Grado en Ingeniería Informática
30205 - Arquitectura y organización de computadores 1

Guía docente para el curso 2012 - 2013

Curso: 1, Semestre: 2, Créditos: 6.0

Información básica

Profesores

- **Carlos José Pérez Jiménez** cjperez@unizar.es
- **Luis Carlos Aparicio Cardiel** luisapa@unizar.es
- **María Villarroya Gaudó** maria.villarroya@unizar.es
- **Rubén Gran Tejero** rgran@unizar.es
- **Javier Diaz Maag** jdmaag@unizar.es
- **Luis Caballero Fernandez**
- **Luis Carlos Gállego Rodriguez** lcgalleg@unizar.es

Recomendaciones para cursar esta asignatura

Para cursar esta asignatura es prerrequisito haber cursado la asignatura Introducción a los Computadores.

Actividades y fechas clave de la asignatura

El calendario de exámenes y las fechas de entrega de trabajos de evaluación se anunciará con suficiente antelación.

Inicio

Resultados de aprendizaje que definen la asignatura

El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados...

1:
Conoce a nivel básico los parámetros que definen la de arquitectura de lenguaje máquina (repertorio, formato y codificación de instrucciones, almacenes, tipos de datos, modos de direccionamiento, control del secuenciamiento y transferencias de control, gestión de excepciones).

2:

Conoce y puede manejar la arquitectura de lenguaje máquina de un procesador de referencia.

3:
Distingue los conceptos de lenguaje máquina y ensamblador.

4:
Conoce los métodos de representación y codificación de la información y sus operaciones básicas. Es capaz de traducir estructuras de datos y control de lenguajes de alto nivel a ensamblador. Utiliza llamadas a procedimiento.

5:
Entiende el modelo genérico de registros de un controlador de dispositivo periférico y los métodos básicos de sincronización y transferencia. Puede programar cualquier dispositivo de E/S y sabe cómo tratar las excepciones.

6:
Sabe integrar código ensamblador y rutinas de librería en programas escritos en lenguajes de alto nivel.

Introducción

Breve presentación de la asignatura

En esta asignatura se comprenderá cuál es la estructura interna de un computador, en un primer acercamiento a una arquitectura real dentro del Grado en Ingeniería Informática.

Contexto y competencias

Sentido, contexto, relevancia y objetivos generales de la asignatura

La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:

Esta asignatura de introducción busca que cada estudiante sea capaz de comprender una arquitectura de lenguaje máquina y diseñar programas en lenguaje ensamblador capaces de comunicarse con periféricos.

Contexto y sentido de la asignatura en la titulación

Esta asignatura forma parte de la materia básica de Arquitectura de Computadores en el Grado de Ingeniería Informática. La asignatura enlaza con Introducción a los Computadores y es requisito para cursar Arquitectura y Organización 2.

Al superar la asignatura, el estudiante será más competente para...

1:
Resolver problemas y tomar decisiones con iniciativa, creatividad y razonamiento.

2:
Aplicar las tecnologías de la información y las comunicaciones en ingeniería.

3:
Utilizar conocimientos básicos sobre el uso y programación de los ordenadores, bases de datos y programas informáticos con aplicación en ingeniería.

4:
Aplicar 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.

Estas dos últimas competencias se desarrollan de forma específica para reforzar las dos primeras,

persiguiendo los resultados de aprendizaje de la asignatura (léase apartado Introducción, resultados de aprendizaje de esta guía).

Importancia de los resultados de aprendizaje que se obtienen en la asignatura:

Esta asignatura fundamenta el diseño, programación y uso eficiente del computador, ya se este de propósito general o específico (empotrado, supercomputación, etc.).

Evaluación

Actividades de evaluación

El estudiante deberá demostrar que ha alcanzado los resultados de aprendizaje previstos mediante las siguientes actividades de evaluacion

1:

La evaluación de la asignatura se realiza en base a dos pruebas:

P1. Prueba escrita en la que responder cuestiones y resolver ejercicios y problemas. Se requiere una nota mínima de 5.0 puntos en esta prueba para aprobar la asignatura. Si se obtiene esta nota mínima, entonces la prueba pondera un 70% en la nota de la asignatura y, si no se alcanza este mínimo, entonces la calificación en la asignatura es la de esta prueba.

