

Guía docente

330228 - PBN - Programación de Bajo Nivel

Última modificación: 25/04/2024

Unidad responsable: Escuela Politécnica Superior de Ingeniería de Manresa
Unidad que imparte: 750 - EMIT - Departamento de Ingeniería Minera, Industrial y TIC.

Titulación: GRADO EN INGENIERÍA DE SISTEMAS TIC (Plan 2010). (Asignatura obligatoria).

Curso: 2024 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: SEBASTIAN VILA MARTA

Otros: Aguila Lopez, Francisco Del
Bonet Dalmau, Jordi

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:

1. El conocimiento y la capacidad de aplicar los procedimientos algorítmicos fundamentales a la resolución de problemas haciendo uso de lenguajes de alto y bajo nivel.
2. La capacidad de analizar, diseñar y mantener aplicaciones informáticas así como el conocimiento de los principios y herramientas de la ingeniería del software y su aplicación.
3. El conocimiento y la capacidad de usar las herramientas y la instrumentación existentes para el análisis, el diseño, el desarrollo y la verificación de sistemas electrónicos, informáticos y de comunicaciones.
4. La capacidad para desarrollar las actividades propias del grado considerando los estándares, reglamentos y normas reguladoras correspondientes.
5. La capacidad para analizar, diseñar e implementar, seleccionar y utilizar sistemas de tratamiento de datos, control y automatización en tiempo real, especialmente en sistemas empujados.

Transversales:

7. USO SOLVENTE DE LOS RECURSOS DE INFORMACIÓN - Nivel 2: Después de identificar las diferentes partes de un documento académico y de organizar las referencias bibliográficas, diseñar y ejecutar una buena estrategia de búsqueda avanzada con recursos de información especializados, seleccionando la información pertinente teniendo en cuenta criterios de relevancia y calidad.
8. APRENDIZAJE AUTÓNOMO - Nivel 3: Aplicar los conocimientos alcanzados en la realización de una tarea en función de la pertinencia y la importancia, decidiendo la manera de llevarla a cabo y el tiempo que es necesario dedicarle y seleccionando las fuentes de información más adecuadas.
9. COMUNICACIÓN EFICAZ ORAL Y ESCRITA - Nivel 2: Utilizar estrategias para preparar y llevar a cabo las presentaciones orales y redactar textos y documentos con un contenido coherente, una estructura y un estilo adecuados y un buen nivel ortográfico y gramatical.
- 05 TEQ N2. TRABAJO EN EQUIPO - Nivel 2: Contribuir a consolidar el equipo planificando objetivos, trabajando con eficacia y favoreciendo la comunicación, la distribución de tareas y la cohesión.

METODOLOGÍAS DOCENTES

La asignatura se estructura en dos clases de dos horas por semana. De estas cuatro horas presenciales semanales una se dedica a presentar los principales contenidos de manera expositiva, la segunda a la resolución de problemas bajo demanda del estudiantado y las dos restantes a resolver problemas prácticos en el laboratorio informático.

Al estudiante se le indican semanalmente las tareas de estudio y solución de problemas que debe hacer. Estas tareas se aconseja hacerlas, por lo menos parcialmente, trabajando en equipo. Periódicamente se evalúa el progreso de cada estudiante individualmente. La asignatura también incorpora un proyecto de desarrollo de software de tamaño medio que hay que trabajar en equipo.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Después de superar esta asignatura el estudiante:

1. Sabrá usar un lenguaje de programación y la cadena de herramientas asociadas al desarrollo de software que interacciona directamente con el hardware.
2. Conocerá los principios de la ingeniería del software y su aplicación práctica a la gestión y desarrollo de proyectos de programación propios del ámbito del título.
3. Sabrá obtener e interpretar información técnica y ser capaz de comunicar resultados de forma oral y escrita.
4. Conocer los principios y algunas herramientas de verificación y validación o de software y ser capaz de aplicarlas a problemas reales.
5. Poder redactar memorias técnicas sencillas, también en una tercera lengua, y presentarlas oralmente.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTE

Tipo	Horas	Porcentaje
Horas grupo pequeño	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00
Horas grupo grande	30,0	20.00

Dedicación total: 150 h

CONTENIDOS

TEMA 1: Lenguaje de programación C

Descripción:

Se inicia al estudiante en el uso del lenguaje de programación C, en su definición de acuerdo con último estándar ISO. Paralelamente el estudiante avanza en el conocimiento y la capacidad de usar la cadena de trabajo típica del proceso de compilación y las herramientas que la conforman, herramientas que incluyen el compilador, el montador, el gestor de librerías y make. En cuanto al lenguaje C, enfatizan aquellas estructuras y características del lenguaje que serán posteriormente más útiles para escribir programas de bajo nivel: apuntadores, tablas como regiones de memoria, uso de registros en paso de parámetros, etc.

