

Linux industrial, de tiempo real e integrado

Curso práctico de 4 días - 28h

Ref.: LXT - Precio 2024: 2 020€ sin IVA

En este curso, descubrirá la arquitectura general de un sistema embebido e implementará las herramientas de desarrollo industrial disponibles libremente en Linux. Implementará mecanismos de tiempo compartido y programación flexible en tiempo real, y accederá al desarrollo en tiempo real estricto con la extensión LinuxRT o Xenomai. Por último, verá cómo personalizar el arranque de un sistema Linux.

OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Descubra las herramientas de desarrollo industrial en Linux

Mecanismos maestros de programación en tiempo compartido y flexible en tiempo real

Acceso al desarrollo en tiempo real estricto con la extensión LinuxRT o Xenomai

Personalización del arranque de un sistema Linux

Los numerosos ejercicios y casos prácticos progresivos se realizan en una red de servidores Linux. Todos los programas creados durante las sesiones prácticas están disponibles en forma de esqueletos que los participantes pueden completar por sí mismos.

PROGRAMA

última actualización: 05/2024

1) Arquitectura

- Sistemas informáticos ordinarios y de a bordo.
- Limitaciones de los sistemas integrados.
- Arquitectura general de un sistema empujado.
- Puesta en marcha del sistema, fase de arranque.
- Arquitectura del núcleo Linux. Ubicación de las fuentes.
- Puesta en marcha del sistema, fases de arranque (código dependiente y común).

Trabajo práctico : Detección de errores en tiempo de compilación, en tiempo de enlace, uso de Eclipse/CDT, uso de una cadena de compilación cruzada. Depuración. Detección de fugas de memoria y desbordamientos de búfer. Pruebas de cobertura en la ejecución de la aplicación.

2) Desarrollo industrial con Linux

- Entorno Linux.
- Modo de funcionamiento: usuario, supervisor.
- Licencias e implicaciones para el desarrollo industrial.
- Herramientas de desarrollo de código abierto (compilador, depurador, herramientas de análisis, seguimiento y prueba).
- Los distintos IDE (Entorno de Desarrollo Integrado) : Eclipse...
- Métodos avanzados de compilación.
- La cadena de compilación cruzada.
- Gestión de la memoria.
- Detección de fugas de memoria.
- Desbordamiento del búfer.

PARTICIPANTES

Desarrolladores Linux/Unix.

REQUISITOS PREVIOS

Buenos conocimientos de un sistema Linux/Unix y de programación en C.

COMPETENCIAS DEL FORMADOR

Los expertos que imparten la formación son especialistas en las materias tratadas. Han sido validados por nuestros equipos pedagógicos, tanto en el plano de los conocimientos profesionales como en el de la pedagogía, para cada curso que imparten. Cuentan al menos con entre cinco y diez años de experiencia en su área y ocupan o han ocupado puestos de responsabilidad en empresas.

MODALIDADES DE EVALUACIÓN

El formador evalúa los progresos pedagógicos del participante a lo largo de toda la formación mediante preguntas de opción múltiple, escenificaciones de situaciones, trabajos prácticos, etc. El participante también completará una prueba de posicionamiento previo y posterior para validar las competencias adquiridas.

MEDIOS PEDAGÓGICOS Y TÉCNICOS

- Los medios pedagógicos y los métodos de enseñanza utilizados son principalmente: ayudas audiovisuales, documentación y soporte de cursos, ejercicios prácticos de aplicación y ejercicios corregidos para los cursos prácticos, estudios de casos o presentación de casos reales para los seminarios de formación.
- Al final de cada curso o seminario, ORSYS facilita a los participantes un cuestionario de evaluación del curso que analizarán luego nuestros equipos pedagógicos.
- Al final de la formación se entrega una hoja de presencia por cada media jornada de presencia, así como un certificado de fin de formación si el alumno ha asistido a la totalidad de la sesión.

MODALIDADES Y PLAZOS DE ACCESO

La inscripción debe estar finalizada 24 horas antes del inicio de la formación.

ACCESIBILIDAD DE LAS PERSONAS CON DISCAPACIDAD

¿Tiene alguna necesidad específica de accesibilidad? Póngase en contacto con la Sra. FOSSE, interlocutora sobre discapacidad, en la siguiente dirección psh-accueil@orsys.fr para estudiar de la mejor forma posible su solicitud y su viabilidad.

3) Tiempo compartido flexible y programación en tiempo real

- Exactitud de las mediciones y expectativas de tiempo.
- Programación a tiempo compartido. Reglas de anticipación.
- Funcionamiento general del planificador, prioridades y reparto de CPU.
- ¿Cuándo es preferente el núcleo?
- Gestión del tiempo y temporizadores de precisión.
- Tiempo real flexible Posix.1b: principios.
- Configuración de la programación de procesos e hilos.
- Problemas algorítmicos vinculados al tiempo real.
- Planificador de Linux: núcleos 3.x

Trabajo práctico : Creación y gestión de procesos. Prioridades y compartición de CPU.

Examen del funcionamiento del planificador. Cambio de procesos en tiempo real.

Comprobación de la precisión de temporizadores y durmientes.

4) Tiempo real estricto - Extensión Xenomai

- Principio estricto de tiempo real.
- Visión general de la extensión Xenomai.
- Conceptos de tiempo real estricto: principio de los microkernels Adeos, Xenomai, LinuxRT.
- Instalación y API de Xenomai.
- Uso de la extensión LinuxRT.
- Programación estricta en tiempo real en modo usuario.
- Interrupciones (activación, desactivación, etc.).
- Protección contra interrupciones.
- Gestión de las comunicaciones.
- Presentación de la API Xenomai, instalación de Xenomai.
- Estricta gestión de tareas en tiempo real.

Trabajo práctico : Instalación de Xenomai. Creación de tareas estrictas en tiempo real.

Gestión de las comunicaciones. Escribiendo un manejador de interrupciones y un proceso programado de tiempo real estricto. Instalación y uso de Xenomai.

5) Entornos restringidos, sistemas integrados

- Problemas de los sistemas integrados.
- Sistema LinuxRT, Xenomai: API, desarrollo.
- Embedded Linux: elegir una versión del kernel.
- Bibliotecas del sistema (Newlib, DietLibc).
- Aplicaciones y utilidades a bordo.
- Interfaz de usuario.
- Interfaces gráficas optimizadas (directfb, etc).
- Presentación y configuración de un cargador de arranque.
- Generación de un núcleo reducido. Información general sobre el sistema de archivos.
- Instalación de la cadena de compilación.

Trabajo práctico : Instalación y compilación de una aplicación LinuxRT personalizada, Xenomai y un núcleo de tamaño reducido. Creación de un sistema de archivos.

Incorporación de aplicaciones mínimas. Escritura de aplicaciones utilizando una interfaz LED o LCD. Instalación de un servidor HTTP integrado.

6) Personalización del arranque del sistema

- Las distintas fases de arranque (encendido, Bios, cargador del kernel (Grub, UBoot, etc.).
- El papel del proceso Init. El nivel de ejecución.
- El contenido del Init.
- Sustitución del proceso Init por una versión personalizada.
- Inicialización desde el área de usuario.

Trabajo práctico : Creación y personalización de un disco initrd. Sustitución del proceso Init por una versión personalizada y sustitución por un script de Shell.

FECHAS

Contacto