P2. Trabajos y prueba de laboratorio. Esta prueba pondera un 30% en la nota de la asignatura. Cada alumno deberá entregar los trabajos que se indiquen en las prácticas de la asignatura y presentarse a una prueba de laboratorio dónde deberán realizar alguna tarea relacionada con una parte del trabajo entregado. Se valorará tanto el trabajo entregado como el realizado en la prueba, con pesos análogos. El alumno que no presente los trabajos de programación que se indiquen o que no se presente a la prueba será calificado con un cero.

Pruebas con carácter voluntario. A lo largo del cuatrimestre se plantearán varias pruebas voluntarias consistentes en la resolución, por escrito en clase o en casa, de ejercicios y problemas. El 10% de la calificación de estas pruebas, es decir, entre 0 y 1.0 puntos, se sumará a la calificación obtenida por el alumno en la prueba **P1**, siempre y cuando haya aprobado la asignatura (0.7*P1+0.3*P2>=5.0).

Convocatoria de Septiembre. La evaluación de la asignatura se realiza en base a dos pruebas análogas a las de la convocatoria de Junio, con las mismas ponderaciones y exigencia de notas mínimas Las calificaciones del alumno obtenidas en la convocatoria de Junio en cualquier de las pruebas (P1 y P2) se mantienen en Septiembre, salvo que el alumno opte por presentarse a la prueba correspondiente en esta nueva convocatoria, en cuyo caso prevalecerá la nueva calificación.

Actividades y recursos

Presentación metodológica general

El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:

La asignatura se asienta sobre clases presenciales al principio ya que el alumnado no dispone de conocimientos previos. Esta actividad irá acompañada del estudio teórico por parte de cada estudiante.

En grupos reducidos se establecerán sesiones semanales de aprendizaje basado en problemas y/o aplicaciones de teoría y

de sesiones prácticas en laboratorio. Será preciso trabajo autónomo de preparación de las actividades y entregas en las sesiones prácticas.

Complementario a la sesión de prácticas se realizará un trabajo tutorizado de forma personalizada y de carácter eminentemente práctico, hacia el final de cuatrimestre . Este trabajo será evaluado de forma independiente.

Así mismo se realizarán tutorías no académicas y podrían complementarse con otro tipo de actividades formativas voluntarias (seminarios, por ejemplo).

Actividades de aprendizaje programadas (Se incluye programa)

El programa que se ofrece al estudiante para ayudarle a lograr los resultados previstos comprende las siguientes actividades...

1:

El programa a desarrollar en esta asignatura es:

- Arquitectura del Procesador: Interpretación y traducción, lenguaje máquina y ensamblador, entorno de desarrollo, representación y codificación de la información, operaciones básicas, almacenes, modos de direccionamiento, repertorio de instrucciones, traducción de estructuras de datos y control de lenguajes de alto nivel, llamadas a procedimiento
- Subsistema de E/S. Modelo genérico de registros de controlador de dispositivo. Métodos básicos de sincronización y transferencia. Excepciones. Integración de periféricos en microcontroladores.
- Caso práctico. Integración de código de alto nivel con código ensamblador y rutinas de biblioteca .

Los contenidos de la asignatura se desarrollaran entorno a los siguientes puntos

- Introducción
- Arquitectura
- Arquitectura del Lenguaje Máquina ALMA
- Control de Flujo
- Subrutinas
- Entrada Salida
- Excepciones
- Sincronización

2:

Para ello se realizarán las siguientes actividades:

- Actividades presenciales: Campus Río Ebro

Actividad tipo 1 (clases magistrales) 2 horas/semana 2 grupos

Actividad tipo 2 (clases participativas) 1 hora/semana 4 grupos

Actividad tipo 3 (clases de prácticas) 2 horas/quincena 10 grupos

Actividad tipo 4 (trabajo práctico tutorizado) 8 horas de trabajo de estudiante, fuera de laboratorio

Tutorías y actividades de evaluación.

- Actividades presenciales: Campus de Teruel

Actividad tipo 1 (clases magistrales) 2 horas/semana 1 grupo

Actividad tipo 2 (clases participativas) 1 hora/semana 2 grupos

Actividad tipo 3 (clases de prácticas) 1 hora/semana 2 grupos

Actividad tipo 4 (trabajo práctico tutorizado) 8 horas de trabajo de estudiante, fuera de laboratorio

Tutorías y actividades de evaluación.