Palabras clave: Lenguaje C. Proceso de compilación y montaje. Módulo. Librerías.

Preprocesador. Make.

Objetivos específicos:

Una vez alcanzado el tema:

1. El estudiante debe ser capaz de escribir programas usando C y procesarlos hasta obtener el ejecutable correspondiente.
2. El estudiante debe conocer la función de cada uno de los pasos del proceso de compilación-montaje y de los datos intermedios que se generan.
3. El estudiante debe saber sacar provecho de las herramientas de trabajo clásicas (compilador, montador, gestor de librerías y make) en un entorno de trabajo Linux.

Actividades vinculadas:

Todas las que constan.

TEMA 2: El lenguaje C en el contexto de la AVR

Descripción:

Una vez introducido el lenguaje de programación C desde un punto de vista general, el objetivo es ahora exponer al estudiante las variaciones y especificaciones típicas que puede encontrarse al usar C como lenguaje para programar pequeños dispositivos como puede ser el MCU AVR.

Los aspectos a contemplar incluyen: especificados del lenguaje C y su relación con el estándar, acceso a los dispositivos periféricos, tratamiento de interrupciones, la influencia del modelo de MCU, el proceso de implantación del programa, aspectos sobre la eficiencia del programa compilado, programas híbridos ensamblador-C.

Palabras clave: Microcontrolador AVR, compilación y depuración cruzada, periféricos, interrupciones.

Objetivos específicos:

Una vez alcanzado el tema, el estudiante:

1. Debe saber diseñar programas para el MCU AVR, compilarlos y montarlos en una estación de trabajo e implantarlos sobre el microcontrolador.
2. Comprenderá la relación existente entre los lenguajes de alto y bajo nivel.
3. Ser capaz de implementar aplicaciones parcialmente escritas con ensamblador y el lenguaje de programación C.
4. Ser capaz de sacar provecho de un emulador de un MCU AVR en el proceso de desarrollo.

Actividades vinculadas:

Todas las que constan.

TEMA 3: Python en el contexto de bajo nivel

Descripción:

El objetivo es aprender a extender el intérprete de Python usando extensiones escritas en C hasta conseguir crear un entorno de trabajo Python capaz de gobernar dispositivos de bajo nivel de forma coherente.

Palabras clave: Extensiones de Python. Interacción entre una aplicación Python y un dispositivo hardware.

Objetivos específicos:

Una vez alcanzado el tema, el estudiante:

1. Debe poder escribir aplicaciones usando Python que interaccionen con dispositivos basados en microcontroladores.
2. Debe ser capaz de extender la funcionalidad de Python usando código escrito en C.

Actividades vinculadas:

Todas las que constan.

ACTIVIDADES

ACTIVIDAD 1: EXAMEN

Descripción:

La asignatura contempla un examen profesional que consiste en un conjunto de ejercicios a resolver individualmente sobre papel sin apoyo de ningún tipo de material y en un tiempo acotado.

Material:

Se entrega la solución individual del examen y se evalúa la misma.

Dedicación: 2h

Grupo grande/Teoría: 2h

ACTIVIDAD 2: ESTUDIO DE CONTENIDOS

Descripción:

El estudio de los contenidos es la actividad individual o colectiva que conduce a entender y asumir los conocimientos, vocabulario y técnicas que forman parte de los contenidos de la asignatura.

Material:

Los materiales de apoyo son:

- Referencia principal de la asignatura (libro en formato web).
- Colección de problemas de la asignatura.

Dedicación: 25h

Aprendizaje autónomo: 25h

ACTIVIDAD 3: CLASE EXPOSITIVA

Descripción:

Son clases presenciales específicamente dedicadas a la comprensión de los contenidos de la asignatura, especialmente aquellos de carácter más bien teóricos.

Material:

Los materiales de apoyo son:

- Referencia principal de la asignatura (libro en formato web).
- Bibliografía básica.
- Colección de problemas de la asignatura.

Dedicación: 12h

Grupo grande/Teoría: 12h

ACTIVIDAD 4: CLASE DE PROBLEMAS

Descripción:

Son clases presenciales específicamente dedicadas a la resolución de problemas. Se hacen en un aula ordinaria y son complementarias de la actividad en el laboratorio. Son clases que requieren la participación de los estudiantes.

Material:

Los materiales de apoyo son:

- Referencia principal de la asignatura (libro en formato web).
- Bibliografía básica.
- Colección de problemas de la asignatura.

Dedicación: 12h

Grupo grande/Teoría: 12h

ACTIVIDAD 5: CLASE DE LABORATORIO

Descripción:

El estudiante tiene como objetivo la solución de pequeñas prácticas que complementan los contenidos y colaboran en la mejor comprensión de los mismos. Las prácticas se realizan en el laboratorio y comportan la implementación real de programas sobre el computador y su comprobación. La actividad puede comportar la finalización de las prácticas en tiempo de aprendizaje autónomo.

