Estructura interna de los sistemas operativos CÓDIGO:12055

Ingeniero en Informática (en extinción)
Escuela de Ingeniería y Arquitectura, Zaragoza

Departamentos: Informática e Ingeniería de Sistemas

Áreas: Arquitectura y Tecnología de Computadores

Curso: 5
Duración: 1º cuatrimestre
Carácter: Optativa
Tipo: Teórica Práctica
Idioma: Inglés

Horas teóricas: 3
Horas prácticas: 15
Créditos UZ: 6
Créditos ECTS: 5,1

Oferta de plazas de libre elección:
Propia Titulación:
Otras Titulaciones: S/L
y/u:
Otros Centros: S/L
Nº Plazas optativas: S/L

Objetivos y Programa | Profesores y Bibliografía | Horario / Observaciones

Objetivos

Estructura Interna de Sistemas Operativos (EISO; Internal Structure of Operating Systems, ISOS) incide en el diseño e implementación de un sistema operativo multiprogramado para un computador basado en un procesador monothread, como fundamento para comprender las bases de los sistmas operativos disponibles más extendidos. Esto proporciona una sólida base para tareas de Programación y Administración de Sistemas. Al final del curso los alumnos obtienen:
Un buen conocimoento de los fundamenos de un núcleo monolítico, a fin de que el estudiante puede entender con claridad sus ventajas, inconvenientes y límites, y la diferencia entre sistemas basados en esta clase de núcleos y los basados en micronúcleos.
Un conocimiento en profundidad del núcleo de Linux y de la estructura de SVR4/Solaris
Experiencia básica - intermedia (según el trabajo del alumno) en programación en el núcleo y de drivers /módulos.
Perspectiva en la interacción ISA - SO, con un buen conocimiento acerca de la arquitectura de sistema de Intel, y una visión amplia de las de MIPS, SPARC y Power PC en algunos aspectos importantes.
EISO /ISOS complementa el curso de Administración de Sistemas, particularmente en el campo de monitorización y configuración de un sistema operativo. Se tienen particularmente en cuenta las características de sistemas operativos modernos como SVR4/Solaris o Mach (adoptado como soporte del sistema operativo Leopard)Junto con la optativa "Dispositivos de E/S y Periféricos", EISO /ISOS permite al estudiante hacer frente a la programación de drivers de dispositivos con relativa facilidad.
La importancia de leer y manipular código real ha sido subrayada a menudo por los educadores de Computer Science (cfr. Orientaciones Académicas de la ACM), al menos por dos razones:
Un SO es un proyecto de programación muy complejo. Las ideas y técnicas utilizadas en él son el mejor ejemplo para muchos algoritmos estudiados en otros cursos de sistemas operativos y de programación, y proporcionan una experiencia de programación realmente utilizable.
La mayoría de los cursos sobre SO tratan procesos, ficheros, threads etc como abstracciones. Los programadores de Sistemas necesitan ir más allá de estas abstracciones, para conocerlas en términos de estructuras y funciones reales.
EISO /ISOS endereza la curva de aprendizaje fácilmente para quellos programadores de sistemas que necesitan adaptar software libre en sistemas críticos y sistemas tiempo real empotrados. El curso aborda con seriedad y desde un punto de vista práctico técnicas de mejora de la seguridad desde el interior del núcleo del sistema operativo.
Nos referimos continuamente a casos reales a lo largo del curso: GNU/Linux para el trabajo en Laboratorio, y toda la familia Unix para la teoría, desde Unix clásicos (SVR3, BSD4.2)a los modernos (SVR4 y Solaris, 4.4BSD, Mach 3.0), con algunas referencias puntuales a otras variantes comerciales (HP-UX, IRIX). El código del núcleo de Linux se utiliza directamente como ejemplo siempre que el criterio pedagógico lo aconseja.

Programa

Unit 0 - Course Introduction
PART 1 - Fundamentals
Unit 1: Architecture support to the Operating System
Unit 2 - The Unix structure: general overview
PART 2 - PROCESS MANAGEMENT
Unit 3 - Kernel structures in process management
Unit 4 - Process scheduling
Unit 5 - Process control
Unit 6 - Introducing threads
PART 3 - MEMORY MANAGEMENT
Unit 7 - Memory management
Unit 8 - Kernel memory management
PART 4 - FILE & I/O MANAGEMENT
Unit 9 - Unix File Systems and I/O
Unit 10 - The Unix Buffer Cache and the File Subsystem
Unit 11 - I/O and Files: system call implementation
[Unit 12 - I/O in modern Unix'es]

Ejercicios de Laboratorio y Miniproyecto
Ejercicio 0
Objetivos: Familiarizar al estudiante con los procedimientos de operación básicos en el entorno del Laboratorio (núcleo Linux).
Descripción: Instalación de fuentes del núcleo, modificaciones elementales, compilación y obtención de un ejecutable modificado, y puesta en marcha del sistema modificado.
Exercise 1
Objetivos: Consolidar el conocimiento sobre las estructuras y funciones internas del núcleo para la gestión de procesos y el interfaz de llamadas al sistema. Estudio de las características singulares de un kernel monolítico y del sistema de sicronización basado en bloqueo en colas de espera. Consolidar los conceptos de implementación de threads de librería, de sistema y de núcleo. Complementa la Unidad 2 y el Módulo 2.
Descripción: Implementación de una nueva llamada al sistema. Por medio de esta llamada al sistema un proceso de usuario puede obtener información acerca de llamadas al sistema efectuadas por procesos de un determinado usuario. La

llamada tiene que tener carácter atómico y falla si es ejecutada por otro proceso. Se ofrecen diferentes extensiones para los alumnos interesados en profundizar en ello (llamada reentrante etc).
Miniproject
Objetivos: Consolidar conceptos de la asignatura y adquirir experiencia práctica.
Descripción. Los miniproyectos se basan en los siguientes temas:
Programación de drivers / módulos instalables
Instrumentación del núcleo para monitorización y sintonización o mejora de la seguridad (ocultación de ficheros y módulos sensibles, acceso a funciones y estructuras internas de ficheros, procesos o gestión de memoria etc).
El Ejercicio 0 entrenamiento práctico. El Exercise 1es obligatorio para superar la asignatura. El Miniproyecto es voluntario. Tanto el Miniproyecto como partes optativas del Ejercicio 1 añaden puntos extra a la puntuación final.

Evaluación

Deben de entregarse un cierto número de ejercicios de clase y para casa, a menudo a través de Moodle. Las prácticas se evalúan mediante presentación oral con demostración práctica.

universidad de zaragoza ©2006