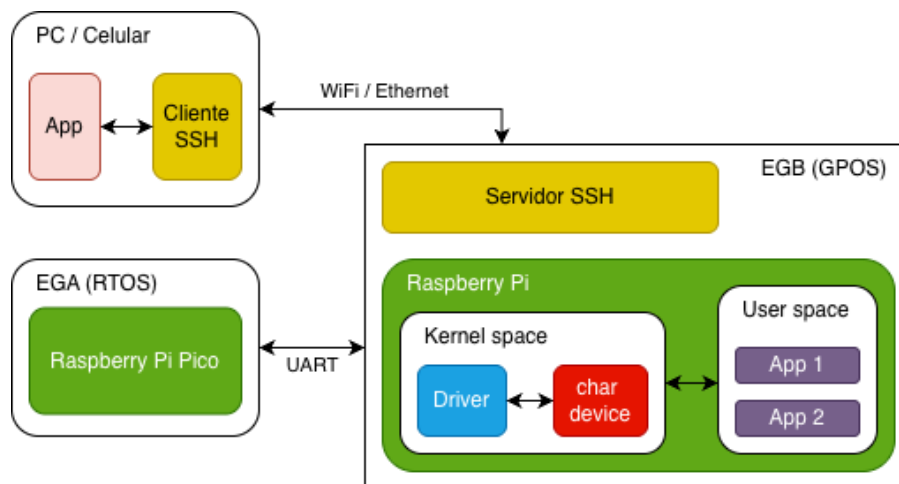


Figure 2: Diagrama conceptual de EGB