Material:

Los materiales de apoyo son:

- Referencia principal de la asignatura (libro en formato web).
- Colección de problemas de la asignatura.
- Manuales del software utilizado.
- Bibliografía básica.

Entregable:

Periódicamente se entregan los resultados de diferentes prácticas que se van haciendo durante el curso.

Dedicación: 41h

Grupo pequeño/Laboratorio: 26h

Aprendizaje autónomo: 15h

ACTIVIDAD 6: RESOLUCIÓN DE PROBLEMAS

Descripción:

Es una actividad que hace el estudiante autónomamente y que consiste en la solución de problemas de programación, generalmente sin ser necesario el apoyo del computador.

Material:

Los materiales de apoyo son:

- Referencia principal de la asignatura (libro en formato web).
- Colección de problemas de la asignatura.

Entregable:

La actividad conlleva la entrega durante el curso de algunos problemas que se corrigen convenientemente y forman parte de la evaluación de la asignatura.

Dedicación: 30h

Aprendizaje autónomo: 30h

ACTIVIDAD 7: PROYECTO

Descripción:

La asignatura requiere realizar un proyecto de tamaño medio que integra una parte importante de los contenidos de la asignatura. Esta actividad se realiza en grupo y comporta, además, la escritura de un informe técnico sobre el programa. Esta actividad tiene naturaleza de síntesis de todos los conocimientos de la asignatura.

Material:

Los materiales de apoyo son:

- Servicio de laboratorio informático/ de electrónica de la EPSEM.
- Enunciado y guión del proyecto.
- Ejemplo de informe.
- Apuntes personales y demás material de apoyo del curso.

Entregable:

Como resultado de la actividad se entregan:

1. El informe del proyecto.
2. El código fuente resultado del proyecto.

La entrega se realiza con la presencia de todo el equipo de trabajo. Se evalúa el informe y el resultado a que se ha llegado en la confección del proyecto.

Dedicación: 28h

Grupo grande/Teoría: 4h

Grupo pequeño/Laboratorio: 4h

Aprendizaje autónomo: 20h

SISTEMA DE CALIFICACIÓN

La calificación se realiza en base a 3 elementos:

1. La evaluación del trabajo autónomo del estudiante (A). Este componente contiene tanto el progreso se hecho en los aspectos teóricos como en los prácticos. Su medición se realiza a base de ejercicios obligatorios entregados durante el curso.
2. La evaluación del trabajo práctico (P). Se realiza a partir de una entrega presencial del proyecto de curso que puede comportar una presentación pública y la confección de una memoria y de la entrega de las prácticas que se van realizando durante el curso.
3. El evaluación final (F). Se hace a través de un examen final que tiene naturaleza global e integra todos los conocimientos y destrezas de carácter teórico adquiridos durante el curso.

A partir de estos elementos se calcula la nota final con las siguientes ponderaciones:

$$\text{Final} = 0.25 A + 0.50 P + 0.25 F$$

NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Las actividades se realizarán siguiendo los usos y costumbres del trabajo académico y, particularmente, se respetarán las siguientes pautas:

1. Aquellas actividades que sean explícitamente declaradas como individuales, sean de naturaleza presencial o no, se realizarán sin ninguna colaboración por parte de otras personas.
2. Las fechas, formatos y demás condiciones de entrega que se establezcan serán de obligado cumplimiento.
3. El uso del laboratorio informático se reservará exclusivamente para las actividades académicas y en ningún caso se podrá hacer un uso abusivo.



BIBLIOGRAFÍA

Básica:

- Kernighan, Brian W.; Dennis M. Ritchie. The C programming language [en línea]. 2nd ed. New Jersey: Prentice Hall, 1988 [Consulta: 20/06/2024]. Disponible a : https://search-ebshost-com.recursos.biblioteca.upc.edu/login.aspx?direct=true&AuthType=ip,uid&db=nlebk&AN=1598613&site=ehost-live&ebv=EB&ppid=pp_Cover. ISBN 0131103628.
- Barnett, Richard H.; O'Cull, Larry; Cox, Sarah. Embedded C programming and the Atmel AVR. 2th ed. New York: Delmar, 2007. ISBN 9781418039592.
- Barrett, Steven F. Embedded systems design with the Atmel AVR microcontroller [en línea]. San Rafael, Calif.: Morgan & Claypool, 2010 [Consulta: 14/06/2024]. Disponible a : <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=881375>. ISBN 9781608451272.
- Harbison, Samuel P.; Steele, Guy L. C, a reference manual. 5th ed. Upper Sadle River: Prentice-Hall, 2002. ISBN 013089592X.