



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

***Departamento de
Ingeniería Electrónica***

Técnicas Digitales III

Guía de Trabajos Prácticos

Primer cuatrimestre

TABLA DE CONTENIDO

1. ABI y Assembler ARMv7	2
2. Interrupciones Básicas - SWI	2
3. Interrupciones de HW y Excepciones – Timer tick.....	3
4. MMU – Paginación Básica	4
5. Task Scheduler Simple	4
6. Manejo de privilegios	4



La presente guía puede ser resuelta utilizando el lenguaje que considere más adecuado. La cátedra recomienda el uso de ensamblador y C (C11).

1. ABI Y ASSEMBLER ARMv7

Se desea escribir un programa para simular en QEMU, que en el vector de arranque pueda copiarse a si mismo a la dirección 0x70030000. Para tal fin implementar:

void *td3_memcpy(void *destino, const void *origen, unsigned int num_bytes);

Una vez que se copia a esa dirección debe saltar a una etiqueta **end**, en donde se espera una instrucción que frene la ejecución del procesador.

La pila debe ubicarse en 0x70020000.

Se sugiere utilizar un archivo de linker script para poder gestionar las áreas de memoria durante el ensamblado del binario

Objetivos conceptuales

- I. *Familiarizarse con los comandos de **Qemu** y su utilización*
- II. *Familiarizarse con Linker Script*
- III. *Entender el concepto de ABI para poder utilizar convenientemente C y ASM, aprovechando sus ventajas para cada caso.*
- IV. *Entender manejo de pila **R13***
- V. *Entender como cargar el registro **LR** para poder retornar de las funciones.*

2. INTERRUPCIONES BÁSICAS - SWI

Se desea continuar el ejercicio anterior e incorporar la atención de interrupciones/excepciones.

Para tal fin, se deberá definir un handler genérico para atender la interrupción **SVC**.

Para corroborar su funcionamiento, debemos poder llamar a la instrucción **SWI** y validar que **atendemos el request correctamente**.

Objetivos conceptuales

- i. *Familiarizarse con la inicialización de interrupciones/excepciones*
- ii. *Entender el concepto de servicio de sistema operativo, que luego sera usado por código de usuario para acceder a acciones privilegiadas, como hace un S.O real.*



3. INTERRUPCIONES DE HW Y EXCEPCIONES – TIMER TICK

Se desea continuar el ejercicio anterior e incorporar la inicialización de interrupciones por HW (IRQ/FIQ) y la atención de excepciones.

Para tal fin se debe inicializar el timer tick del procesador, y recibir la interrupción de HW en el handler para tal fin.

Se propone con un flag de compilación, poder recompilar el código de modo que fuerce una excepción. La misma debe ser atendida correctamente por el handler para tal fin.

El programa ahora debe quedar esperando por los eventos de interrupciones/excepciones. Para tal fin sugerimos no utilizar “B.” y usar algún mecanismo que no consuma recursos, a la espera de la atención de las interrupciones /excepciones.

Objetivos conceptuales

-
- i. *Familiarizarse con IRQ vs FIQ.*
 - ii. *Comprender la atención de excepciones.*
 - iii. *Comprender la importancia de poner el procesador en alta impedancia para reducir el consumo de energía mientras esperamos por eventos.*