

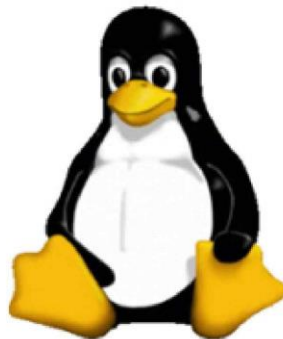
LECTURA 1

INFORME DE ARTICULO I

Alumno: Vicente Mata Velasco

Profesor: M.S.C. Eduardo Flores Gallegos

Asignatura: Sistemas Operativos I



Motivación

En este apartado motiva al docente a que tenga la noción de lo que se va a prender en este curso y como podrá ampliarlo para enseñárselo a sus alumnos La arquitectura ARM presenta muchas características que la distinguen de otras arquitecturas contemporáneas y ,por otro lado, al estar basada en RISC, es relativamente sencilla [5]. Además, el hecho de que ARM sea una arquitectura actual y ampliamente difundida, especialmente en dispositivos móviles, smartphones y tablets, es un factor especialmente motivador para los estudiantes, después de esto se inició un proceso de redefinición de las guías docentes y de los materiales utilizados en la enseñanza tanto teórica como práctica de las distintas asignaturas relacionadas con la materia de Arquitectura de Computadores.

. El simulador xspim nos parecía un entorno adecuado para las prácticas de entrada/salida de la asignatura debido a que dicho simulador incorpora un pequeño programa monitor(cuyo código fuente está disponible y puede ser adaptado) y simula un conjunto de dispositivos de entrada/salida: teclado, pantalla y reloj, que pueden sincronizarse por consulta de estado o por interrupciones.

Creemos que esta segunda opción es mucho más enriquecedora para el estudiante que simplemente limitarse a observar en la pantalla de un simulador si su código de gestión de la entrada/salida se comporta como teóricamente debería. Es más, puesto que mucha de la problemática de la entrada/salida está directamente relacionada con la interacción hombre-máquina, esta segunda opción ofrece al estudiante la oportunidad de enfrentarse a un escenario real, en el que si no se toman las debidas precauciones, no todo funcionará como se espera.

En concreto, el modelo Due de Arduino3 presenta las siguientes características que consideramos interesantes desde el punto de vista de la docencia de la entrada/salida utilizando ARM como arquitectura de referencia:

- Microcontrolador Atmel SAM3X8E con una CPU ARM Cortex-M3.
- 54 pines de entrada/salida digitales (de los cuales 12 pueden utilizarse como salidas PWM).
- 12 entradas analógicas.
- 4 UARTs (puertos serie hardware).
- Conexión USB con capacidad on-the-go (OTG).
- 2 conversores digital a analógico.

- Reloj (a 84 Mhz).
- Conectores SPI y JTAG.

Contexto docente

Es una asignatura de orientación básica tiene la siguiente asignación básica «Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería».

La asignatura se ha estructurado en tres temas, el tercero de los cuales abarca los sistemas de entrada/salida.

El estudiante, al finalizar la asignatura, deberá ser capaz de: 1. Describir la problemática general de la entrada/salida.

2. Definir los términos de latencia, tasa de transferencia y productividad y ser capaz de estimar o calcular sus valores partiendo de datos suficientes.

3. Caracterizar los sistemas de entrada/salida en función de su comportamiento, interlocutor, tasa de transferencia y latencia.

4. Describir la estructura general de los dispositivos de entrada/salida.

5. Enumerar y describir las formas por medio de las cuales el procesador puede acceder a los dispositivos de entrada/salida.

6. Enumerar los tipos de registros de los dispositivos de entrada/salida y explicar cómo y para qué se utilizan.

7. Explicar y evaluar el funcionamiento, así como las ventajas y desventajas, de los dos sistemas de sincronización de la entrada/salida: I) consulta de estado (polling) e II) interrupciones.

8. Explicar y evaluar el funcionamiento, así como las ventajas y desventajas, de los dos sistemas de transferencia de datos de entrada/salida: I) transferencias por programa realizadas por el procesador y II) acceso directo a memoria (DMA).

9. Describir las características generales de la entrada/salida de propósito general (GPIO) y de los mecanismos de temporización de los ordenadores.

10. Analizar programas escritos en ensamblador Thumb 2 que gestionen dispositivos de entrada/salida.

11. Desarrollar programas en ensamblador Thumb 2 que gestionen dispositivos de entrada/salida.

Entrada/salida con arduino

Introducción a la entrada/salida. En esta primera sesión se muestra cómo encender o apagar el LED de la tarjeta de entrada/salida; cómo escribir información por medio del puerto serie de la tarjeta Arduino; y cómo acceder a la información del reloj de tiempo real para obtenerla fecha y hora actual de la tarjeta Arduino Due.

Entrada/salida por consulta de estado. En esta sesión se incide sobre la consulta de estado y su problemática. El estudiante deberá programar el entorno Arduino para detectar la pulsación del pulsador de la tarjeta de entrada/salida y actuar en consecuencia. También tendrá que programar una alarma en el reloj en tiempo real de la tarjeta Arduino y averiguar por consulta de estado si la alarma ha saltado o no.

Entrada/salida por interrupciones. En esta sesión el estudiante debe configurar la tarjeta Arduino para activar las interrupciones y probar las distintas congelaciones disponibles. Se propone utilizar reloj pulsador de la tarjeta de entrada/salida para comprobar cómo varía el tratamiento de las interrupciones en función de la configuración seleccionada. El estudiante también deberá programar el reloj en tiempo real para fijar una alarma de tal forma que cuando ésta salte, se genere una interrupción (y comparar el comportamiento de esta práctica con el observado en la práctica anterior, en la que se averiguaba si la alarma se había generado o no por medio de la consulta de estado).

Entorno de desarrollo modificado El entorno de desarrollo de Arduino, que puede descargarse desde su página web, no soporta la realización de proyectos Arduino en otros lenguajes de programación que no sean C o C++. Afortunadamente, puesto que dicho entorno de desarrollo es software libre, hemos podido estudiar su código y modificarlo para que también aceptara y compilara programas en lenguaje ensamblador. Utilizando esta versión modificada es posible desarrollar proyectos que tengan ficheros fuente tanto en C/C++, como en ensamblador de ARM. Esto nos ha permitido graduar la dificultad de las prácticas de tal forma que determinadas secciones se realizan inicialmente en C/C++ y posteriormente se muestra qué acciones se deben realizar a bajo nivel para conseguir el mismo efecto.