



RIDAA
Repositorio Institucional
Digital de Acceso Abierto de la
Universidad Nacional de Quilmes



**Universidad
Nacional
de Quilmes**

Guagliano, Celeste

Organización y arquitectura de computadores



Esta obra está bajo una Licencia Creative Commons Argentina.
Atribución - No Comercial - Compartir Igual 2.5
<https://creativecommons.org/licenses/by-nc-sa/2.5/ar/>

Documento descargado de RIDAA-UNQ Repositorio Institucional Digital de Acceso Abierto de la Universidad Nacional de Quilmes de la Universidad Nacional de Quilmes

Cita recomendada:

Guagliano, C. (2024). *Instrumentos y mediciones. (Programa)*. Bernal, Argentina: Universidad Nacional de Quilmes. Disponible en RIDAA-UNQ Repositorio Institucional Digital de Acceso Abierto de la Universidad Nacional de Quilmes <http://ridaa.unq.edu.ar/handle/20.500.11807/6249>

Puede encontrar éste y otros documentos en: <https://ridaa.unq.edu.ar>

PROGRAMA de ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORES

Carrera: Ingeniería en Automatización y Control Industrial.

Asignatura: Organización y Arquitectura de Computadores ¹

Núcleo al que pertenece: *Núcleo Superior Básico* ²

Profesoras/les: *Celeste Guagliano*

Asignaturas previas necesarias para favorecer el aprendizaje: Informática, Algoritmos y Programación - Técnicas Digitales "A"

Objetivos:

Se espera que quienes cursen la asignatura logren:

- Familiarizarse con la estructura moderna de microcomputación basada en el paradigma de Von-Neumann
- Cubrir de forma teórico-práctica aspectos de organización interna, arquitectura, herramientas de programación, y manejo de las interfaces con la memoria y con el mundo externo (entrada-salida).

Contenidos mínimos:

Introducción a la organización y arquitectura de computadores. Modelo Von Neumann. Funcionamiento. Estructuras y buses. Máquina de estados finitos (FSM) y camino de datos (DP). Ejecución de instrucciones. Análisis de prestaciones. Unidad Central de Procesos (CPU): operación, ejemplificación con procesadores comerciales típicos. Conjunto de instrucciones. CISC versus RISC. Subsistema de memoria: jerarquías: principal y secundaria. Memoria caché: prestaciones e implementación. Memoria virtual. Subsistema de entrada-salida (I/O): periféricos serie y paralelo, formatos típicos de transmisión. Discos magnéticos. Mapeado de la entrada-salida. Administración vía polling e interrupciones. Acceso directo a memoria (DMA). Programación. Asignación dinámica de la memoria: listas y árboles.

1 En el plan vigente Plan vigente, RCS N°455-15. Para el Plan RCS N° 277/11 y Res CS N.º 179/03 se llama *Computadores I*

2 En el plan vigente *Plan vigente*, RCS N°455-15. Para el Plan RCS N° 77/11 y Res CS N.º 179/03 pertenece al *Núcleo Complementario del Ciclo Diplomatura*

Gestión de archivos. Nociones de firmware y de sistema operativo como administrador recursos y aplicaciones.

Carga horaria semanal: 6 horas

Programa analítico:

1. Concepto de Arquitectura. Relación con Organización de Computadoras. Repaso del modelo de von Neumann. Descripción del funcionamiento de un sistema basado en un microprocesador. Buses, teoría de operación, buses sincrónicos y asincrónicos. Ejemplos. Repaso de ejecución de instrucciones. Ejecución solapada ("pipeline"). Su aplicación en procesadores contemporáneos. Análisis de performance. Arquitecturas reconfigurables: conceptos.
2. Máquinas que ejecutan instrucciones almacenadas en memoria. Ejemplificación en procesadores típicos. Análisis del conjunto de instrucciones de procesadores de uso comercial. Concepto de máquinas CISC y RISC. Ejemplificación con las familias x86 y ARM. Lineamientos básicos en el diseño de un procesador RISC. Análisis de prestaciones. Ejemplos. Interrupciones: tratamiento general. Interrupciones por software y por hardware, vectores, descripción y tratamiento particular de cada una. Relación entre las interrupciones y el manejo de operaciones de E/S.
3. Organización jerárquica de la memoria, memoria principal y memoria secundaria. Memoria caché, concepto y descripción, análisis de prestaciones, métodos de implementación típicos, múltiples niveles. Ejemplos. Conceptos de memoria virtual. Tecnologías y medidas de performance. Discos rígidos. Caracterización y medidas de performance. Esquemas *RAID*. Cintas y sistemas de *Back-Up*.
4. Concepto de E/S y su relación con la CPU, tipos de puertas. Concepto de puerta de Entrada y Salida paralelo. Concepto de puerta de Entrada y Salida serie. Tipos de transmisión serie (UART, SPI, I2C). Descripción del formato de transmisión serie asincrónica y sincrónica. Descripción funcional de una puerta de E/S serie, acceso a registros internos para control y determinación del estado de operación de la puerta. Mapeado del subsistema E/S y la memoria. Administración de las puertas por encuesta (polling) o por interrupción. Tratamiento de la CPU de las operaciones de E/S, por

interrupción o por software. Transferencias de E/S por hardware, DMA, implementación. Ejemplificación con discos rígidos, conversores A/D, etc.