- Actividades no presenciales: Ambos Campus

Trabajos prácticos, estudio teórico, estudio práctico y actividades complementarias, con una dedicación efectiva de aproximadamente 80 horas por semestre.

Planificación y calendario

Calendario de sesiones presenciales y presentación de trabajos

La organización docente de la asignatura es la siguiente.

- Clases teóricas (2 horas semanales)
- Clases de problemas (1 hora semanal)
- Clases prácticas de laboratorio (2 horas cada dos semanas). Son sesiones de trabajo de programación en laboratorio, tuteladas por un profesor, en las que participan los alumnos en grupos reducidos.
- Tutorías y actividades de evaluación

Los horarios de todas las clases y las fechas de las sesiones de prácticas se anunciarán con suficiente antelación a través de las webs del centro y de la asignatura.

Trabajo del estudiante

La dedicación del estudiante para alcanzar los resultados de aprendizaje en esta asignatura se estima en 150 horas distribuidas del siguiente modo:

- 56 horas, aproximadamente, de actividades presenciales (clases teóricas, de problemas y prácticas en laboratorio)
- 51 horas de estudio personal efectivo (estudio de apuntes y textos, resolución de problemas, preparación clases y prácticas, desarrollo de programas)
- 40 horas de trabajo de programación en equipo
- 3 horas de examen final escrito

Bibliografía

Bibliografía de la asignatura

Bibliografía básica:

1. ***ARM Assembly Language: Fundamentals and Techniques***, William Hohl. CRC press. 2009

Bibliografía complementaria:

1. ***ARM System-on-Chip Architecture*** (2nd ed.), Steve Furber. Addison-Wesley. 2000
2. ***ARM Architecture Reference Manual*** (2nd ed.), David Seal. Addison-Wesley. 2000
3. ***ARM System Developer's Guide: Designing and Optimizing System Software***, Andrew N. Sloss, Dominic Symes, Chris Wright. Elsevier. 2004
4. ***Organización y Arquitectura de computadores*** (7ª ed.). W. Stallings. Prentice Hall. 2006

Referencias bibliográficas de la bibliografía recomendada

Escuela de Ingeniería y Arquitectura

- 1. Hohl, William. ARM Assembly language : fundamentals and techniques / William Hohl Boca Raton (Florida) : CRC Press, cop. 2009
- 4. Sloss, Andrew. ARM system developer's guide : designing and optimizing system software / Andrew Sloss, Dominic Symes, Chris Wright ; with a contribution by John Rayfield San Francisco, CA : Elsevier/ Morgan Kaufman, cop. 2004
- 5. Stallings, William. Organización y arquitectura de computadores / William Stallings ; traducción Antonio Cañas Vargas ... [et al.] ; coordinación y revisión técnica Alberto Prieto Espinosa . - 7ª ed. Madrid : Pearson Educación, D. L. 2006
- Furber, Stephen Bo. ARM system-on-chip architecture / Steve Furber . 2nd ed. Harlow : Addison-Wesley, 2000
- Seal, David. ARM Architecture reference manual / David Seal. - 2nd ed. Harlow : Addison-Wesley, 2000

Escuela Universitaria Politécnica

- Furber, Stephen B.|q(Stephen Bo). ARM system-on-chip architecture / Steve Furber . - 2nd ed. Harlow : Addison-Wesley, 2000
- Hohl, William. ARM Assembly language : fundamentals and techniques / William Hohl Boca Raton (Florida) : CRC Press, cop. 2009
- Seal, David. ARM Architecture Reference Manual / David Seal. . - (2nd ed.) [s.l.] : Addison-Wesley. 2000
- Sloss, Andrew. ARM system developer's guide : designing and optimizing system software / Andrew Sloss, Dominic Symes, Chris Wright ; whit a contribution by John Rayfield San Francisco, CA : Elsevier/ Morgan Kaufman, cop. 2004
- Stallings, William. Organización y arquitectura de computadores / William Stallings ; traducción Antonio Cañas Vargas ... [et al.] ; coordinación y revisión técnica Alberto Prieto Espinosa . - 7ª ed. Madrid : Pearson Educación, D. L. 2006