

# Trabajo Práctico

## ARMando el juego de la vida

R-222 - Arquitectura del Computador

### Introducción

El objetivo de este trabajo es que el alumno aprenda y se familiarice con la arquitectura ARM y los detalles que la caracterizan. Se espera además que, a través del desarrollo de este trabajo, el alumno adquiera conocimientos prácticos en el manejo de *toolchains*, *cross-compilers*, *make* y virtualización de plataformas.

### Metodología

Para realizar el trabajo, el alumno debe resolver los siguientes puntos:

- Preparar su entorno de desarrollo; se necesitará una *toolchain* para poder compilar programas para ARM desde una PC de escritorio, al igual que un simulador para probarlos. Puede encontrar extensa información sobre como realizar esto en Internet; se recomienda que instale la *toolchain* de su distribución para ARM, o en caso de no existir, descargue binarios precompilados desde Linaro o CodeSourcery. En cuanto al simulador, recomendamos utilizar QEMU, que puede ser instalado desde el gestor de paquetes de casi todas las distribuciones.
- Desarrollar un programa en assembler que permita leer y escribir desde la terminal serie sin un sistema operativo en ejecución (también conocido como *bare metal*). Se recomienda que lo realice de forma modular, separando las rutinas de lectura y escritura para facilitar su posterior cambio. Este programa debe ser ejecutado en un simulador; se recomienda QEMU. Puede encontrar material de referencia sobre cómo escribir un programa ‘Hola Mundo!’ *bare metal* para un procesador Cortex-A8 y cómo realizar la posterior prueba del mismo con QEMU al final del documento.
- Implementar el ‘Juego de la vida’ de Conway. Este juego de cero jugadores es un ejemplo de autómata celular. En otras palabras, el autómata celular evoluciona siguiendo una serie de reglas fijas sobre un estado inicial, definido por el usuario. Puede verse el artículo de Wikipedia sobre el juego para obtener todos los detalles para su implementación.
- La implementación deberá permitir al usuario, como mínimo, configurar el estado inicial y deberá ofrecer un mecanismo para poder comenzar la simulación de la evolución celular. Probablemente necesite emitir códigos de escape de terminal para ‘dibujar’ la interfaz del juego; puede encontrar una lista en la sección de referencias. El código deberá ser compilable con *make*.

## Características adicionales

El alumno puede extender el trabajo (opcionalmente) con las siguientes mejoras:

- Adaptar el código creado para permitir su ejecución en un Cubieboard, BeagleBone Black, Raspberry Pi u otro dispositivo ARM. Puede encontrar un BeagleBone Black en el Departamento de Ciencias de la Computación.
- Adaptar el código creado para permitir su ejecución como una aplicación en espacio de usuario. Necesitará investigar la convención de llamada de C en ARM, o alternativamente, la forma de realizar *syscalls* si no desea usar *libc*. Puede probar su programa en hardware ARM, o utilizando la emulación de espacio de usuario de QEMU.

## Entrega del trabajo

El trabajo será evaluado por la cátedra mediante una presentación en computadora y opcionalmente, hardware ARM. El alumno debe entregar un informe de al menos dos páginas incluyendo datos académicos (integrantes del grupo, legajos, fechas) y reportando problemas y soluciones encontradas durante la realización del trabajo y posibles extensiones al mismo.

## Material y referencias

- Wikipedia: Juego de la vida
- Using Ubuntu ARM cross-compiler for bare metal programming
- Descarga de toolchain CodeSourcery
- Descarga de toolchain Linaro
- Códigos de terminal (ANSI/VT100)
- Resumen de la arquitectura ARM
- Tarjeta de referencia rápida de instrucciones ARM
- Ejemplo de syscalls en ARM