5. Programación de microprocesadores: Conjunto de Instrucciones, Modos de Direccionamiento y Tipos de Datos. Formato de Instrucción. Código Objeto. Representación simbólica, lenguaje ensamblador. Lenguajes de programación de Alto Nivel (*HLL*). Soporte de la arquitectura a Procedimientos y Funciones de un *HLL*. Manejo de la pila, Marco de Pila (*Stack Frame*). Pasaje de parámetros. Ejemplos de implementación en lenguaje C estándar.
6. Sistemas operativos: funciones y tipos. Sistemas básicos de entrada-salida (*BIOS*). *Firmware*. *System Calls*. El modelo de proceso. Protección, sincronización y conmutación de procesos. Manejo de interrupciones. Administración de memoria: sus objetivos. Memoria Virtual, traducción de direcciones y sus mecanismos: reubicación. Paginado, segmentación y segmentación paginada. Accesos a recursos compartidos. Bloqueos (“dead – locks”) y su prevención
7. Programación de microcontroladores en lenguaje Assembler y en lenguaje C estándar. Comparación, ventajas y desventajas. Programación estructurada. Herramientas de desarrollo: Ambientes Integrados de Desarrollo (*IDE*). Compiladores y el proceso de compilación. Asignación dinámica de la memoria: listas y árboles. Gestión de archivos. Firmware y facilidades del fabricante de microcontroladores para el programador: librerías, interfaces, estándares

Trabajos prácticos:

TP1. Comenzando a programar en assembler

Objetivos: que el/la estudiante comprenda el funcionamiento interno de una computadora basada en Intel mediante el uso del simulador MSX88 programando en código máquina de bajo nivel para poder realizar el seguimiento de los datos en diferentes partes del sistema

Actividades: Realización de programación variada en assembler según el set de instrucción de MSX88 explorando las características de la arquitectura Intel 8086

TP2. Assembler 2: arquitectura ARM

Objetivos: que el/la estudiante sea capaz de comprender las diferentes arquitecturas disponibles en el mercado y analizar ventajas y desventajas de cada una en diferentes situaciones problemáticas

Actividades: Realización de programación variada en assembler modo thumb en ARM según el set de instrucciones de QTARMSim explorando las características de la arquitectura ARM de 32 bits y modo reducido.

Bibliografía obligatoria:

- **Organización y Arquitectura de Computadoras** - Diseño para optimizar prestaciones, William Stallings. Séptima Edición - Ed. Prentice Hall (2006). (libro de cabecera)
- **Arquitectura de Computadores** - Un enfoque cuantitativo, John Hennessy & David Patterson. Ed. Mc Graw Hill (1999).
- **Organización de Computadoras**, Andrew Tanenbaum. Ed. Prentice Hall (2000).
- **Sistemas operativos modernos**. Tanenbaum, A. S. (2003). Pearson Educación.
- **Manuales varios de las familias Intel x86 y ARM.**
- **Manuales varios simulador MSX88 y ARM.**

Organización de las clases:

Clases teóricas complementadas con el desarrollo de trabajos prácticos de escritorio, y de laboratorios en donde los estudiantes utilizan tanto programas simuladores de microcontroladores como así también *kits* de evaluación de microprocesadores de la familia ARM, programando en lenguaje *Assembler* y en C estándar la resolución de problemas de cómputo sencillos asociados al manejo de la microprocesador y sus periféricos.

Modalidad de evaluación:

Dos evaluaciones parciales, cada una con un recuperatorio, y evaluación integradora. Aprobación de un trabajo final de laboratorio.

Aprobación de la asignatura según Régimen de Estudios vigente de la Universidad Nacional de Quilmes:

La aprobación de la materia bajo el régimen de regularidad requerirá: Una asistencia

no inferior al 75 % en las clases presenciales previstas, y cumplir con al menos una de las siguientes posibilidades:

(a) la obtención de un promedio mínimo de 7 puntos en las instancias parciales de evaluación y de un mínimo de 6 puntos en cada una de ellas.

(b) la obtención de un mínimo de 4 puntos en cada instancia parcial de evaluación y en el examen integrador, el que será obligatorio en estos casos.

Este examen se tomará dentro de los plazos del curso.

Los alumnos que obtuvieron un mínimo de 4 puntos en cada una de las instancias parciales de evaluación y no hubieran aprobado el examen integrador mencionado en el Inc. b), deberán rendir un examen integrador en las instancias que la UNQ destine para tal fin.

Modalidad de evaluación exámenes libres:

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral en instancias de evaluación similares a las realizadas en la modalidad presencial. Los contenidos a evaluar serán los especificados anteriormente.

Anexo II

CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Presentación – Unidad 1	X				
2	Unidad 1	X	X			
3	Unidad 2	X	X			
4	Unidad 2	X	X	X		
5	Unidad 3	X	X			
6	Unidad 3	X	X	X		
7	Unidad 4	X	X			
8	Unidad 4-Repaso	X	X			
9						1° Examen
10	Recuperatorio-Unidad 5					Rec. 1° Examen
11	Unidad 5	X	X	X		
12	Unidad 6	X	X			
13	Unidad 6	X	X			
14	Unidad 7	X	X	X		
15	Unidad 7-Repaso	X	X			
16						2° Examen
17				X		Entrega Laboratorio
18	Consulta-Integrador					Rec. 2° Examen / Integrador